

Robotic Arm Kinematics and Calibration 6-DOF Powerball LWA 4P

Dr. Curtis Bradley PE

February 28, 2019

This work covers the specifics to performing kinematic calibration on the Schunk Powerball Light Weight Arm (LWA) 4.6 using a serial approach starting with Circular Point Analysis (CPA) followed by a Gauss-Newton error minimization method. The serial approach allows a simplified regressor for the Gauss-Newton method. While this serial technique of combining these two vastly different parameter identification methods is completely developed it is not fully not optimized for numerical accuracy. Additionally, a solution to the inverse kinematics is completely solved and presented as the precursor for follow-on research.



The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

The citation in this report of the names of commercial firms or commercially available products or services does not constitute official endorsement by or approval of the U.S. Government.

Destroy this report when no longer needed by any method that will prevent disclosure of its contents or reconstruction of the document. Do not return to the originator.

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> OMB No. 0704-0188		
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.						
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.						
1. REPORT DATE (<i>DD-MM-YYYY</i>) 28-02-2019		2. REPORT TYPE Technical Report		3. DATES COVERED (<i>From - To</i>)		
4. TITLE AND SUBTITLE Robotic Arm Kinematics and Claibration 6-DOF Powerball LWA 4P				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) C. Bradley				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Combat Capabilities Devlopment Command Armaments Center Benet Laboratories Watervliet, NY 12189-4000				8. PERFORMING ORGANIZATION REPORT NUMBER ARWSB-TR-19014		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)		
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER		
12. DISTRIBUTION AVAILABILITY STATEMENT Distribution A: Approved for public release: distribution unlimited.						
13. SUPPLEMENTARY NOTES PHD Thesis Work, Rensselaer Polytechnic Institute, Troy, NY 12180.						
14. ABSTRACT This work covers the specifics to performing kinematic calibration on the Schunk Powerball Light Weight Arm (LWA) 4.6 using a serial approach stating with Circular Point Analysis (CPA) followed by a Gauss-Newton error minimization method. The serial approach allows a simplified regressor for the Gauss-Newton method. While this serial technique of combining these two vastly different parameter identification methods is completely developed it is not fully not optimized for numerical accuracy. Additionally, a solution to the inverse kinematics is completely solved and presented as the precursor for follow-on research.						
15. SUBJECT TERMS Kinematics; Kinematic Calibration; Schunk Powerball Light Weight Arm (LWA); Circular Point Analysis (CPA); Gauss-Newton method						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 143	19a. NAME OF RESPONSIBLE PERSON C. Bradley	
a. REPORT U	b. ABSTRACT SAR	c. THIS PAGE SAR	19b. TELEPHONE NUMBER (<i>Include area code</i>) (518) 266-4838			

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g., 30-06-1998; xx-08-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. 1F665702D1257.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. AFOSR-82-1234.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORS AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

CONTENTS

1. Introduction	1
2. Kinematics	2
2.1. Forward Kinematics	2
2.2. Inverse Kinematics	4
3. Control	6
4. Kinematic Calibration	9
4.1. Circle Point Analysis Estimation	10
4.2. Gauss-Newton Estimation	14
5. Results	17
6. Conclusion	19
A. Inverse Kinematics Matlab Code	21
B. Circular Point Analysis Matlab Code	25
C. Gauss-Newton Minimization Matlab Code	34
D. Jacobian for DH Parameters Matlab Code	41

LIST OF FIGURES

1.1. Powerball LWA 4.6	1
2.1. Nominal Powerball Zero Configuration Diagram	3
2.2. Inverse Kinematics Example Solution	6
3.1. System Overview	7
3.2. Individual Joint Motion Profile	8
3.3. Joint Angle Data	9
4.1. Circular Point Analysis Results	13
4.2. Gauss Newton Solution	16
4.3. Cartesian Components of Camera Data Vs. FK	17
5.1. 3D plot of Camera Data Vs. FK	18
5.2. Gauss-Newton Mean Error	18

LIST OF TABLES

2.1. Nominal DH Parameters	3
2.2. Test Joint Angles (radians)	5
2.3. IK Solution Joint Angles	6
4.1. Final DH Parameters	16

ACKNOWLEDGMENTS

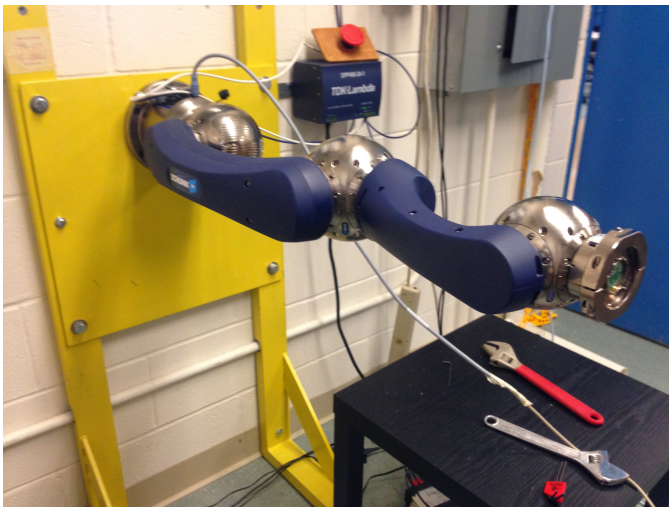
I would like to express my great appreciation to Professor Trinkle for giving me such complete access to his lab and robot along with support for all aspects of my research. His time and generosity made this work not only possible, but enjoyable. I am grateful to Professor Bevilacqua for helping to make this opportunity possible. I would like to express my deepest gratitude to Bryant Pong for his tireless pursuit of capturing data and the recording, formatted, and outputting thereof, which greatly facilitated this research. His programming skill in various languages and ROS expertise solved execution problems in a timely manner. I appreciate the technical support received from Jesse Hayes at Schunk to keep the arm running smoothly. Jun Dong for always lending a hand with Python coding, general construction and words of encouragement. Shuai Li for donating his monitor as well as Python scripting help.

(THIS PAGE INTENTIONALLY LEFT BLANK)

1. INTRODUCTION

The purpose of this independent research study was to gain an in-depth familiarity with this current high end robotic manipulator. The robot used was a Schunk Powerball light weight arm (LWA) 4.6, the latest 6-Degree-of-Freedom (DOF) arm from Schunk, leaders in manipulator design [1]. Two areas of research were explored, namely formulating the full inverse kinematics and performing a kinematic calibration of the system. The inverse kinematics (IK) solution is necessary for fast operation in the world coordinate frame and also required for compliant control algorithms like impedance control. The model type used for calibration is a parametric kinematic type using Denavit-Hartenberg (DH) parameters. Kinematic calibration is the first calibration type that allows the proceeding calibrations, like inertial calibration and friction identification, to be more accurate.

The Powerball is an arm designed for both industrial automation as well as research. It has a high gear ratio for each of its axes making it insensitive to gravitational loading, but since the gearing is harmonic and the ratio is not too high it can still be back driven for impedance control and friction identification. The light compact and modular design allowed fast recovery from multiple failures during initial set-up. Axis 4 failed shortly after the system was functional and was replaced when spare parts become available. A subsequent failure of axes 1 and 2, were able to be quickly replaced by dismantling both the original arm and a partial replacement arm to reconstruct a completely functional arm without any direct intervention required from the manufacturer. The low maximum speed was a necessary performance trade in exchange for payload and weight, but makes control schemes like minimum-time control irrelevant in this velocity limited regime. The Powerball can be used with any controller that supports multiple CANOpen devices, similar to the Robotic Operating System (ROS) framework combined with a suitable communication card. ROS allowed seamless integration with the camera tracking system and robot controller while allowing further extensions of the system in the future. A high resolution rendering of the model used to simulate and visualize the Powerball is shown in Fig. 1.1(a).



(a) Schunk Powerball lab installation



(b) Schunk Powerball rendered model

Figure 1.1: Powerball LWA 4.6

The solution to the IK problem was solved using the nominal DH parameters determined from the manufacturers dimensions and from actuating the joints to determine offsets and axis direction. Using the nominal parameters allowed the IK solution to be conducted in parallel with the kinematic calibration work. The IK solution is universally applicable inside the work envelope and includes topology information for the configuration of the robot since the general IK solution has eight valid joint config-

uration. The solution presented in this research also removes solutions beyond the joint limits. Further there is an optional input for the last known joint position so only the closest solution to the that previous joint configuration is returned to simplify trajectory generation.

Kinematic calibration used a joint trajectory sequentially using all six joints where the end effector motion profile was measured where calibration used both Circular Point Analysis (CPA) and Gauss-Newton error minimization. CPA was used to orient the end effector tracking data captured in the camera frame to the robot frame where the DH parameters are defined. Once the data is oriented Gauss-Newton error minimization can be used to directly solve for all the DH parameters at one time without adding slack variables for extra degrees of freedom. Iteratively minimizing the error between the forward kinematics (FK) and the tracking data using the Gauss-Newton method requires multiple equation substitutions. To simplify this process and more it more computationally efficient a symbolic version of the forward kinematics was compiled to a function. This function simplification also extends to the Jacobian allowing the simplified versions of these equations to be more computationally efficient.

2. KINEMATICS

Kinematics defines the rigid body motion of the robotic manipulator by defining the relationship between joint configurations and a working reference frame. The FK solution maps joint angles to an end effector position and orientation in the working reference frame. The IK solution maps a given end effector position and orientation to a set of joint angles. Depending on the configuration of the robot the mapping in one direction or the another may be mathematically trivial or there could be multiple solutions or no closed form solution at all. The FK is required for the kinematics calibration while the IK is required for advanced control algorithms and direct workspace motion planning. Robot structures are often described with an ordered list of joint types, "R" for a revolute joint and "T" for translational joint, the Powerball is an RRRRRR robot with all six joints being revolute, shown in Fig. 2.1.

2.1. FORWARD KINEMATICS

Deriving the FK can be done in a variety of ways each with advantages and disadvantages, but they all start with a parameterization for the joints positions and orientations. Denavit-Hartenberg(DH) parameters are one parameterization used in research and industry. Standard DH parameters have been used throughout this research to allow more straight forward comparisons to other research in the area. The DH parameters are θ , d , a , and α , with θ the input variable for revolute joints and d the input variable for translational joints. The DH parameters define the transformation from one joint coordinate system to the next joint in the kinematic chain, θ defines rotation between the z-axis, d defines the distance along the z-axis between joints, a defines the distance along the x-axis between joints, and α defines the angle between joint z-axes. Each of the 6 degrees of freedom of the Powerball are a revolute joint making the θ DH parameter the input variable for every set of four DH parameters that define each joint axis. Even though θ is an independent input variable a DH parameter, θ_{offset} , must still be added to θ to form the exact FK based on the actual joint angles defined by the absolute joint axis motor encoders. In the case of the Powerball the DH parameter representation is not unique because axes pairs one/two, three/four, and five/six are near intersecting additionally axes two and three are nearly parallel [2]. Table 2.1 defines the nominal DH parameter set used in this research. Using table 2.1 the coordinate transformation between adjacent joints in the serial chain are defined by Eq. 2.1 [3],

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.1)$$

Powerball Nominal DH Configuration

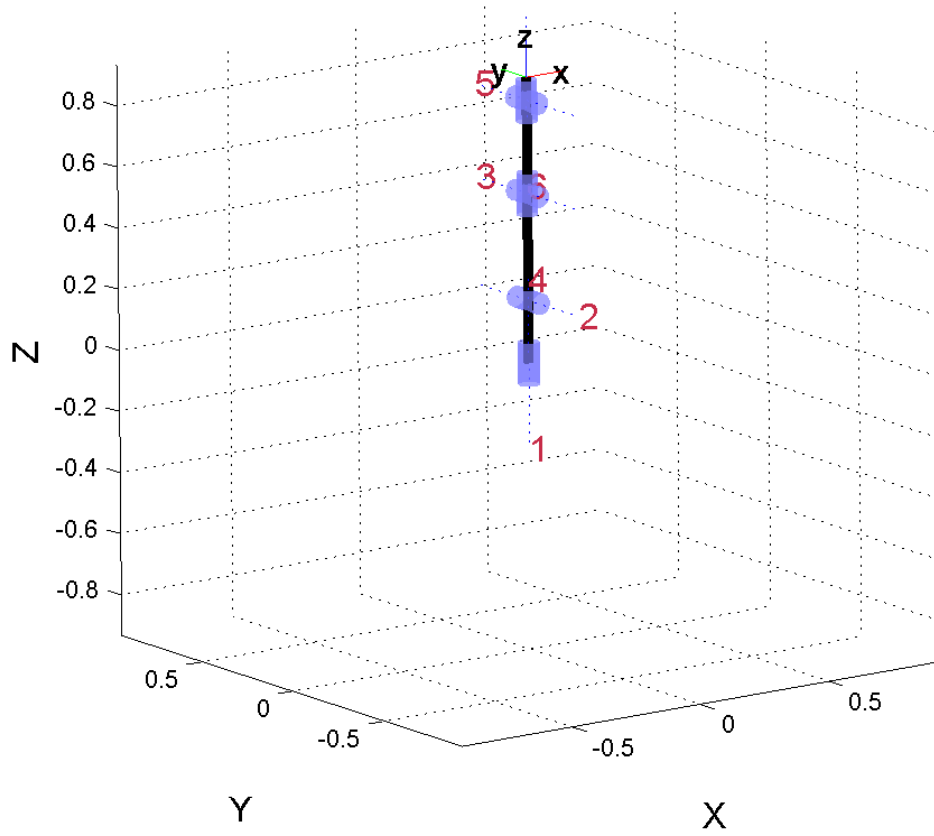


Figure 2.1: Nominal Powerball Zero Configuration Diagram

Since the Powerball is composed of a set of serially linked joints the complete FK solution results from

Joint Axis	θ_{offset}	d	a	α
1	0	$0.205m$	0	$-\frac{\pi}{2}$
2	$-\frac{\pi}{2}$	0	$0.350m$	π
3	$-\frac{\pi}{2}$	0	0	$-\frac{\pi}{2}$
4	0	$0.305m$	0	$\frac{\pi}{2}$
5	0	0	0	$-\frac{\pi}{2}$
6	0	$0.075m$	0	0

Table 2.1: Nominal DH Parameters

the matrix multiplication of each successive joint transform defined by Eq. 2.1, so for this or any serial 6-DOF robot this Eq. 2.2 holds true,

$${}^0T_6 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6. \quad (2.2)$$

The 4×4 homogeneous transform matrix, ${}^0T_6$ can be thought of as divided into an $SO(3)$ rotation matrix ($R_{3 \times 3}$), a translation vector ($t_{3 \times 1}$), a perspective vector ($p_{1 \times 3}$), and a scalar scale factor ($s_{1 \times 1}$). The

composition of ${}^0T_6$ is seen in Eq. 2.3. For rigid body or affine transformations that in this case represent the robot axes transformations p_{1x3} equals 0_{1x3} and s_{1x1} equals 1,

$${}^0T_6 = \begin{bmatrix} R_{3x3} & t_{3x1} \\ p_{1x3} & s_{1x1} \end{bmatrix}. \quad (2.3)$$

So given all six joint values, $[\theta_1 \dots \theta_6]$ and the correct set of twenty-four DH parameters, the FK solution is fully defined by the homogeneous transform matrix, ${}^0T_6$ which defines the position and orientation of the end effector in the sixth axis of the Powerball. The general formulation using all twenty-four DH parameters for ${}^0T_6$ will resolve to a manageable equation as there are only six variables and for most robots the nominal DH parameters have several zero entries. If the FK matrix, ${}^0T_6$, retains a symbolic representation for the non-zero DH parameters it will be large and if all twenty-four DH parameters retain their symbolic representation ${}^0T_6$ becomes very large, more than 25,000 characters per matrix row.

2.2. INVERSE KINEMATICS

The IK will be necessary for fast direct workspace trajectory solutions and advanced control algorithms like impedance control. The inverse kinematics problem is to solve for the joint angles given a specific robot end effector position and orientation. For a 6-DOF robot with a spherical wrist there is a closed form solution; the Powerball arm is of this class of robot. For simplicity of implementation this solution uses a homogeneous transformation matrix as an input along with an optional "previous angles" vector that will allow the return of a single solution that is closest to the previous angles vector in the vector norm sense for easier trajectory implementation. The complete functional implementation using Matlab[®] is in appendix A and solves in under a millisecond on a single CPU.

The outline for this solution is to solve joint 3 then joints 1 and 2 simultaneously, then solve joints 5, 4, and 6 as a set of Euler angles which have a straight forward standard solution. To solve angle 3, first ${}^0T_6$ must be moved to the center of the spherical wrist by translating the reference frame by the distance of the constant tool vector length in the direction of the current orientation of the end effector using Eq. 2.4,

$$dx = {}^0T_6(1:3, 1:3) \begin{bmatrix} 0 \\ 0 \\ d_6 \end{bmatrix}. \quad (2.4)$$

Similarly the vector from the robot base to the common center of joints 1 and 2, shoulder joint center, must also be removed from the end effector vector yielding the vector from the center of the shoulder joint to the center of the spherical wrist from Eq. 2.5,

$$d_{elbow} = {}^0T_6(1:3, 4) - dx - \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix}. \quad (2.5)$$

With the distance d_{elbow} all three sides of the triangle formed by the two primary sections of the Powerball arm, a_2 and d_4 , and the vector d_{elbow} can be used to solve the law of cosines for θ_3 as in Eq. 2.6,

$$\theta_3 = \pm \left(\pi - \arccos \frac{a_2^2 + d_4^2 - |d_{elbow}|^2}{2 * a_2 * d_4} \right). \quad (2.6)$$

The angle θ_3 , the "elbow", has two solutions classically termed "elbow up" and "elbow down", these solutions are independent of all other joints making it a convenient choice to solve first. The topology for the solution is retained as a binary number where each binary digit stores the topology of that solution, in the case of θ_3 a 1 in the second digit means the solution has the elbow up. This allows for better

trajectory planning to avoid unnecessary jumps between topologies.

The method chosen to solve joints 1 and 2 uses the second Paden-Kahan standard subproblem and has two solutions for each of the two previous solutions for θ_3 , for a total of four solutions so far. The second Paden-Kahan standard subproblem also called just subproblem 2 simultaneously solves the problem of rotating a vector two consecutive rotations about two arbitrary unit vectors to another vector [4]. The arbitrary rotation unit vectors are the axes for axis 1 which is also the z axis and the other is axis 2, the y axis. To outline a solution to subproblem 2 the axes and vectors needed to solve it are shown in Eq. 2.7,

$$\text{axis } k_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \text{axis } k_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{p} = \begin{bmatrix} 0 \\ 0 \\ a_2 \end{bmatrix} + \begin{bmatrix} -d_4 \sin \theta_3 \\ 0 \\ d_4 \cos \theta_3 \end{bmatrix}, \quad \vec{q} = \overrightarrow{d_{elbow}}. \quad (2.7)$$

Subproblem 2 solves for the rotation of $\vec{p}$, the zero configuration of the vector from the shoulder to the wrist, first rotated about k_1 then about k_2 which will align it with $\vec{q}$, the actual vector from shoulder to wrist. This must be done for both θ_3 solutions. The specific numerical methods used to solve subproblem 2 are clearly labeled in appendix A as the function "subproblem2". The two solutions are traditionally termed "shoulder left" and "shoulder right", these topologies are stored in the least significant digit, the right-most-bit, 1 for right shoulder.

With the angles for the first three joints the end effector orientation can be rotated back to the base coordinate system which will then only represent the spherical wrist transform, also an Euler angle rotation. The $SO(3)$ rotation matrix formed by these three rotations, roll, pitch, roll, has a straight forward closed form solution directly from the rotation matrix, ${}^4R_6$, equation from the homogeneous transformation matrix,.

$${}^4R_6 = ({}^0R_1^1 R_2^2 R_3^3)^T {}^0R_6 \quad (2.8)$$

$${}^4R_6 = \begin{bmatrix} \cos \theta_4 \cos \theta_5 \cos \theta_6 - \sin \theta_4 \sin \theta_6 & -\cos \theta_6 \sin \theta_4 - \cos \theta_4 \cos \theta_5 \sin \theta_6 & -\cos \theta_4 \sin \theta_5 \\ \cos \theta_4 \sin \theta_6 + \cos \theta_5 \cos \theta_6 \sin \theta_4 & \cos \theta_4 \cos \theta_6 - \cos \theta_5 \sin \theta_4 \sin \theta_6 & -\sin \theta_4 \sin \theta_5 \\ \cos \theta_6 \sin \theta_5 & -\sin \theta_5 \sin \theta_6 & \cos \theta_5 \end{bmatrix} \quad (2.9)$$

The solution is then to use Eq. 2.8 on the given ${}^0R_6$ rotation matrix portion of the transform matrix ${}^0T_6$ with the rotation matrices for the first three joints already found. Then in the (3,3) position for the $SO(3)$ rotation matrix from Eq. 2.9 is the cosine of joint angle 5. Then the remaining angles are defined as $\theta_4 = \text{atan2}((2,3), (1,3))$ and $\theta_6 = \text{atan2}((3,2), -(3,1))$. All Euler angle transforms have 2 solutions and in this case they represent the "wrist up" and "wrist down" configurations, making the total number of solutions 8 in general, limited only by joint limits and singularities. The wrist up topology is represented by a 1 in the most significant bit, the left most bit.

To illustrate the IK solution 6 random angles are generated within the joint limits then use the FK from Eq. 2.2 to generate the homogeneous transformation matrix, ${}^0T_6$ (Eq. 2.10), which is the input to the IK function,

$${}^0T_6 = \begin{bmatrix} -0.834 & -0.118 & -0.538 & -65.739 \\ -0.200 & 0.975 & 0.095 & -56.379 \\ 0.514 & 0.187 & -0.837 & 468.762 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.10)$$

One set of sample joint angles are in Table 2.2 and the corresponding solution are in Table 2.3.

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
-1.950	-0.717	-2.081	2.575	1.634	0.938

Table 2.2: Test Joint Angles

The complete set of 8 solutions all exist for this particular set of starting joint angles and are shown in table 2.3. The set of 8 illustrated IK solutions is shown in Fig. 2.2 where the actual solution is highlighted

	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6	Solution 7	Solution 8
θ_1	1.191	-1.951	1.191	-1.951	1.191	-1.951	1.191	-1.9506
θ_2	0.717	1.130	0.717	1.130	-1.130	-0.717	-1.130	-0.717
θ_3	2.081	2.081	2.081	2.081	-2.081	-2.081	-2.081	-2.0807
θ_4	-0.566	0.715	2.575	-2.427	-2.427	2.575	0.714	-0.567
θ_5	1.634	2.186	-1.634	-2.186	2.186	1.634	-2.186	-1.634
θ_6	0.938	-1.699	-2.203	1.443	-1.699	0.938	1.443	-2.203
Topology	111	110	011	010	101	100	001	000

Table 2.3: IK Solution Joint Angles

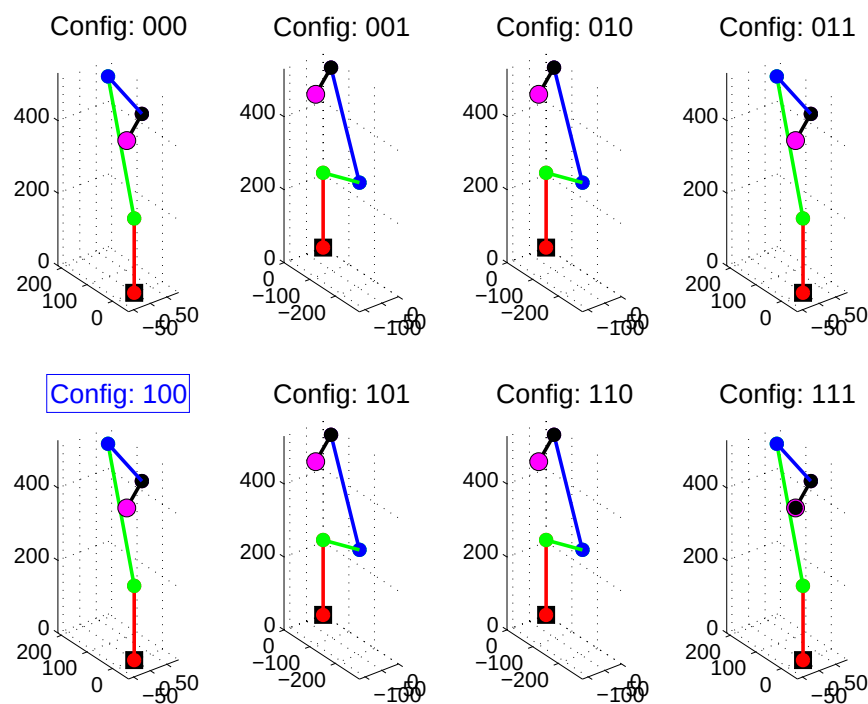


Figure 2.2: Inverse Kinematics Example Solution

in blue and the subplot titles are the binary numbers representing the topology of each solution. Every solution shows the end effector in the same location and orientation, but the robot uses a different set of joint angles to achieve the pose. With the Powerball arm it is difficult to see the difference between some configurations because there are three pairs of intersecting axes, making left shoulder and right shoulder look identical.

3. CONTROL

The system overview is outlined in Fig. 3.1. The Powerball robot arm is composed of six independently controlled servo motors each using the CANopen protocol over a common serial port using the CANopen drivers in the `ipa_canopen` ROS package. The `ipa_canopen` package includes several tools for operating

a previous version of the Powerball, but are not compatible with the Powerball LWA 4.6. The CANopen drivers successfully implement the portion of the protocol for sending velocity commands and receiving joint axis position and velocity feedback. Motor current or motor effort feedback is not available from the driver in ROS, but is implemented in the CANopen protocol and the joint axis controller hardware. Having state feedback and velocity control in ROS allows for several control activities, but the missing portions of the CANopen protocol preclude direct torque control, position control, and inertial calibration using joint torques at this time. Included with the ROS package are several functions for operating

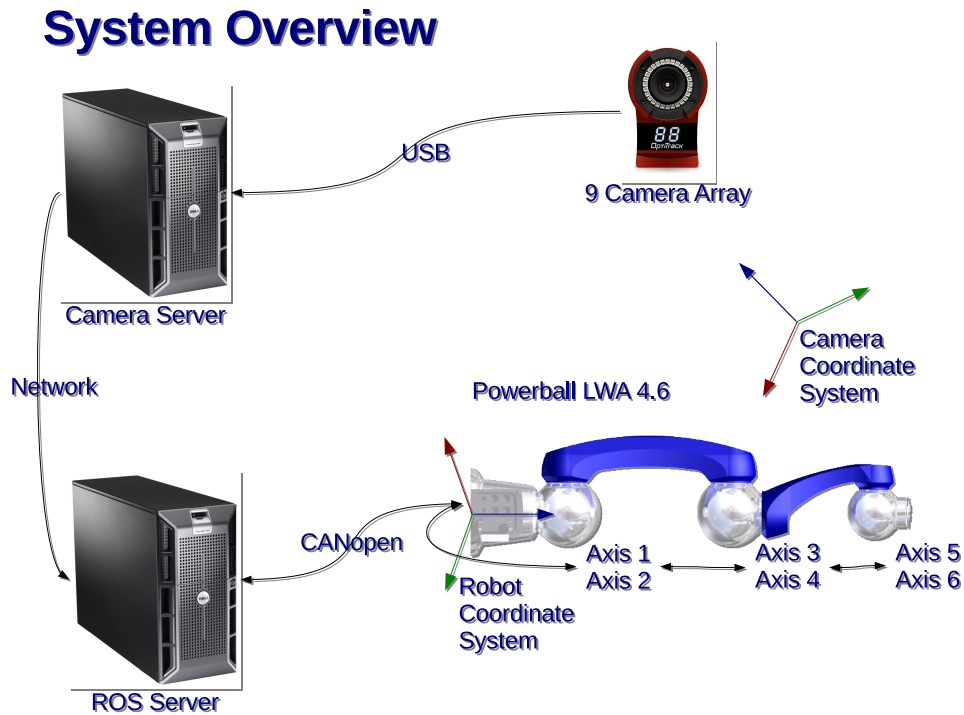


Figure 3.1: System Overview

and visualizing the Powerball specifically. The dependencies for these functions were not well documented, but after tracking down the build failures one at a time what seemed like all the dependencies were met without the Powerball dashboard ever working. The heart of the issue seems to be an incomplete update between versions of ROS, from Fuerte to Groovy. Ultimately only the rudimentary CANopen functions could be used in conjunction with the stand-alone diagnostic functions provided by Schunk for the Powerball.

Using the direct velocity control available with the CANopen driver in ROS in the `ipa_canopen` package changes velocity with a step function. The step changes in velocity excited structural vibrations that were higher than expected because of the combination of a light weight robot arm coupled with a flexible mounting plate. Structural vibrations add what amounts to noise to the camera tracking data since both kinematic calibration methods use a rigid body assumption for the model of the robot links. A profile with bounded jerk was used to minimize vibration. The profile uses a sinusoidal acceleration function which has a closed form solution for creating a velocity profile with a given displacement and execution time, seen in Fig. 3.2. The closed form solution makes the programmatic implementation of the profiles using velocity control possible as just a function of time in an open-loop manner. Closed loop implementations were investigated, but could not be made viable without substantial additional scripting for ROS. The open-loop method was sufficient since the only strict requirement for the motion profile was

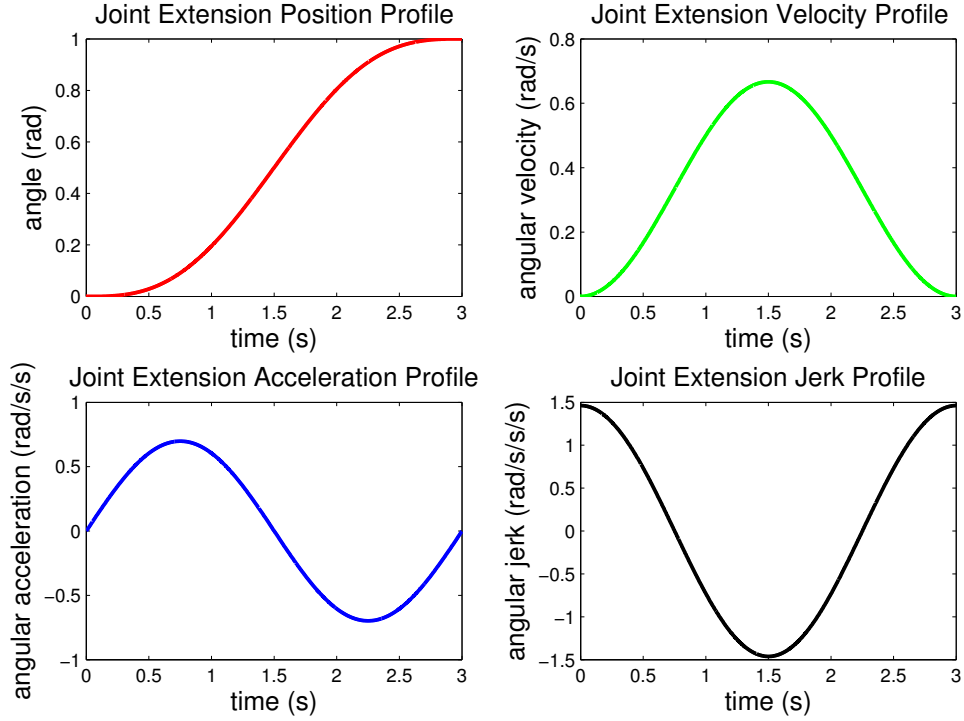


Figure 3.2: Individual Joint Motion Profile with Bounded Jerk

that joints be actuated one at a time with bounded jerk. The velocity profile is generated by Eq. 3.1 as a function of the current ROS time initialized to zero at the start of a joint motion command,

$$velocity(t) = range \, Fs - range \, Fs \cos(t * Fs * 2 * \pi), \quad (3.1)$$

where Fs is the frequency of the motion, $range$ is the total range of motion of the move, and t is time starting at zero and continuing linearly to the reciprocal of Fs .

Since the purpose of the joint actuation was to perform kinematic calibration all the joints needed to be actuated to make the DH parameters observable. The CPA kinematic calibration method requires the joint be sequentially actuated while all other joints remain fixed. In this analysis the joints were actuated from joint one to six over a range from zero to a maximum back to approximately zero. A Gauss-Newton error minimization can also use this joint trajectory allowing both kinematic calibration methods to be used sequentially on the same data set.

The system architecture of ROS uses a single central server that communicates to individual process nodes using “subscribers” and “publishers” and manages the communication between them. This server-node architecture works well with the Powerball because it acts as six separate CANopen nodes all communicating over the CAN bus additionally the OptiTrack camera tracking system broadcasts its data making for easy integration as another publishing node in ROS. During testing the data timing and availability varies between subscribers that record joint angle data and the subscriber that records camera tracking position data. The built-in ROS system absolute time is used to sync data sets and is in units of nanoseconds. Despite using what should be the same time scale there is a significant delay between joint and camera data that seems to be in the joint data set. The delay was found manually using features in the data to match the start of motion in both the joint and camera data. Any slight error in this delay can have a large effect on the overall accuracy of a solution to the calibration problem.

Another aspect of the data timing issues is the rate difference that data can be either subscribed or published. In many situations the data is available from various processes at a rate much faster than the rate it can be accessed by the main ROS processing loop, making timing differences between processes much smaller than the difference between complete loop iterations times. In this set of experiments there is a high variation in when joint data is available from the Powerball modules. There was a maximum lag between readings of 1.8 s, and a median of 0.002 s. This is compared to the camera tracking data with a maximum lag between reading 0.009 s and a median of 0.0083 s. This large discrepancy between data sets meant that only data that was captured at relatively the same time could be used as input data points for any calibration using joint data. Following in Fig. 3.3 are two plots showing data point selection with the entire set of over 11,000 joint data points in Fig. 3.3(a), and the 172 data points that had tight agreement with camera data in Fig. 3.3(b). Syncing the data would have been required even without the inconsistent

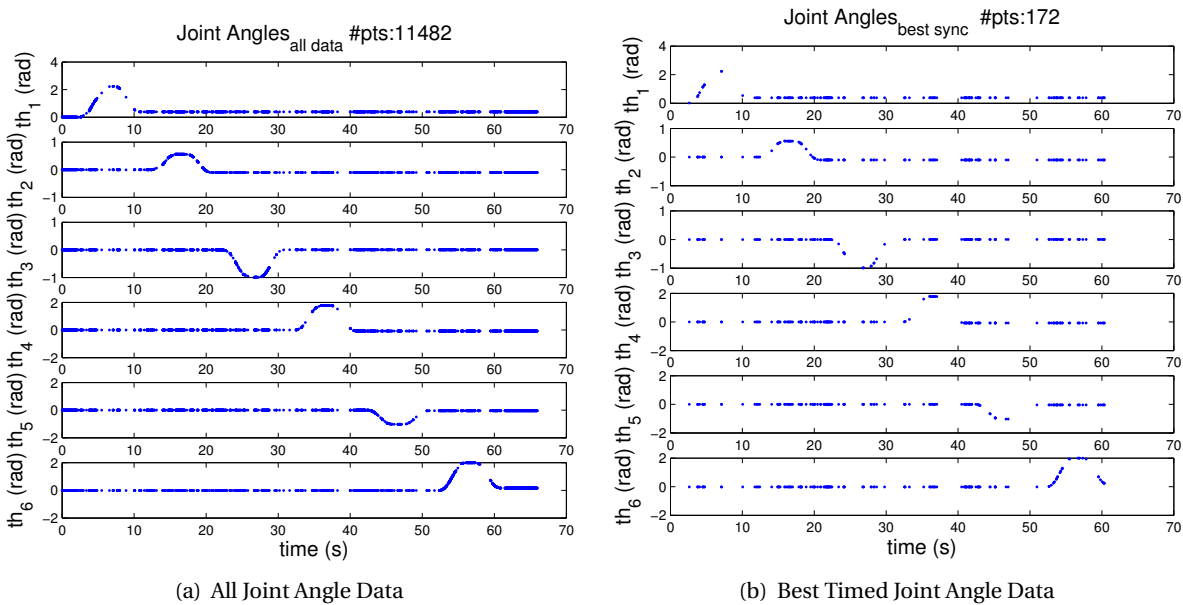


Figure 3.3: Joint Angle Data for All Six Joints Showing Timing

timing of the joint data because the base data rates for the camera data and the joint data are different, but unfortunately there are gaps in joint data seen in Fig. 3.3(a). Gaps in the joint data output also meant loss of control of the joint modules which caused jumps in the motion profile and unnecessary vibration of the mounting structure adding to noise in camera capture data. The source of the data gaps is unclear whether it is from the robot modules sending data or the ROS server collecting the data.

4. KINEMATIC CALIBRATION

Calibration is the process of determining the precise system parameters that affect performance in order to achieve the highest level of performance possible. Refining the FK is the purpose of kinematic calibration. For this work the robot involved is relatively rigid and so is well described by a parametric model using DH parameters for the FK. For robot arms the various calibration methods are divided into two main classes open loop and closed loop, where closed-loop constrains at least one degree of freedom of the end effector motion [5], while open-loop does not [6]. Both calibration methods used, CPA and Gauss-Newton minimization, are done open-loop. Kinematic calibration was performed by using CPA to fully orient the camera data from the camera frame to the robot frame and set some of the DH

parameters as the starting point of the Gauss-Newton least-squares minimization.

4.1. CIRCLE POINT ANALYSIS ESTIMATION

Circular point analysis (CPA) is an early method used to perform kinematic calibration because of the straight forward nature of the calculation. CPA leverages the ability to control each given joint in a known sequence while holding all other joint angles or positions fixed and tracking at least one point distal to the joint being identified [7]. In terms of DH parameters, a revolute joint has a locus of points in a circle centered at the joint z-axis in a plane perpendicular to that axis, while a translational joint has a locus of points that forms a line parallel to the z-axis. In the case of the Powerball arm all joints are revolute.

The accuracy of the CPA method for each joint is highly sensitive to the radial distance of the tracking point from the joint axis. This is seen in Eq. 4.1 where as r goes to 0 Eq. 4.1 becomes degenerate as the points all overlap and fail to define a plane [8],

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2. \quad (4.1)$$

One advantage of using the CPA method is that it only requires tracking discrete points and not the full pose of the end effector. This made the camera tracking process easier since tracking just the position of a body requires fewer track points than tracking the full pose of a rigid body [9]. CPA can be performed using purely the tracking data from the camera with only the loss of the joint axis direction defined by α and θ_{offset} . Using only the tracking data for the CPA portion of the kinematic calibration allowed the elimination of the extra link reference frame typically added to align the camera reference frame to the robot reference frame, simplifying the Gauss-Newton minimization. Also, the nominal values for α and θ_{offset} were sufficient as a starting point for the Gauss-Newton method to converge to a solution.

The analysis was done one joint axis at a time, identifying first the direction then the location of the joint axis relative to the camera reference frame. The direction of the axis was calculated using a least-squares approximation solving for the normal of the plane containing all the tracking points using the following vector form of the plane equation,

$$n_x(x - x_0) + n_y(y - y_0) + n_z(z - z_0) = 0 \quad [8], \quad (4.2)$$

using the centroid of the points, the arithmetic mean, as the reference point on the plain (x_0, y_0, z_0) ,

$$x_0 = \bar{x}, \quad y_0 = \bar{y}, \quad z_0 = \bar{z},$$

and then arbitrarily scaling the normal vector by assigning a value to one of the components of the normal, for instance the z direction. Then incorporating all the camera data points into a matrix form Eq. 4.2 then becomes,

$$\begin{bmatrix} (x_1 - x_0) & (y_1 - y_0) \\ (x_2 - x_0) & (y_2 - y_0) \\ \vdots & \vdots \\ (x_n - x_0) & (y_n - y_0) \end{bmatrix} \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} (z_1 - z_0)n_z \\ (z_2 - z_0)n_z \\ \vdots \\ (z_n - z_0)n_z \end{bmatrix}. \quad (4.3)$$

The normal must be scaled in the direction with the minimum range to avoid the resulting Eq. 4.3 from becoming singular if the normal was scaled in a direction perpendicular to the plane being solved for. Equation 4.3 is overdetermined because there are more than three equations to define (n_x, n_y, n_z) , this overdetermination helps account for zero biased measurement noise. The solution that minimizes the $L2$ -norm between the points and the model of the plane uses the Moore-Penrose pseudoinverse of the

matrix multiplying the unknowns of the plane normal in Eq. 4.3. If the following equation holds,

$$p = \begin{bmatrix} n_x \\ n_y \end{bmatrix},$$

$$A = \begin{bmatrix} (x_1 - x_0) & (y_1 - y_0) \\ (x_2 - x_0) & (y_2 - y_0) \\ \vdots & \vdots \\ (x_n - x_0) & (y_n - y_0) \end{bmatrix}, \text{ and}$$

$$q = \begin{bmatrix} (z_1 - z_0)n_z \\ (z_2 - z_0)n_z \\ \vdots \\ (z_n - z_0)n_z \end{bmatrix},$$

then the least-squares solution for (n_x, n_y, n_z) is,

$$p = A^* q, \text{ where } A^* \text{ is the pseudoinverse.} \quad (4.4)$$

With the solution of the plane parameters, $(n_x, n_y, n_z, x_0, y_0, z_0)$, that define the joint axis direction and the plane that contains the points defining the circle traced out by the joint then the location of the axis in that plane can be solved as well. Since a circle is a subset of a sphere the locus of points for that CPA should satisfy the sphere equation, Eq. 4.1. Equation 4.1 is nonlinear, but a linear least-squares solution using the pseudoinverse can still be used by rearranging the equation. First the linear constraints are separated out by expanding out Eq. 4.1 [10],

$$x^2 - 2xx_0 + x_0^2 + y^2 - 2yy_0 + y_0^2 + z^2 - 2zz_0 + z_0^2 = r^2, \quad (4.5)$$

then rearranged to collect linear, non-linear, known and unknown terms,

$$2xx_0 + 2yy_0 + 2zz_0 + (x_0^2 + y_0^2 + z_0^2 + r^2) = (x^2 + y^2 + z^2), \text{ where } (x_0^2 + y_0^2 + z_0^2 + r^2) = \zeta, \text{ then,} \quad (4.6)$$

$$\begin{bmatrix} 2x & 2y & 2z & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \zeta \end{bmatrix} = x^2 + y^2 + z^2. \quad (4.7)$$

Now Eq. 4.7 can be expanded to include all the camera data points,

$$\begin{bmatrix} 2x_1 & 2y_1 & 2z_1 & 1 \\ 2x_2 & 2y_2 & 2z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 2x_n & 2y_n & 2z_n & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \zeta \end{bmatrix} = \begin{bmatrix} x_1^2 + y_1^2 + z_1^2 \\ x_2^2 + y_2^2 + z_2^2 \\ \vdots \\ x_n^2 + y_n^2 + z_n^2 \end{bmatrix}. \quad (4.8)$$

If we substitute into Eq. 4.8,

$$p = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \zeta \end{bmatrix},$$

$$A = \begin{bmatrix} 2x_1 & 2y_1 & 2z_1 & 1 \\ 2x_2 & 2y_2 & 2z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 2x_n & 2y_n & 2z_n & 1 \end{bmatrix},$$

$$q = \begin{bmatrix} x_1^2 + y_1^2 + z_1^2 \\ x_2^2 + y_2^2 + z_2^2 \\ \vdots \\ x_n^2 + y_n^2 + z_n^2 \end{bmatrix}.$$

Then the solution of Eq. 4.8 uses the pseudoinverse to solve the least-squares problem the same as the plane least-squares solution in Eq. 4.4,

$$p = A^* q. \quad (4.9)$$

This solution is degenerate because Eq. 4.1 defines a sphere, so the solution is ill-conditioned in the normal direction of the plane containing the circle of camera tracking points and already found in Eq. 4.4 as the vector (n_x, n_y, n_z) . This is not an issue because the joint axis can be defined at any point along the axis, but in order to keep the coordinates within the working envelope of the robot the point can be reflected back to the plane as follows [11],

$$\text{Let } N = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \begin{bmatrix} n_x & n_y & n_z \end{bmatrix},$$

then the projection equation is,

$$\begin{bmatrix} x_{proj} & y_{proj} & z_{proj} \end{bmatrix} = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix} (I - N) + \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix} N. \quad (4.10)$$

The solution using this CPA method shows all the axes in Fig. 4.1(a), but this only finds the axis locations in the camera reference frame where they were captured, which is not useful for the Gauss-Newton error minimization because the DH parameters solve the forward kinematics in the robot reference frame where the z_0 axis is aligned with the z_{world} axis. The CPA solved for all the axes using over 8000 captured camera data points in 2 seconds using a single CPU thread.

With the locations of all the joint axes from the CPA method solved this information can be used to both align the z_0 axis with the z_{world} axis of the data and generate a starting point for the kinematic calibration using a Gauss-Newton error minimization between the forward kinematics and the camera data aligned to the robot reference frame. The camera data can all be moved by a coordinate transformation in six DOF. Four degrees-of-freedom are needed to align the z axes, two rotation and two translation. The remaining two degrees of freedom are used to align two of the DH parameters roughly to their default values, translation to set d_1 as the default height of the robot base and rotation to set $\theta_{offset:1}$ equal to zero. Ideally these values would be determined from a geometrically suitable datum that can be directly measured and has significance for the real-world operation of the robot, like the actual mounting height of the base to define d_1 . The robot operations are, as of yet undefined, so this research only deals with the accurate set-up of the robot through calibration that can be extended to suit future work.

The rotation matrix uses the normalized cross product between the identified axis 1 and the desired z

axis, n_{cam} , and the $acos$ of the dot product, θ_{cam} to orient the data to axis 1 using the matrix exponential Eq. 4.11,

$$R_{cam} = e^{[n_{cam}]_x \theta_{cam}}, \quad (4.11)$$

where $[n_{cam}]_x$ is the skew-symmetric matrix operator used to perform the cross product as a matrix multiplication. The translation offset for the data is then,

$$offset_{cam} = -R_{cam} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}.$$

To set d_1 to the default value, which is used to set the location of axis 2 in the aligned coordinate system. To set $\theta_{offset:1}$ the data is simply rotated about the z axis to align the camera point where the θ_1 equals zero to the x axis to agree with the default DH parameters. The final alignment of the data showing the location of the identified axes is show in Fig. 4.1(b).

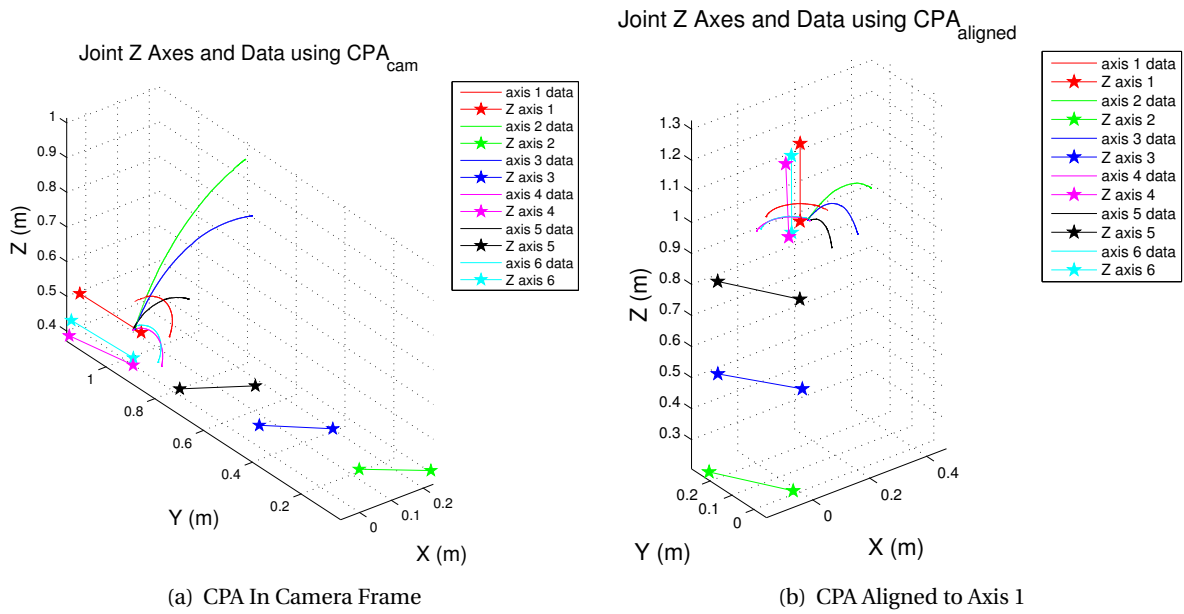


Figure 4.1: Circular Point Analysis Results

In order to get initial estimates for two of the distance DH parameters the minimum distance between certain axes must be calculated, namely a_2 between axes 2 and 3 and d_4 between 3 and 5. The equation uses the normal vectors for the joints and their normalized cross product, for example, n_2 , n_3 and n_{dist} with coordinates on those axes similar to the identified joint circle center, c_2 and c_3 , then

$$dist_1 = c_1 - \frac{((c_2 - c_1 \times n_2) \cdot n_{dist})}{(n_{dist} \cdot n_{dist})} n_1 \quad (4.12)$$

$$dist_2 = c_2 - \frac{((c_2 - c_1) \times n_1) \cdot n_{dist})}{(n_{dist} \cdot n_{dist})} n_2 \quad (4.13)$$

$$distance = norm(dist_2 - dist_1) [12]. \quad (4.14)$$

The value for a_2 is derived to be 319 mm versus the nominal value of 350 mm and for d_4 the derived value is 300 mm versus 305 mm nominal. The source for the error between the derived and nominal cannot be known with measuring the effects of a flexible mounting, camera data error and actual robot

manufacturing differences. The Matlab[®] code that uses CPA to align the camera data to speed and simplify the Gauss-Newton calibration is in appendix B.

4.2. GAUSS-NEWTON ESTIMATION

The Gauss-Newton method is an estimation method that allows the solution to a least squares minimization problem for a non-linear well defined function [13]. The method uses a linear approximation of the non-linear function formed from partial derivatives of the unknown variables, also known as a Jacobian, J . Equation 4.15 shows the structure of the non-linear parametric equation as a function of constant parametric values and inputs,

$$\Delta \mathbf{y}^i = f(\mathbf{x}^i, \phi), \text{ where} \quad (4.15)$$

$\mathbf{y}^i = y_1, \dots, y_n$ is the vector of all measured outputs, $\mathbf{x}^i = x_1, \dots, x_n$ is the vector of all measured inputs, and ϕ is the set of constants defining the non-linear parametric model, in this case the DH parameters [13]. For kinematic calibration the non-linear function is the FK matrix ${}^0T_6$ or some subset of it. For this analysis just the translational portion of ${}^0T_6$ will be used, defining the location of the end effector with Cartesian coordinates $f_x(\mathbf{x}^i, \phi)$, $f_y(\mathbf{x}^i, \phi)$, $f_z(\mathbf{x}^i, \phi)$. This kinematics calibration solution uses the complete set of DH parameters from the forward kinematics equation which is the same for all RRRRRR robots, θ_{offset} , d , a , and α . Using just the position portion of the forward kinematics equations has several advantages, most notably the camera tracking system has a high degree of accuracy for a minimum set of track points where as orientation tracking requires more track points to maintain the same degree of accuracy as mentioned in the CPA subsection. All the DH parameters will be observable using just end effector position except α_6 , traditionally taken as zero since it defines the arbitrary orientation of the end effector coordinate frame independent of robot geometry.

Since $f_x(\mathbf{x}^i, \phi)$, $f_y(\mathbf{x}^i, \phi)$, $f_z(\mathbf{x}^i, \phi)$ are nonlinear there is no closed form solution method to solve for the values of ϕ , so the Gauss-Newton or some other search method must be used. This Gauss Newton method uses a first term Taylor series expansion of Eq. 4.15,

$$y^i = f^i(\phi + \Delta\phi) \quad (4.16)$$

$$\approx f^i(\phi) + \left. \frac{\partial f^i(\phi)}{\partial \phi} \right|_{\phi} \Delta\phi \quad [13]. \quad (4.17)$$

If we substitute into the Taylor series term in Eq. 4.17, $J^i = \frac{\partial f^i}{\partial \phi}$, also know as a regressor in this role then,

$$y^i \approx f^i(\phi) + J\Delta\phi. \quad (4.18)$$

Rearranging Eq. 4.18 and substituting $\Delta y = y^i - f^i(\phi)$, the error term between the actual, y^i , and the calculated, $f^i(\phi)$, output yields the linearized update equation,

$$\Delta y^i = J^i \Delta\phi.$$

The superscript i indicates the i^{th} equations which includes different functions $f^i(\phi)$ and different measurements, but the least squares estimation form uses all the equations stacked in tall vectors and a tall

matrix and is,

$$\begin{bmatrix} x_1 - f_x^1 \\ \vdots \\ x_n - f_x^n \\ y_1 - f_y^1 \\ \vdots \\ y_n - f_y^n \\ z_1 - f_z^1 \\ \vdots \\ z_n - f_z^n \end{bmatrix} = \begin{bmatrix} x J_{\theta_{offset:1}}^1 & \dots & x J_{\theta_{offset:6}}^1 & x J_{d_1}^1 & \dots & x J_{d_6}^1 & x J_{a_1}^1 & \dots & x J_{a_6}^1 & x J_{\alpha_1}^1 & \dots & x J_{\alpha_6}^1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x J_{\theta_{offset:1}}^n & \dots & x J_{\theta_{offset:6}}^n & x J_{d_1}^n & \dots & x J_{d_6}^n & x J_{a_1}^n & \dots & x J_{a_6}^n & x J_{\alpha_1}^n & \dots & x J_{\alpha_6}^n \\ y J_{\theta_{offset:1}}^1 & \dots & y J_{\theta_{offset:6}}^1 & y J_{d_1}^1 & \dots & y J_{d_6}^1 & y J_{a_1}^1 & \dots & y J_{a_6}^1 & y J_{\alpha_1}^1 & \dots & y J_{\alpha_6}^1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y J_{\theta_{offset:1}}^n & \dots & y J_{\theta_{offset:6}}^n & y J_{d_1}^n & \dots & y J_{d_6}^n & y J_{a_1}^n & \dots & y J_{a_6}^n & y J_{\alpha_1}^n & \dots & y J_{\alpha_6}^n \\ z J_{\theta_{offset:1}}^1 & \dots & z J_{\theta_{offset:6}}^1 & z J_{d_1}^1 & \dots & z J_{d_6}^1 & z J_{a_1}^1 & \dots & z J_{a_6}^1 & z J_{\alpha_1}^1 & \dots & z J_{\alpha_6}^1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ z J_{\theta_{offset:1}}^n & \dots & z J_{\theta_{offset:6}}^n & z J_{d_1}^n & \dots & z J_{d_6}^n & z J_{a_1}^n & \dots & z J_{a_6}^n & z J_{\alpha_1}^n & \dots & z J_{\alpha_6}^n \end{bmatrix} \begin{bmatrix} \theta_{offset:1} \\ \vdots \\ \theta_{offset:6} \\ d_1 \\ \vdots \\ d_6 \\ a_1 \\ \vdots \\ a_6 \\ \alpha_1 \\ \vdots \\ \alpha_6 \end{bmatrix}. \quad (4.19)$$

Then the compact estimation form of Eq. 4.19 is then

$$\Delta y = J \Delta \phi. \quad (4.20)$$

The least square solution for the difference in ϕ for one set of data is then

$$\Delta \phi = J^* \Delta y, \quad (4.21)$$

where J^* is the Moore-Penrose pseudoinverse. Since J is non-linear several iterations of estimates will need to be generated before ϕ will converge to a solution by updating the previous estimate with $\Delta \phi$ from Eq. 4.21, $\phi^{k+1} = \phi^k + \Delta \phi$.

For a given set of data consisting of a series of joint angles, $\theta_{1...n}^1$, and corresponding camera tracking points, $\begin{bmatrix} x^{1...n} \\ y^{1...n} \\ z^{1...n} \end{bmatrix}$. Where n must be large enough so there are enough measurements to have full rank of J meaning all parametric model parameters, ϕ are observable. As mentioned earlier in this section α_6 is unobservable because only end effector position is tracked, but using the 172 data points with well correlated camera and joint data mentioned in section 3 J maintained full rank with $n = 172$ as expected since the motion profile has motion for every joint. This makes the Jacobian J of size 516 by 24 for this data set.

As mentioned in section 2.1 the FK solution that generates the functions for the Jacobian is large, around 25,000 characters, so taking the partial derivatives of the functions for position, $f_x(\mathbf{x}, \phi)$, $f_y(\mathbf{x}, \phi)$, $f_z(\mathbf{x}, \phi)$, will also be large. In this case the portion of J for one data point is over 296,000 characters. This equation would have been even larger if an additional reference frame was added to account for the camera frame as mentioned in subsection 4.1. Using the CPA method to align the data for use in the Gauss-Newton Estimation allowed the elimination of one frame that traditionally accounts for aligning the camera frame to the robot frame. Each data point requires a symbolic substitution into that large 3 by 24 subset of the Jacobian. Using the best built-in Matlab[®] function for this type of substitution, "subs", each data point took about 1.18 s, for 174 points and 15 iterations the solution takes about 3000 s making investigations into different starting points for ϕ and for larger data sets prohibitively time consuming. Parallel processing multiple data points at a time only gave a marginal reduction in processing time running 12 threads on a workstation. A reduction in processing time from 1.18 s to 8 μ s was accomplished by computing the fully symbolic Jacobian for one data point as a function and then compiling the function to

an executable "C-MEX" file for direct execution. Now a complete Gauss-Newton solution took about 45 s. The complete Matlab[®] code that the C-MEX code is generated from is in appendix D.

The combination of non-linearities, maintaining ill-conditioned DH parameters like those for parallel axes and biased errors made the direct application of the Gauss-Newton method unstable. During iterations of the solution limit-switching because of the bounded trigonometric functions, constrained to $\pm\pi$, would cause the error to oscillate and with some initial conditions go unstable. To keep the angular parameters reasonable without effecting the solution the values were forced to roll-over between $\pm 2\pi$ which produces the same output without discontinuities. The relative size of the parameters in ϕ are all on the order of 1, so there are no gross scaling issues to contribute to instability. To stabilize the search a relaxation term was added to the regressor calculation. The relaxation term is the "golden ratio", $\frac{2}{1+\sqrt{5}}$. This allowed the algorithm to converge to a solution despite the numerical issues mentioned. The technique converges to different solutions depending on the initial values chosen. The solution to all 24 DH parameters is shown in table 4.1 while the convergence of the parameters are shown in Fig. 4.2. The Matlab[®] code used to solve the Gauss-Newton calibration is in appendix C. The code in appendix C also calls the code in appendix D. There is a large difference between the DH parameters derived from

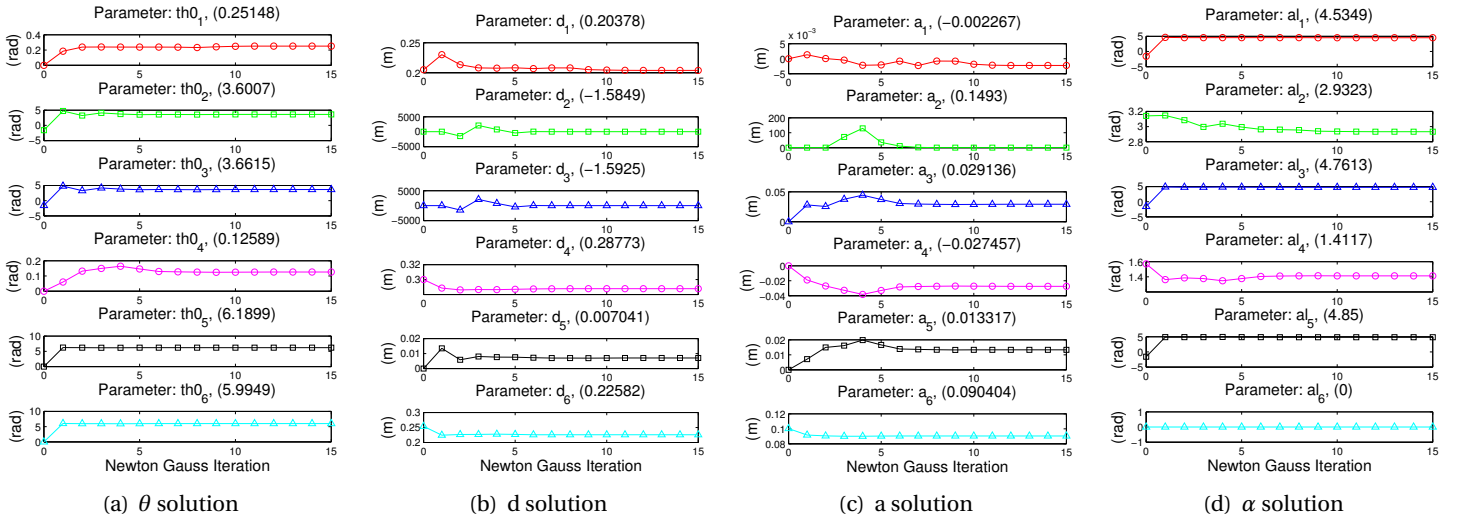


Figure 4.2: Gauss Newton Analysis Results

Joint Axis	θ_{offset}	d	a	α
1	0.251	0.204m	-0.002m	4.535
2	3.600	-1.585m	0.350m	2.932
3	3.662	-1.593m	0.029m	4.761
4	0.125	0.288m	-0.027m	1.411
5	6.190	0.007m	0.013m	4.85
6	5.995	0.226m	0.090m	0

Table 4.1: Final DH Parameters

the Gauss-Newton method in table 4.1 and the initial DH parameters in table 2.1. The Cartesian components of the FK using the initial DH parameters and final DH parameters are in Fig. 4.3, where the actual camera data is represented by red, green, and blue $\bullet$, while the FK calculated data is represented by magenta, black and cyan $\circ$. In Fig. 4.3 some data points have better agreement than others ideally

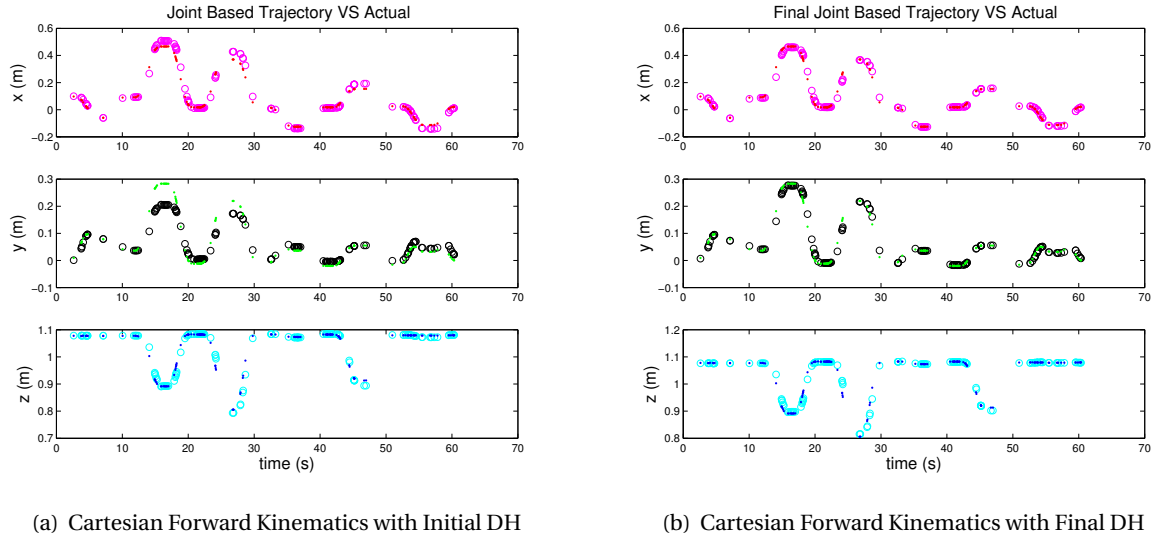


Figure 4.3: Cartesian Components of Camera Data Vs. FK

the plot would be a dot centered inside each circle. The timing issue discussed in section 3 may have caused joint angle data to become mistimed with camera data preventing an accurate solution.

5. RESULTS

A robust solution was found for the IK problem that is closed form and solves fast while maintaining tracking topology of the solution. Optionally, the IK solution finds a solution nearest to another joint configuration in the vector norm sense. Also shown is a method for successfully solving the set of DH parameters that minimizes the error of the FK solution without adding an additional axis to solve for the camera reference frame. Using the CPA method to pre-align the data prior to starting the Gauss-Newton error minimization method was shown as an option for decreasing the order of the Gauss-Newton approximation. This solution was optimized for speed and modified from the standard Gauss-Newton method to relax the convergence and increase the stability making it more tolerant to initial estimates that are further from a solution. The solution to the CPA method created uses only least square solutions without having to resort to more complex search methods or other more time intensive numerical methods.

The kinematic calibration method shows an improvement of fit for the derived parameters that is seen directly from the side-by-side comparison in Fig. 5.1. The better agreement is noted by the camera data when compared to the FK using the Gauss-Newton derived DH parameters than the initial estimate. As noted in subsection 2.1 multiple sets of DH parameters are possible for serial link mechanisms that have intersecting axes like the Powerball [2]. In Fig. 5.2 there is an acute error increase approaching 200 m around iteration 4 before converging, so it is reasonable to expect that the solution may not converge to a solution near the starting estimate. The plots in Fig. 5.1 gives the best intuitive assessment of the fitness of the solution judging how closely the red and blue three dimensional paths overlap. Figure 5.2 shows the minimization of the error over the fifteen iterations until the stopping criterion is met, maximum absolute parameter change less than $1e-4$.

There are issues with the absolute accuracy of the final DH parameter result that will require further testing to completely diagnose at a deeper level that stems from several sources of error. The camera configuration was non-ideal for the robot working envelope and has non-zero biased noise that is not

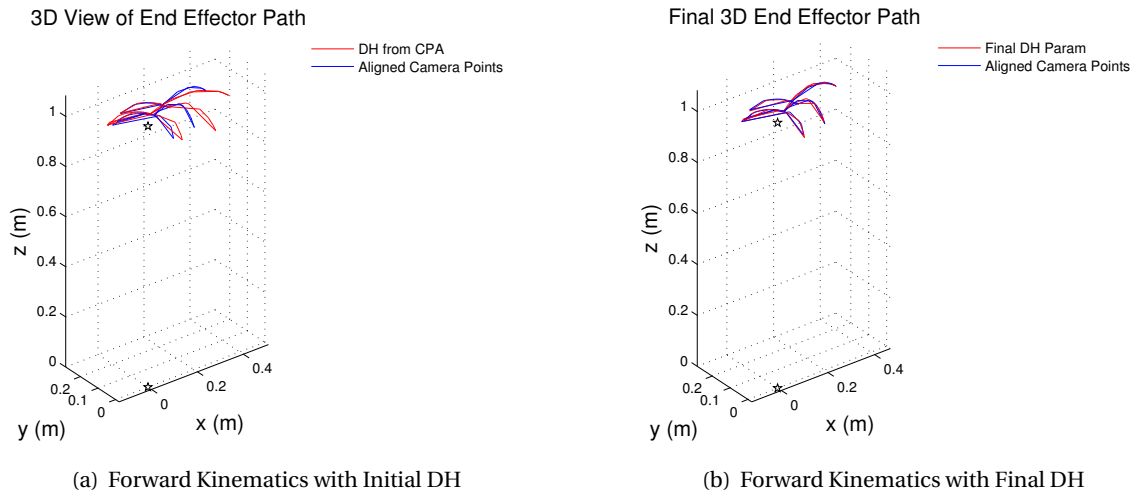


Figure 5.1: Camera Track Points VS FK from Joint Data and DH Parameters

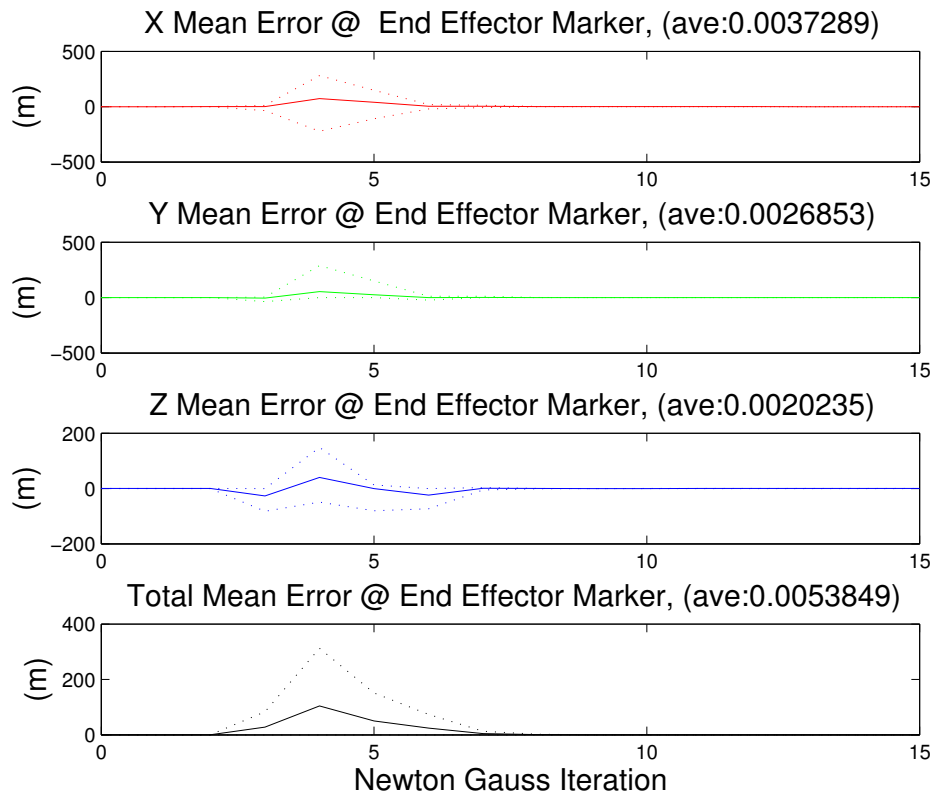


Figure 5.2: Gauss-Newton Mean Error

easily removed with either the CPA or Gauss-Newton methods as they both solve for the least-square minimum errors. Similarly the support structure for the robot was flexible enough to introduce several millimeters of additional non-zero biased error to readings. This was exacerbated by the vibrations introduced by the timing delays in the motion control loop. Finally, the absolute time stamp is another

issue that unnecessarily adds error the solution..

The current motion profile returned near to the zero configuration after each joint move so there were many more data points near the zero configuration biasing the result toward that location. The bias can be generally seen in Fig. 4.3 where the best agreement of the data are along dwell points in the motion where there are the most data points. A video of the system performing several experiments is available [here](#).

6. CONCLUSION

The IK solution presented in this work is shown to be both numerically accurate and computationally efficient making it suitable for use in any future research requiring joint angles given an end effector pose. While this IK solution is useful it is also specific to the non-unique DH representation chosen in this work. Additionally, using the homogeneous matrix representation for end effector pose may not be as computationally efficient or stable as other representations such as quaternions.

The kinematic calibration method presented is shown to be computationally efficient and accurate with respect to the measurement resolution of the robot arm and camera system. Separating the kinematic calibration into a camera alignment phase and robot calibration phase using the CPA and Gauss-Newton methods respectively shows promise in terms streamlining the kinematic calibration computations. Though the final calibration shows an increase in end effector positional accuracy additional research is required to assess the accuracy of the end effector orientation. Additionally, the final solution is very sensitive to the initial estimate of DH parameters and further study should instigate a method that either uses that sensitivity of the Gauss-Newton method or uses a higher order approximation to converge to the set of parameters in a more robust manner.

For future arm calibrations like inertial and friction calibrations new CANopen Process Data Objects (PDOs) will be needed to transmit motor current or torque as the current ROS CANOpen implementation lacks access to this data as well as direct torque control. For the current set of experiments packing paper was a viable, albeit unaesthetic, IR shielding method to permit camera tracking, but future experiments requiring greater range of motion will require a different solution. Paint or some other semi-permanent method may be best. The current configuration of the camera array should also be addressed as the camera work envelop is not currently matched with the Powerball workspace.

REFERENCES

- [1] Powerball lightweight arm lwa 4p. [schunk.com/Powerball Lightweight Arm LWA 4P](http://schunk.com/Powerball%20Lightweight%20Arm%20LWA%204P), 2013.
- [2] Subir Kumar Saha. *Introduction to robotics*. Tata McGraw-Hill, 2008.
- [3] Robert J Schilling. *Fundamentals of robotics: analysis and control*. Simon & Schuster Trade, 1996.
- [4] B. Paden. *Kinematics and Control Robot Manipulators*. Phd thesis department of electrical engineering and computer science, University of California Berkley, 1986.
- [5] Milan Ikits and John M Hollerbach. Kinematic calibration using a plane constraint. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 4, pages 3191–3196. IEEE, 1997.
- [6] John M Hollerbach and Charles W Wampler. The calibration index and taxonomy for robot kinematic calibration methods. *The international journal of robotics research*, 15(6):573–591, 1996.
- [7] John M Hollerbach, Lydia Giugovaz, Martin Buehler, and Yangming Xu. Screw axis measurement for kinematic calibration of the sarcos dextrous arm. In *Intelligent Robots and Systems' 93, IROS'93*.

- Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 3, pages 1617–1621. IEEE, 1993.
- [8] James Stewart. *Essential Calculus: Early Transcendentals*. CengageBrain. com, 2010.
 - [9] Jorge Santolaria and Manuel Ginés. Uncertainty estimation in robot kinematic calibration. *Robotics and Computer-Integrated Manufacturing*, 2012.
 - [10] Neil Tenenholz. Nonlinear model fitting using least squares. harvard.edu/ntenenholz, 2012.
 - [11] Projection of a point to a plane. mathworks.com/plane_projection, 2010.
 - [12] Minimum distance between skew lines. mathworks.com/line_distance, 2007.
 - [13] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2008.

A. INVERSE KINEMATICS MATLAB CODE

```

1 function th_out=powerball_inverse_kinematics(T06,th_p)
2 %%codegen
3 %% Inverse Kinematics Schunk Powerball LWA 4.6
4 % th_out=inverse_kinematics(T06, theta_previous)
5 % T06= [4x4]
6 % theta_previous= [1x6]
7
8
9 th_ik=zeros(7,8); % initialize working solution variable
10 th_limit=[170; 110; 155; 170; 140; 170;]*pi/180; %Symetric Joint limits Schunk
    Powerball 4.6
11 d_1=205; %base
12 a_2=350; %upper arm
13 d_4=305; %forearm
14 d_6=75; %hand
15
16
17
18 %%
19 % Solve for $\theta_3$
20 dx=T06(1:3,1:3)*[0;0;d_6]; % Vector from spherical
    wrist to tool tip
21 d_elbow=T06(1:3,4)-dx-[0;0;d_1]; % Vector with the tool tip distance and
    base distance removed
22 d_elbow_norm =norm(d_elbow); % Distance from joints 1&2 to spherical
    wrist
23
24 temp=pi-acos((a_2^2+d_4^2-d_elbow_norm^2)/(2*a_2*d_4)); % Angle of Elbow
25
26 th_ik(3,1:4)=temp; th_ik(7,1:4)= th_ik(7,1:4)+ 2^1;% Elbow up
27 th_ik(3,5:8)=-temp; %th_ik(7,5:8)=th_ik(7,5:8)+0*2^1; % Elbow Down
28
29
30 %%
31 % Solve for $\theta_1$ & $\theta_2$
32
33 [theta1_1,theta2_1]=subproblem2([0;0;1],[0;1;0],[0; 0; a_2; ]+
    [-d_4*sin(th_ik(3,1)); 0; d_4*cos(th_ik(3,1))],d_elbow); % 2 Solutions for
    theta3(1) Elbow up
34 [theta1_2,theta2_2]=subproblem2([0;0;1],[0;1;0],[0; 0; a_2;
    ]+[-d_4*sin(th_ik(3,5)); 0; d_4*cos(th_ik(3,5))],d_elbow); % 2 Solutions
    for theta3(2) Elbow down
35
36 % Replace invalid solutions from subproblem2
37 if isnan(theta1_1(1)) || isnan(theta1_1(2))
38     theta1_1=[th_p(1), th_p(1)];
39 end

```

```

40
41 if isnan(theta1_2(1)) || isnan(theta1_2(2))
42     theta1_2=[th_p(1), th_p(1)];
43 end
44
45 if isnan(theta2_1(1)) || isnan(theta2_1(2))
46     theta2_1=[th_p(1), th_p(1)];
47 end
48 if isnan(theta2_2(1)) || isnan(theta2_2(2))
49     theta2_2=[th_p(1), th_p(1)];
50 end
51
52 th_ik(1,1:4)=[theta1_1(1:2) theta1_1(1:2)];    th_ik(7,[1,3])=th_ik(7,[1,3])+2^0;
53     %Shoulder right
54 th_ik(1,5:8)=[theta1_2(1:2) theta1_2(1:2)];    th_ik(7,[5,7])=th_ik(7,[5,7])+2^0;
55     %Shoulder right
56
57 th_ik(2,1:4)=[theta2_1(1:2) theta2_1(1:2)];
58 th_ik(2,5:8)=[theta2_2(1:2) theta2_2(1:2)];
59
60 %%
61 % $\theta_4$, $\theta_5$, $\theta_6$
62 for z=[1:2, 5:6]
63     th_1=th_ik(1,z);
64     th_2=th_ik(2,z);
65     th_3= th_ik(3,z);
66     % [th_1,th_2,th_3]
67     T01 = symDH(th_1, d_1, 0, -pi/2); % Create homogenous
68     transform for joint angle 1 for EACH joint configuration.
69     T12 = symDH(th_2-pi/2, 0, a_2, pi);
70     T23 = symDH(th_3-pi/2, 0, 0,-pi/2);
71     %
72
73     T02=T01*T12;
74     T03=T02*T23; % Create homogenous transform for first 3 joint
75     angles for EACH joint configuration.
76
77     Twrist=( [T03(1:3,1:3).', -T03(1:3,1:3).'*T03(1:3,4); 0 0 0 1] ) * T06;
78     % Remove first 3 joint angles to isolate joints
79     % $\theta_4$, $\theta_5$, $\theta_6$
80
81     th_ik(4,z)=atan2(-Twrist(2,3),-Twrist(1,3));
82     th_ik(7,z)=th_ik(7,z)+2^2; %Wrist UP
83     th_ik(5,z)=acos(Twrist(3,3)); %
84     th_ik(6,z)=atan2(-Twrist(3,2),Twrist(3,1));
85
86     th_ik(4,z+2)=atan2(Twrist(2,3),Twrist(1,3));
87     %th_ik(7,z+2)=th_ik(7,z+2)+0; %Wrist DOWN
88     th_ik(5,z+2)=-acos(Twrist(3,3)); %
89     th_ik(6,z+2)=atan2(Twrist(3,2),-Twrist(3,1));
90 end

```

```

82
83 %%
84 % Joint limits and find closest to previous
85 counter=0;
86 temp_out=zeros(7,8);
87 for x=1:8 %Joint limit check
88     if sum(abs(th_ik(1:6,x)) <=th_limit)==6
89         counter=counter+1;
90         temp_out(1:7,counter)=th_ik(1:7,x);
91     end
92 end
93     temp_out_lim=temp_out(1:7,1:counter); % Keep solutions within
        Joint limits
94
95 if max(abs(th_p))>0 % Check for existance of th_p (theta
    previous)
96     min_diff=zeros(counter,1);
97     for x=1:counter %Find closest solution to th_p (theta
        previous)
98         min_diff(x)=norm(temp_out_lim(1:6,x).'-th_p); %
            Distance of each solution
99
100     end
101     th_out=temp_out_lim(:,min_diff==min(min_diff)); % Use closest solution
102 else
103     th_out=temp_out_lim; % Use all solutions with joint limits
104 end
105
106
107
108
109 end
110
111 function T = symDH(th,d,a,al)
112 % homogenous transform matrix from DH parameters
113 T=[ cos(th) -sin(th)*cos(al) sin(th)*sin(al)
    a*cos(th);...
114 sin(th) cos(th)*cos(al) -cos(th)*sin(al)
    a*sin(th);...
115 0 sin(al)
    cos(al) d;
116 0 0 0
        1];
117 end
118
119 function [theta1,theta2]=subproblem2(k1,k2,p,q)
120 % Finds theta1 & theta2 where two vectors p & q intersect when rotated about
    vectors k1 and k2
121 k12=k1'*k2;
122 pk=p'*k2;

```

```

123 qk=q'*k1;
124
125 % check if solution exists
126
127 if abs(k12^2-1)<eps; theta1=[NaN NaN]; theta2=[NaN NaN];
128     %disp('no solution (***1***) ');
129     return;
130 end
131
132 a=[k12 -1;-1 k12]*[pk;qk]/(k12^2-1);
133
134 bb=(norm(p)^2-norm(a)^2-2*a(1)*a(2)*k12);
135 if abs(bb)<eps;bb=0;end
136 if bb<0;theta1=[NaN NaN]; theta2=[NaN NaN];
137 %     disp('no solution (***2***) ');
138     return;
139 end
140
141 % check if there is only 1 solution
142 gamma=sqrt(bb)/norm(cross(k1,k2));
143 if abs(gamma)<eps;
144     c1=[k1 k2 cross(k1,k2)]*[a;gamma];
145     theta2=[subproblem1(k2,p,c1) NaN];
146     theta1=[-subproblem1(k1,q,c1) NaN];
147 %     disp('One solution ');
148     return
149 end
150
151 % general case: 2 solutions
152
153 theta1=zeros(1,2);
154 theta2=zeros(1,2);
155
156 c1=[k1 k2 cross(k1,k2)]*[a;gamma];
157 c2=[k1 k2 cross(k1,k2)]*[a;-gamma];
158 theta2(1)=subproblem1(k2,p,c1);
159 theta2(2)=subproblem1(k2,p,c2);
160
161 theta1(1)=-subproblem1(k1,q,c1);
162 theta1(2)=-subproblem1(k1,q,c2);
163 end
164
165 function [theta]=subproblem1(k,p,q)
166 % [theta]=subproblem1(k,p,q)
167 % 1 unique solution
168
169 k=k/norm(k);
170 pp=p-(p.'*k)*k;
171 qp=q-(q.'*k)*k;
172

```

```

173 epp=pp/norm(pp);
174 eqp=qp/norm(qp);
175
176 theta=subproblem0(epp,eqp,k);
177 end
178
179 function [theta]=subproblem0(p,q,k)
180 % [theta]=subproblem0(p,q,k)
181 % 1 unique solution
182
183
184 pp=p/norm(p);
185 qp=q/norm(q);
186
187 theta=2*atan2(norm(pp-qp),norm(pp+qp));
188
189 if k*(cross(p,q))<0
190     theta=-theta;
191 end
192 end

```

B. CIRCULAR POINT ANALYSIS MATLAB CODE

```

1 function powerball_kinematics_ID_cpa_ros()
2 close all
3 tic
4 %% Symbolic computation of fwd kin
5 % syms th_0 d_0 a_0 al_0; T00=symDH(th_0, d_0, a_0, al_0);
6 % th = sym('th_%d', [6 1]);
7 % th0 = sym('th0_%d', [6 1]);
8 % d = sym('d_%d', [6 1]);
9 % a = sym('a_%d', [6 1]);
10 % al = sym('al_%d', [6 1]);
11 % s_pi2 = sym(pi/2);
12 % T01 = symDH(th(1)+th0(1), d(1), a(1), al(1))
13 % T12 = symDH(th(2)+th0(2), d(2), a(2), al(2))
14 % T23 = symDH(th(2)+th0(3), d(3), a(3), al(3))
15 % T34 = symDH(th(2)+th0(4), d(4), a(4), al(4))
16 % T45 = symDH(th(5)+th0(5), d(5), a(5), al(5))
17 % T56 = symDH(th(6)+th0(6), d(6), a(6), al(6))
18 % T06 = (T01 * T12 * T23 * T34 * T45 * T56);
19 % T01=rewrite(T01, 'sincos');
20 % T02=rewrite(T01*T12, 'sincos');
21 % T03=rewrite(T02*T23, 'sincos');
22 % T04=rewrite(T03*T34, 'sincos');
23 % T05=rewrite(T04*T45, 'sincos');
24 % T06=rewrite(T05*T56, 'sincos');
25
26 % NOMINAL *****
27 % T01 = symDH(th(1), d(1), 0, s_pi2);

```

```

28 % T12 = symDH(th(2)+s_pi2, 0, a(2), 0);
29 % T23 = symDH(th(3)+s_pi2, 0, 0, s_pi2);
30 % T34 = symDH(th(4), d(4), 0, s_pi2);
31 % T45 = symDH(th(5)+pi, 0, 0, s_pi2);
32 % T56 = symDH(th(6), d(6), 0, 0);
33
34 %% Nominal values
35 d_1=.205;
36 a_2=.350;
37 d_4=.305;
38 d_6=.075;
39 m_d=400; m_a=3*pi/2; % Max distance, max angle
40 DH0=[0 d_1 0 pi/2; ...
41      pi/2 0 a_2 0; ...
42      pi/2 0 0 pi/2; ...
43      0, d_4, 0, pi/2;...
44      pi, 0, 0, pi/2;...
45      0, d_6, 0, 0];
46
47 pose_import
48
49 %% Extrapolation of orientation
50 % vr_mark=[0;0; .1];
51 % for i=1:length(time_p)
52 %     theta=2*atan2(norm([xq(i),yq(i),zq(i)]),w(i));
53 %     vr_pos=expm(hat([xq(i) yq(i) zq(i)])*theta)*vr_mark;
54 %     x2(i)=x(i)+vr_pos(1);
55 %     y2(i)=y(i)+vr_pos(2);
56 %     z2(i)=z(i)+vr_pos(3);
57 % end
58 % x=[x x2.'];
59 % y=[y y2.'];
60 % z=[z z2.'];
61
62
63 starts_t=(4.25+[1:10:6*10]); stops_t=(14.25+[1:10:6*10]);
64 for i=1:6
65     starts_1(i)=find(time_p<=starts_t(i),1,'last');
66     stops_1(i)=find(time_p<=stops_t(i),1,'last');
67 end
68
69
70 plane_normal_out1=(-plane_normal(x([starts_1(1):stops_1(1)
    ],1),y([starts_1(1):stops_1(1) ],1),z([starts_1(1):stops_1(1) ],1)
    ,1,1)).';
71 center_point_out1=(circle_point(plane_normal_out1,x([starts_1(1):stops_1(1)
    ],1),y([starts_1(1):stops_1(1) ],1),z([starts_1(1):stops_1(1) ],1)
    ,1,1)).';
72 z_cam=[0;0;-1];
73 n_cam=(hat(z_cam)*plane_normal_out1)/norm(cross(z_cam,plane_normal_out1)) ;

```

```

74 theta_cam=acos(sum(z_cam.*plane_normal_out1));
75 R_cam=expm(hat(n_cam)*theta_cam) .';
76 axis_offset=R_cam*center_point_out1;
77
78 plane_normal_out2=(-plane_normal(x([starts_1(2):stops_1(2)
    ],1),y([starts_1(2):stops_1(2) ],1),z([starts_1(2):stops_1(2) ],1)
    ,2,1)).';
79 center_point_out2=(circle_point(plane_normal_out1,x([starts_1(2):stops_1(2)
    ],1),y([starts_1(2):stops_1(2) ],1),z([starts_1(2):stops_1(2) ],1)
    ,2,1)).';
80
81
82
83 %% Change Coordinate system to align axis 1 with Z
84 axis_offset(3)= 0.0345275294433222-.205; % difference between nominal value and
    current distance between axes 1 & 2
85 for i=1:size(x,2)
86     temp_pos=R_cam*[x(:,i) .';y(:,i) .';z(:,i) .'];
87     x(:,i)=(temp_pos(1,:)-axis_offset(1)).';
88     y(:,i)=(temp_pos(2,:)-axis_offset(2)).';
89     z(:,i)=(temp_pos(3,:)-axis_offset(3)).';
90 end
91 theta_1_offset=-atan2( y(starts_1(1)),x(starts_1(1))); % thetal offset
92 R_theta_1_offset=expm( hat([0;0;1;]) *theta_1_offset); % thetal offset rotation
    matrix
93
94 for i=1:size(x,2) % Align data with thetal offset
95     temp_pos=R_theta_1_offset*[x(:,i) .';y(:,i) .';z(:,i) .'];
96     x(:,i)=temp_pos(1,:) .';
97     y(:,i)=temp_pos(2,:) .';
98     z(:,i)=temp_pos(3,:) .';
99 end
100
101
102 close all
103
104 %% Calculate each axis and angle
105 for j=1:size(x,2) % # of
106     j
107     for i=1:6 % # of
108
109         plane_normal_out=plane_normal(x([starts_1(i):stops_1(i)
            ],j),y([starts_1(i):stops_1(i)
            ],j),z([starts_1(i):stops_1(i) ],j) ,i,i);
110
111         center_point_out=circle_point(plane_normal_out,x([starts_1(i):stops_1(i)
            ],j),y([starts_1(i):stops_1(i)
            ],j),z([starts_1(i):stops_1(i) ],j) ,i,i);
112
113         %

```

```

        center_point_out=R_cam*center_point_out.'+[ axis_offset(1)
        axis_offset(2) 0].';
114
        %
        plane_normal_out=R_cam*plane_normal_out.'+[ axis_offset(1)
        axis_offset(2) 0].';
115
        joint_axes(i,1:6,j)=[center_point_out, plane_normal_out];
116
117
118     end
119
120 end
121
122 legend([{'axis 1 data'},{'Z axis 1'},{'axis 2 data'},{'Z axis 2'},{'axis 3
        data'},{'Z axis 3'},{'axis 4 data'},{'Z axis 4'},{'axis 5 data'},{'Z axis
        5'},{'axis 6 data'},{'Z axis 6'}]);
123 joint_axes(:, :, 1)
124
125 %% distances between axes
126 line_style=[ 'or'; 'sg'; '^b'; 'om'; 'sk'; '^c'];
127 for i=1:5
128     center_point_out1=joint_axes(i,1:3); plane_normal_out1=joint_axes(i,4:6);
129     center_point_out2=joint_axes(i+1,1:3);
        plane_normal_out2=joint_axes(i+1,4:6);
130     n_zdist=cross(plane_normal_out2,plane_normal_out1);
        n_zdist=n_zdist/norm(n_zdist);
131     dist_vect=center_point_out2-center_point_out1;
132     dist_1=center_point_out1 -
        dot(cross(center_point_out2-center_point_out1,plane_normal_out2),n_zdist)/dot(n_zd
133     dist_2=center_point_out2 -
        dot(cross(center_point_out2-center_point_out1,plane_normal_out1),n_zdist)/dot(n_zd
134
135     norm(dist_2-dist_1)
136
137     joint_offsets(i,1:6)=[dist_1 dist_2];
138
139 %     plot3(dist_1(1),dist_1(2),dist_1(3),[ line_style(i,1)
        'k'],dist_2(1),dist_2(2),dist_2(3),[ line_style(i,1)
        'k'],'MarkerFaceColor',line_style(i,2), 'MarkerSize', 9)
140
141 end
142 i=3;
143     center_point_out1=joint_axes(3,1:3); plane_normal_out1=joint_axes(3,4:6);
144     center_point_out2=joint_axes(5,1:3); plane_normal_out2=joint_axes(5,4:6);
145     n_zdist=cross(plane_normal_out2,plane_normal_out1);
        n_zdist=n_zdist/norm(n_zdist);
146     dist_vect=center_point_out2-center_point_out1;
147     dist_1=center_point_out1 -
        dot(cross(center_point_out2-center_point_out1,plane_normal_out2),n_zdist)/dot(n_zd
148     dist_2=center_point_out2 -
        dot(cross(center_point_out2-center_point_out1,plane_normal_out1),n_zdist)/dot(n_zd

```

```

149
150         norm(dist_2-dist_1)
151         [dist_1 dist_2]
152
153 %         plot3(dist_1(1),dist_1(2),dist_1(3),[ line_style(i,1)
'k'],dist_2(1),dist_2(2),dist_2(3),[ line_style(i,1)
'k'],'MarkerFaceColor',line_style(i,2), 'MarkerSize', 9)
154
155
156 joint_offsets
157
158 figure(321); cla; hold on; plot3(x,y,z, 'b'); plot3([0 1]*x(1),[0 1]*y(1),[0
1]*z(1),'+-k','LineWidth',1.25); title('Track data'); grid on; view(3); axis
equal;
159
160 figure(322); clf;
161 px(1)=subplot(3,1,1);
162 plot(time_p,x, 'r', 'LineWidth',2); grid on;
163 ylabel('x (m)', 'FontSize',font_size); title('End Effector Trajectory in
Time', 'FontSize',font_size);
164
165 px(2)=subplot(3,1,2);
166 plot(time_p,y, 'g', 'LineWidth',2); grid on;
167 ylabel('y (m)', 'FontSize',font_size);
168
169 px(3)=subplot(3,1,3);
170 plot(time_p,z, 'b', 'LineWidth',2); grid on;
171 ylabel('z (m)', 'FontSize',font_size); xlabel('time (s)', 'FontSize',font_size);
172
173 linkaxes(px, 'x');
174
175 whos x y z time_p start_time_p
176 % save('pose_data_2013-12-07T18-42-12_aligned', 'x','y','z', 'time_p',
'start_time_p')
177
178
179 toc
180
181 return
182
183
184 function T06_out=end_effector_error(DH_param)
185 %% ERROR function
186 % DH_param = []
187 % T06_in = [4x4xn]
188 % th = [6xn]
189 load('powerball_ID_gen.mat')
190 th=[th_1.' th_2.' th_3.' th_4.' th_5.' th_6.'];
191 T06_out=zeros(4,4, size(th,1));
192 for x=1:size(th,1)

```

```

193         T06_out(1:4,1:4,x)=end_effector(DH_param, th(x,:));
194
195         %           error_out(1:4,1:4,x)=(   T06_out-T06(1:4,1:4,x)   ).^2;
196
197     end
198
199     return
200
201     function T06_out=end_effector(DH_param, th)
202     %% End effector function
203
204     % DH_param=[]
205     % th = [6x1]
206
207     th_1=th(1);th_2=th(2);th_3=th(3);th_4=th(4); th_5=th(5);th_6=th(6);
208     [th0_1]=DH_param(1,1); [d_1]=DH_param(1,2); [a_1]=DH_param(1,3);
209         [a1_1]=DH_param(1,4);
210     [th0_2]=DH_param(2,1);
211         [d_2]=DH_param(2,2);[a_2]=DH_param(2,3);[a1_2]=DH_param(2,4);
212     [th0_3]=DH_param(3,1);
213         [d_3]=DH_param(3,2);[a_3]=DH_param(3,3);[a1_3]=DH_param(3,4);
214     [th0_4]=DH_param(4,1);
215         [d_4]=DH_param(4,2);[a_4]=DH_param(4,3);[a1_4]=DH_param(4,4);
216     [th0_5]=DH_param(5,1);
217         [d_5]=DH_param(5,2);[a_5]=DH_param(5,3);[a1_5]=DH_param(5,4);
218     [th0_6]=DH_param(6,1);
219         [d_6]=DH_param(6,2);[a_6]=DH_param(6,3);[a1_6]=DH_param(6,4);
220
221     %%
222     T01 =...
223     [ cos(th0_1 + th_1), -sin(th0_1 + th_1)*cos(a1_1), sin(th0_1 +
224         th_1)*sin(a1_1), a_1*cos(th0_1 + th_1);...
225     sin(th0_1 + th_1), cos(th0_1 + th_1)*cos(a1_1), -cos(th0_1 +
226         th_1)*sin(a1_1), a_1*sin(th0_1 + th_1);...
227     0, sin(a1_1), cos(a1_1),
228         d_1;...
229     0, 0, 0,
230         1];
231
232     T12 =...
233     [ cos(th0_2 + th_2), -sin(th0_2 + th_2)*cos(a1_2), sin(th0_2 +
234         th_2)*sin(a1_2), a_2*cos(th0_2 + th_2);...
235     sin(th0_2 + th_2), cos(th0_2 + th_2)*cos(a1_2), -cos(th0_2 +
236         th_2)*sin(a1_2), a_2*sin(th0_2 + th_2);...
237     0, sin(a1_2), cos(a1_2),
238         d_2;...
239     0, 0, 0,
240         1];
241
242

```

```

229
230 T23 =...
231 [ cos(th0_3 + th_3), -sin(th0_3 + th_3)*cos(al_3), sin(th0_3 +
    th_3)*sin(al_3), a_3*cos(th0_3 + th_3) ; ...
232 sin(th0_3 + th_3), cos(th0_3 + th_3)*cos(al_3), -cos(th0_3 +
    th_3)*sin(al_3), a_3*sin(th0_3 + th_3) ; ...
233 0, sin(al_3),
    cos(al_3), d_3 ; ...
234 0, 0,
    0,
    1];
235
236
237 T34 =...
238 [ cos(th0_4 + th_4), -sin(th0_4 + th_4)*cos(al_4), sin(th0_4 +
    th_4)*sin(al_4), a_4*cos(th0_4 + th_4) ; ...
239 sin(th0_4 + th_4), cos(th0_4 + th_4)*cos(al_4), -cos(th0_4 +
    th_4)*sin(al_4), a_4*sin(th0_4 + th_4) ; ...
240 0, sin(al_4),
    cos(al_4), d_4 ; ...
241 0, 0,
    0,
    1];
242
243
244 T45 =...
245 [ cos(th0_5 + th_5), -sin(th0_5 + th_5)*cos(al_5), sin(th0_5 +
    th_5)*sin(al_5), a_5*cos(th0_5 + th_5);...
246 sin(th0_5 + th_5), cos(th0_5 + th_5)*cos(al_5), -cos(th0_5 +
    th_5)*sin(al_5), a_5*sin(th0_5 + th_5);...
247 0, sin(al_5), cos(al_5),
    d_5;...
248 0, 0, 0,
    1];
249
250
251 T56 =...
252 [ cos(th0_6 + th_6), -sin(th0_6 + th_6)*cos(al_6), sin(th0_6 +
    th_6)*sin(al_6), a_6*cos(th0_6 + th_6);...
253 sin(th0_6 + th_6), cos(th0_6 + th_6)*cos(al_6), -cos(th0_6 +
    th_6)*sin(al_6), a_6*sin(th0_6 + th_6);...
254 0, sin(al_6), cos(al_6),
    d_6;...
255 0, 0, 0,
    1];
256
257 T06_out = (T01 * T12 * T23 * T34 * T45 * T56);
258
259
260 return

```

```

261
262
263 function T = symDH(th,d,a,al)
264 %% Homogeneous transform
265 T=[      cos(th)          -sin(th)*cos(al)          sin(th)*sin(al)
      a*cos(th);...
266      sin(th)          cos(th)*cos(al)          -cos(th)*sin(al)
      a*sin(th);...
267      0          sin(al)
      cos(al)          d;
268      0          0
              0
              1];

269 return
270
271 function khat = hat(k)
272 khat=[0 -k(3) k(2); k(3) 0 -k(1); -k(2) k(1) 0];
273 return
274
275 function center_point_out=circle_point(plane_normal_out,x,y,z,i,j)
276
277 if size(plane_normal_out,1)~=1; plane_normal_out=plane_normal_out.'; end
278 %- 2*z*z0 + x^2 - 2*x*x0 + x0^2 + y^2 - 2*y*y0 + y0^2 + z^2 + z0^2 - r^2
279 % for i=1:length(x)
280 %
281 % end
282 x=squeeze(x); y=squeeze(y); z=squeeze(z);
283 if size(x,2)>1; x=x.'; y=y.'; z=z.'; end
284 y(isnan(x))=[]; z(isnan(x))=[]; x(isnan(x))=[];
285 y(isnan(z))=[]; x(isnan(z))=[]; z(isnan(z))=[];
286 z(isnan(y))=[]; x(isnan(y))=[]; y(isnan(y))=[];
287
288 % m=length(x);
289 % P=[x,y,z];
290 plane_normal_out = plane_normal_out/norm(plane_normal_out); % <-- do this if N
    is not normalized
291 Q=[mean(x) mean(y) mean(z)];
292 N2 = plane_normal_out.'*plane_normal_out;
293 % P0 = P*(eye(3)-N2)+ repmat(Q*N2,m,1);
294 % x=P0(:,1); y=P0(:,2); z=P0(:,3);
295
296 A=[2*x, 2*y, 2*z, ones(length(x),1)];
297 q=x.^2+y.^2+z.^2;
298 center_point_out=pinv(A)*q;
299
300 center_point_out = center_point_out(1:3).'*(eye(3)-N2)+Q*N2;
301
302 font_size=15;
303 line_style=['or'; 'sg'; '^b'; 'om'; 'sk'; '^c'];
304 plane_normal_out =plane_normal_out *.25;
305 % figure(300+i); hold on; plot3(x,y,z,['-' line_style(j,2)]); grid on;

```

```

        xlabel('X (m)', 'FontSize', font_size); ylabel('Y (m)', 'FontSize', font_size);
        ylabel('Z (m)', 'FontSize', font_size); title(['Circle Center & Normal: axis '
        num2str(i) ], 'FontSize', font_size); hold on; %plot3(P0(:,1),P0(:,2),
        P0(:,3), 'or');
306 % plot3(center_point_out(1),center_point_out(2),center_point_out(3),['p'
        line_style(j,2)], [0 plane_normal_out(1)]+center_point_out(1), [0
        plane_normal_out(2)]+center_point_out(2), [0
        plane_normal_out(3)]+center_point_out(3), ['-p'
        line_style(j,2)], 'MarkerFaceColor', line_style(j,2), 'MarkerSize', 9);
        view(3); axis equal;
307
308 figure(400); hold on; ax=gca; plot3(ax,x,y,z,['-' line_style(j,2)]); grid on;
        xlabel('X (m)', 'FontSize', font_size); ylabel('Y (m)', 'FontSize', font_size);
        ylabel('Z (m)', 'FontSize', font_size); title(['Joint Z Axes and Data using
        CPA' ], 'FontSize', font_size); hold on; %plot3(P0(:,1),P0(:,2), P0(:,3), 'or');
309 plot3([0 plane_normal_out(1)]+center_point_out(1), [0
        plane_normal_out(2)]+center_point_out(2), [0
        plane_normal_out(3)]+center_point_out(3), ['-p'
        line_style(j,2)], 'MarkerFaceColor', line_style(j,2), 'MarkerSize', 9);
        view(3); axis equal;
310
311 return
312
313 function plane_normal_out=plane_normal(x,y,z,i,j)
314 x=squeeze(x); y=squeeze(y); z=squeeze(z);
315 if size(x,2)>1; x=x.'; y=y.'; z=z.'; end
316
317 y(isnan(x))=[]; z(isnan(x))=[]; x(isnan(x))=[];
318 y(isnan(z))=[]; x(isnan(z))=[]; z(isnan(z))=[];
319 z(isnan(y))=[]; x(isnan(y))=[]; y(isnan(y))=[];
320
321 center_point_out=[mean(x),mean(y),mean(z)];
322 min_xyz=[max(x)-min(x) max(y)-min(y) max(z)-min(z)];
323
324 min_range=find(min_xyz==min(min_xyz));
325
326 max_dir=.25;
327
328 switch min_range
329     case 1
330         nx=max_dir;
331         A=[(y - center_point_out(2)) (z - center_point_out(3))];
332         plane_normal_temp=pinv(A)*-(x - center_point_out(1));
333         plane_normal_out=[nx plane_normal_temp.'*nx];
334     case 2
335         ny=max_dir;
336         A=[(x - center_point_out(1)) (z - center_point_out(3))];
337         plane_normal_temp=pinv(A)*-(y - center_point_out(2));
338         plane_normal_out=[plane_normal_temp(1)*ny ny
            plane_normal_temp(2)*ny];

```

```

339         case 3
340             nz=max_dir;
341             A=[(x - center_point_out(1)) (y - center_point_out(2))];
342             plane_normal_temp=pinv(A)*-(z - center_point_out(3));
343             plane_normal_out=[plane_normal_temp.'*nz nz];
344         end
345
346         % (nx*(x - x0) + ny*(y - y0) )/nz== (z - z0)
347
348         line_style=['or'; 'sg'; '^b'; 'om'; 'sk'; '^c'];
349
350         plane_normal_out = plane_normal_out/norm(plane_normal_out); % <-- normalize
            normal vector
351         plane_normal_out =plane_normal_out *max_dir;
352         %
353         % figure(300+i); hold on; plot3(x,y,z,['-' line_style(j,2)]); grid on;
            xlabel('X (m)'); ylabel('Y (m)'); zlabel('Z (m)'); title(['PLANE: axis '
            num2str(i) ]); hold on;
354         % plot3([0 plane_normal_out(1)],[0 plane_normal_out(2)],[0
            plane_normal_out(3)],['-p'
            line_style(j,2)], 'MarkerFaceColor',line_style(j,2)); view(3); axis equal;
355         %
356         % figure(400); hold on; plot3(x,y,z,['-' line_style(j,2)]); grid on;
            xlabel('X (m)'); ylabel('Y (m)'); zlabel('Z (m)'); title(['PLANE: axis '
            num2str(i) ]); hold on;
357         % plot3([0 plane_normal_out(1)],[0 plane_normal_out(2)],[0
            plane_normal_out(3)],['-p'
            line_style(j,2)], 'MarkerFaceColor',line_style(j,2)); view(3); axis equal;
358
359         plane_normal_out = plane_normal_out/norm(plane_normal_out); % <-- normalize
            normal vector
360         return

```

C. GAUSS-NEWTON MINIMIZATION MATLAB CODE

```

1 function powerball_kinematics_ID_gauss()
2 close all
3
4 %% Symbolic computation of fwd kin
5 % syms th_0 d_0 a_0 al_0; T00=symDH(th_0, d_0, a_0, al_0);
6 th = sym('th_%d', [6 1]);
7 th0 = sym('th0_%d', [6 1]);
8 d = sym('d_%d', [6 1]);
9 a = sym('a_%d', [6 1]);
10 al = sym('al_%d', [6 1]);
11 th=sym(th, 'positive');
12 th0=sym(th0, 'positive');
13 d=sym(d, 'positive');
14 a=sym(a, 'positive');
15 al=sym(al, 'positive');

```

```

16
17 % d_1=.205;
18 % d_2=0;
19
20 % d([1,2,3, 5])=[.205, 0, 0, 0];
21 % a([1 3:6])=[0 0 0 0 0];
22 % phi_sym=[th0; d([4,6]); a([2]); al]
23 phi_sym=[th0; d; a; al];
24
25 % a_2=*;
26 % d_4=*;
27 % d_6=*;
28
29 T01 = symDH(th(1)+th0(1), d(1), a(1), al(1))
30 T12 = symDH(th(2)+th0(2), d(2), a(2), al(2))
31 T23 = symDH(th(3)+th0(3), d(3), a(3), al(3))
32 T34 = symDH(th(4)+th0(4), d(4), a(4), al(4))
33 T45 = symDH(th(5)+th0(5), d(5), a(5), al(5))
34 T56 = symDH(th(6)+th0(6), d(6), a(6), al(6))
35 % T06 = (T01 * T12 * T23 * T34 * T45 * T56);
36 T01=rewrite(T01, 'sincos');
37 T02=rewrite(T01*T12, 'sincos');
38 T03=rewrite(T02*T23, 'sincos');
39 T04=rewrite(T03*T34, 'sincos');
40 T05=rewrite(T04*T45, 'sincos');
41 T06=rewrite(T05*T56, 'sincos');
42
43
44 % J=jacobian(T06(1:3,4),phi_sym);
45
46 for i=1:length(phi_sym) % Create DH parameter Jacobian
47     J(1:3,i)=[diff(T06(1:3,4),phi_sym(i))];
48 end
49
50
51
52 % NOMINAL *****
53 % T01 = symDH(th(1), d(1), 0, s_pi2);
54 % T12 = symDH(th(2)+s_pi2, 0, a(2), 0);
55 % T23 = symDH(th(3)+s_pi2, 0, 0, s_pi2);
56 % T34 = symDH(th(4), d(4), 0, s_pi2);
57 % T45 = symDH(th(5)+pi, 0, 0, s_pi2);
58 % T56 = symDH(th(6), d(6), 0, 0);
59
60 %% Initial guess
61 d_1=.205;
62 a_2=0.319145526760135; %.350;
63 d_4= 0.299986258998911; %.305;
64 d_6=0.213876304123098 +.04; %.075;
65 % m_d=400; m_a=3*pi/2; % Max distance, max angle

```

```

66 DH0=[0          d_1          0          -pi/2; ...
67      -pi/2      0          a_2          pi; ...
68      -pi/2      0          0          -pi/2; ...
69      0,          d_4,      0,          pi/2;...
70      0,          0,          0,          -pi/2;...
71      0,          d_6,      0.100240233652924 ,      0];
72
73 % DH0=ones(size(DH0)); % BAD Starting position (works
    if d_2 & d_3 ==0 to account for parallel axes)
74 % DH0=zeros(size(DH0)); % BAD Starting position (works
    if d_2 & d_3 ~=0)
75
76 phi=[DH0(:,1); DH0(:,2); DH0(:,3); DH0(:,4)];
77 % phi=[DH0(:,1); DH0([4,6],2); DH0(2,3); DH0(:,4)];
78
79 %% Data import
80 joint_import; clear th_*d
81 load('pose_data_2013-12-07T18-42-12_aligned.mat')
82
83
84 %% Align data start time and stop times
85 start_times=[start_time_p, start_time_j];
86 start_time_idx=find(start_times==max(start_times));
87
88 % Start *****
89 if start_time_idx==1
90     time_j=time_j-(start_time_p-start_time_j);
91     th_1(time_j<0)=[]; th_2(time_j<0)=[]; th_3(time_j<0)=[];
        th_4(time_j<0)=[]; th_5(time_j<0)=[]; th_6(time_j<0)=[];
        time_j(time_j<0)=[];
92 else
93     time_p=time_p-3; %(start_time_j-start_time_p); %Hand calculated because
        there time is a time mismatch between the ROS times
94     x(time_p<0)=[]; y(time_p<0)=[]; z(time_p<0)=[]; time_p(time_p<0)=[];
95 end
96
97 % Stop *****
98 stop_times=[time_p(end), time_j(end)];
99 start_time_idx=find(stop_times==max(stop_times));
100 if start_time_idx==1
101     stop_time_idx=find(time_p<=time_j(end),1,'last');
102     time_p(stop_time_idx:end)=[]; x(stop_time_idx:end)=[];
        y(stop_time_idx:end)=[]; z(stop_time_idx:end)=[];
103 else
104     stop_time_idx=find(time_j<=time_p(end),1,'last');
105     time_j(stop_time_idx:end)=[]; th_1(stop_time_idx:end)=[];
        th_2(stop_time_idx:end)=[]; th_3(stop_time_idx:end)=[];
        th_4(stop_time_idx:end)=[]; th_5(stop_time_idx:end)=[];
        th_6(stop_time_idx:end)=[];
106 end

```

```

107
108 for i=1:length(time_p)           % if pose data is shorter vectors will be
    minimum length; if joint data is minimum vectors will be maximum length
109     time_diff=abs(time_p(i)-time_j);
110     joint_index=find(time_diff==min(time_diff));
111     th_1n(i)=th_1(joint_index); th_2n(i)=th_2(joint_index);
        th_3n(i)=th_3(joint_index); th_4n(i)=th_4(joint_index);
        th_5n(i)=th_5(joint_index); th_6n(i)=th_6(joint_index);
112     time_jn(i)=time_j(joint_index);
113 end
114
115 th_1=th_1n.'; th_2=th_2n.'; th_3=th_3n.'; th_4=th_4n.'; th_5=th_5n.';
    th_6=th_6n.';
116 time_j=time_jn.'; clear th_*n time_jn
117
118 %% Throw out data with bad timing delays
119 bad_time_temp=abs(time_j-time_p);
120 bad_time_threshold=1e-4; %2e-4;
121 th_1(bad_time_temp>bad_time_threshold) = [];
    th_2(bad_time_temp>bad_time_threshold) = [];
    th_3(bad_time_temp>bad_time_threshold) = [];
    th_4(bad_time_temp>bad_time_threshold) = [];
    th_5(bad_time_temp>bad_time_threshold) = [];
    th_6(bad_time_temp>bad_time_threshold) = [];
    time_j(bad_time_temp>bad_time_threshold) = [];
    time_p(bad_time_temp>bad_time_threshold) = [];
    x(bad_time_temp>bad_time_threshold) = [];
    y(bad_time_temp>bad_time_threshold) = [];
    z(bad_time_temp>bad_time_threshold) = [];
122
123 bad_time_temp=(time_j<=2 | time_j>=61.5); % trim pre and post
124 th_1(bad_time_temp) = []; th_2(bad_time_temp) = []; th_3(bad_time_temp) = [];
    th_4(bad_time_temp) = []; th_5(bad_time_temp) = []; th_6(bad_time_temp) = [];
    time_j(bad_time_temp) = []; time_p(bad_time_temp) = []; x(bad_time_temp) = [];
    y(bad_time_temp) = []; z(bad_time_temp) = [];
125
126 %% Initial Data check
127 tic
128 for i=1:length(th_1)
129     T06_out=end_effector_mex(DH0, [th_1(i) th_2(i) th_3(i) th_4(i) th_5(i)
        th_6(i)]);
130     x_check(i)=T06_out(1,4);
131     y_check(i)=T06_out(2,4);
132     z_check(i)=T06_out(3,4);
133
134 end
135
136 disp(['Data check: ' num2str(toc)]);
137 font_size=15;
138 figure(210); clf;

```

```

139 plot3(x_check,y_check,z_check,'r'); hold on;
140 grid on; %axis equal;
141 xlabel('x (m)', 'FontSize',font_size); ylabel('y (m)', 'FontSize',font_size);
    ylabel('z (m)', 'FontSize',font_size); title('3D View of End Effector
    Path', 'FontSize',font_size);
142
143 figure(212); clf;
144 px(1)=subplot(3,1,1);
145 plot(time_j,x_check,'om'); hold on;
146 % ylabel('x (m)', 'FontSize',font_size); title('Joint Based Trajectory VS
    Actual', 'FontSize',font_size);
147
148 px(2)=subplot(3,1,2);
149 plot(time_j,y_check,'ok'); hold on;
150 % ylabel('y (m)', 'FontSize',font_size);
151
152 px(3)=subplot(3,1,3);
153 plot(time_j,z_check,'oc'); hold on;
154 % ylabel('z (m)', 'FontSize',font_size); xlabel('time (s)', 'FontSize',font_size);
155
156 linkaxes(px, 'x');
157 toc
158
159 %% Plot results
160 font_size=15;
161 figure(210); %clf;
162 plot3(x,y,z,'b',[0 0],[0 0],[0 1]* 1.04122069939116,'pk');
163 axis equal; grid on;
164 xlabel('x (m)', 'FontSize',font_size); ylabel('y (m)', 'FontSize',font_size);
    ylabel('z (m)', 'FontSize',font_size); title('3D View of End Effector
    Path', 'FontSize',font_size);
165 legend([{'DH from CPA'},{'Aligned Camera Points'}], 'Location',
    'BestOutside', 'FontSize',10); legend('boxoff')
166
167 figure(212); %clf;
168 px(1)=subplot(3,1,1);
169 plot(time_p,x,'.r');
170 ylabel('x (m)', 'FontSize',font_size); title('Joint Based Trajectory VS
    Actual', 'FontSize',font_size);
171
172 px(2)=subplot(3,1,2);
173 plot(time_p,y,'.g');
174 ylabel('y (m)', 'FontSize',font_size);
175
176 px(3)=subplot(3,1,3);
177 plot(time_p,z,'.b');
178 ylabel('z (m)', 'FontSize',font_size); xlabel('time (s)', 'FontSize',font_size);
179
180 linkaxes(px, 'x');
181 set(gcf, 'Position', [804 329 715 604]);

```

```

182
183 figure(200); clf;
184
185 ax(1)=subplot(6,1,1); plot(time_j,th_1,'.b'); title(['Joint Angles_{best sync}
    #pts:' num2str(length(th_1))'],'FontSize',font_size); ylabel('th_1
    (rad)','FontSize',font_size);
186 ax(2)=subplot(6,1,2); plot(time_j,th_2,'.b'); ylabel('th_2
    (rad)','FontSize',font_size);
187 ax(3)=subplot(6,1,3); plot(time_j,th_3,'.b'); ylabel('th_3
    (rad)','FontSize',font_size);
188 ax(4)=subplot(6,1,4); plot(time_j,th_4,'.b'); ylabel('th_4
    (rad)','FontSize',font_size);
189 ax(5)=subplot(6,1,5); plot(time_j,th_5,'.b'); ylabel('th_5
    (rad)','FontSize',font_size);
190 ax(6)=subplot(6,1,6); plot(time_j,th_6,'.b'); ylabel('th_6
    (rad)','FontSize',font_size); xlabel('time (s)','FontSize',font_size);
191
192 linkaxes([ax px],'x');
193
194 % return
195
196 %% Solve DH
197 pose=[x;y;z];
198
199 A1=zeros(length(x),length(phi_sym));
200 A2=A1; A3=A1;
201
202 % for i=1:3; for j=1:length(phi_sym); J_text{i,j}=char(J(i,j)); end; end; whos
    J_text
203 % fid = fopen('J.txt','w');
204 % for i=1:3; for j=1:length(phi_sym); fprintf(fid,['J_out(' num2str(i) ','
    num2str(j) ')= %s ; \r\n'], J_text{i,j}); fprintf(fid,'\r\n'); end; end;
205 % fclose(fid);
206
207 whos
208 tic
209 i=0;
210 phi_change=ones(size(phi_sym)); % Initialize stopping criterion
211 phi_data(1:length(phi),1)=phi;
    % Store Progress Initialize
212
213
214 pose_err_data(1:length(pose),1)=pose-[x_check.'; y_check.'; z_check.'];
    % Store Progress
215 while max(abs(phi_change))>1e-4 && i<=60 %1e-4
216     i=i+1;
217     if rem(i,5)==0; disp(['Number of iterations: ' num2str(i) ', Change: '
        num2str(max(abs(phi_change)))]); end
218
219     for j=1:length(x) % parfor Loop for each data point

```

```

220
221         T06_out=end_effector_mex(subs( [th0, d, a, al],phi_sym,phi),
            [th_1(j) th_2(j) th_3(j) th_4(j) th_5(j) th_6(j) ]);
            %subs(pose_dh,th, [th_1(j) th_2(j) th_3(j) th_4(j) th_5(j)
            th_6(j) ]);
222         pose1(j)=T06_out(1,4);
223         pose2(j)=T06_out(2,4);
224         pose3(j)=T06_out(3,4);
225
226         J_temp=J_eval_mex(subs( [th0, d, a, al],phi_sym,phi), [th_1(j)
            th_2(j) th_3(j) th_4(j) th_5(j) th_6(j) ]);
227         A1(j,:)=J_temp(1,:);
228         A2(j,:)=J_temp(2,:);
229         A3(j,:)=J_temp(3,:);
230
231
232     end
233     A=[A1;A2;A3];
234     pose_calc=[pose1.';pose2.';pose3.'];
235     pose_err=pose-pose_calc;
236
237     phi_delta=pinv(A)*(pose_err);           % Solve least squares problem
238     phi=phi+phi_delta*2/(1+5^.2);
239     phi([1:6,19:24])=mod(phi([1:6,19:24]), 2*pi);           % keep
        angles +/- 2*pi
240 %     phi(8:9)=0;
        % Account for parallel adjacent axes instead of Hayati (not necessary)
241
242     phi_data(1:length(phi),i+1)=phi;
        % Store Progress
243     pose_err_data(1:length(pose),i+1)=pose_err;
        % Store Progress
244
245     phi_change=phi_data(:,i)-phi;
246
247 end% 1e-5
248 disp(['Total Number of iterations: ' num2str(i)]);
249
250 save('phi_data','phi_data','pose_err_data', 'th_1', 'th_2', 'th_3', 'th_4',
        'th_5', 'th_6','x', 'y', 'z', 'time_j', 'time_p');
251
252 phi_plot           % Final Plotting
253
254
255 toc
256
257 return
258
259 function T = symDH(th,d,a,al)
260 %% Homogeneous transform

```

```

261 T=[      cos(th)          -sin(th).*cos(al)      sin(th).*sin(al)
      a.*cos(th);...
262      sin(th)          cos(th).*cos(al)          -cos(th).*sin(al)
      a.*sin(th);...
263      0                  sin(al)
      cos(al)                  d;
264      0                  0
      0                  0
      0                  0
265 return

```

D. JACOBIAN FOR DH PARAMETERS MATLAB CODE

```

1 function J_out=J_eval(DH_param, th) %#codegen
2 %% Jacobian Function for End Effector Position, x,y,z \wrt all 24 DH Parameters
3
4 J_out=zeros(3,24);
5
6 th_1=th(1);th_2=th(2);th_3=th(3);th_4=th(4); th_5=th(5);th_6=th(6);
7 [th0_1]=DH_param(1,1); [d_1]=DH_param(1,2); [a_1]=DH_param(1,3);
  [al_1]=DH_param(1,4);
8 [th0_2]=DH_param(2,1);
  [d_2]=DH_param(2,2);[a_2]=DH_param(2,3);[al_2]=DH_param(2,4);
9 [th0_3]=DH_param(3,1);
  [d_3]=DH_param(3,2);[a_3]=DH_param(3,3);[al_3]=DH_param(3,4);
10 [th0_4]=DH_param(4,1);
  [d_4]=DH_param(4,2);[a_4]=DH_param(4,3);[al_4]=DH_param(4,4);
11 [th0_5]=DH_param(5,1);
  [d_5]=DH_param(5,2);[a_5]=DH_param(5,3);[al_5]=DH_param(5,4);
12 [th0_6]=DH_param(6,1);
  [d_6]=DH_param(6,2);[a_6]=DH_param(6,3);[al_6]=DH_param(6,4);
13
14 J_out(1,1)= d_6*(cos(th0_5 + th_5)*sin(al_5)*(sin(al_4)*(sin(th0_3 +
  th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
  th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
  th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 +
15 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
  cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
  cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
  th_2)*cos(al_1)*cos(al_2))) +
16 sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
  th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
  th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
  th_2)*sin(th0_1 + th_1) +
17 cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
  th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
  sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
  th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
18 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
  th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
  th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)

```

```

- cos(th0_1 + th_1)*cos(th0_2 +
19 th_2)*cos(al_1)*cos(al_2))) - cos(al_5)*(cos(al_4)*(sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 +
20 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) -
21 sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) +
22 cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
23 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 + th_1)*cos(th0_2 +
24 th_2)*cos(al_1)*cos(al_2))) + sin(th0_5 + th_5)*sin(al_5)*(sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
25 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 + th_1)*cos(th0_2 +
26 th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*(cos(th0_2 +
27 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) -
d_5*(cos(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
cos(al_3)*(cos(th0_1 +
28 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) -
29 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) - sin(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) -
30 cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 +
31 th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +

```



```

th_3)*sin(al_3)*(sin(th0_1 +
44 th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 +
45 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 +
46 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) +
47 cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)))) - a_6*cos(th0_6 + th_6)*(sin(th0_5 +
th_5)*(sin(al_4)*(sin(th0_3 +
48 th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 +
49 th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 +
50 th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 +
51 th_2)*cos(al_1))) - cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 +
52 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)))) -
53 cos(th0_5 + th_5)*(sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
54 th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 +
55 th_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 +
56 th_1)*sin(th0_2 + th_2)*cos(al_1)))))) - a_6*sin(th0_6 +

```

```

th_6)*(sin(al_5)*(cos(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
57 sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 +
58 th_2)*cos(al_1)*cos(al_2))) - sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
59 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) +
cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 +
60 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 +
61 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + cos(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
62 cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 +
63 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
64 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 +
65 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 +
66 th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + sin(th0_5 +
th_5)*cos(al_5)*(sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 +
67 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 +
68 th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3
+ th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +

```

```

th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
69 th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))))
- a_2*cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1) ;
70
71 J_out(1,2)= d_3*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) + d_4*(cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2)*sin(al_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*sin(al_2)) +
72 cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1))) -
73 d_6*(cos(al_5)*(sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2) + cos(th0_2 + th_2)*sin(th0_1
74 + th_1)*cos(al_1))) - cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(th0_2
75 + th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) +
cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_2) - sin(th0_1 +
76 th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) +
77 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)))) + sin(th0_5 +
th_5)*sin(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(cos(th0_1 +
78 th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) -
sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_2)
- sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 +
79 th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)))) +
cos(th0_5 +
80 th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(th0_2 +
81 th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) +
sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_2) - sin(th0_1 +
82 th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(cos(th0_1 +

```

```

th_1)*sin(th0_2 + th_2) + cos(th0_2+ th_2)*sin(th0_1 + th_1)*cos(al_1))) +
cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_2) - sin(th0_1 +
83 th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) +
84 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)))) - d_5*(sin(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) +
cos(th0_3 + th_3)*(cos(th0_1 +
85 th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) -
cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 +
86 th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) +
cos(th0_4 +
87 th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 +
88 th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)))) +
a_4*sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_2) - sin(th0_1 +
89 th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) +
90 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) + a_6*sin(th0_6 +
th_6)*(sin(th0_5 + th_5)*cos(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) +
91 cos(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1))) - sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_2) - sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
92 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)))) -
93 sin(al_5)*(sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2) + cos(th0_2 + th_2)*sin(th0_1 +
94 th_1)*cos(al_1))) - cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2)

```

```

sin(th0_1 + th_1)*sin(th0_2 +
95 th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) +
cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_2) - sin(th0_1 +
96 th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) +
97 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)))) + cos(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
98 th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) +
sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
99 th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)))
+ cos(th0_4 +
100 th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 +
101 th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))))
- a_3*cos(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1))
102 - a_2*cos(th0_1 + th_1)*sin(th0_2 + th_2) - a_6*cos(th0_6 + th_6)*(cos(th0_5 +
th_5)*(cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) +
cos(th0_3 +
103 th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1))) - sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2)*sin(al_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
104 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)))) - sin(th0_5 +
105 th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 +
106 th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) +
sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_2) - sin(th0_1 +
107 th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(cos(th0_1 +

```

```

th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) +
cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_2) - sin(th0_1 +
108 th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) +
109 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)))) + a_5*sin(th0_5 +
th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 +
110 th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) +
sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
111 th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)))
+ cos(th0_4 +
112 th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(th0_2 +
113 th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) -
a_3*sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(th0_2 +
114 th_2)*cos(al_1)*cos(al_2)) - a_4*cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2) + cos(th0_2 +
115 th_2)*sin(th0_1 + th_1)*cos(al_1))) - a_5*cos(th0_5 + th_5)*(cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
116 th_2) + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1))) - sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_2) - sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
117 th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) +
sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2) + cos(th0_2
+ th_2)*sin(th0_1 + th_1)*cos(al_1))) - a_2*cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1) ;
118
119 J_out(1,3)= d_4*(cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) +
120 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) - d_6*(sin(th0_5 +
th_5)*sin(al_5)*(cos(th0_4 + th_4)*(cos(th0_3 + th_3)*(cos(th0_1 +

```

```

th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
121 th_1)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + sin(th0_4 +
th_4)*(cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) -
sin(th0_1 + th_1)*sin(th0_2 +
122 th_2)*cos(al_1)) - sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) -
cos(al_5)*(cos(al_4)*(cos(th0_3 +
123 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
+ cos(th0_2 + th_2)*sin(th0_1 +
124 th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 + th_4)*sin(al_4)*(cos(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*(cos(th0_1 +
125 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) +
cos(th0_4 + th_4)*sin(al_4)*(cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(th0_3 +
126 th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)))) + cos(th0_5 +
th_5)*sin(al_5)*(sin(al_4)*(cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
127 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) +
128 sin(th0_4 + th_4)*cos(al_4)*(cos(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(th0_2 + th_2) -
129 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*cos(al_4)*(cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
130 th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + d_5*(cos(al_4)*(cos(th0_3
+ th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) -
131 sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 + th_4)*sin(al_4)*(cos(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
132 th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))
+ cos(th0_4 +

```

```

133 th_4)*sin(al_4)*(cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
    th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
    th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
    th_1)*sin(al_1)*sin(al_2) +
134 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) - a_3*sin(th0_3 +
    th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
    th_2)*cos(al_1)) - a_4*sin(th0_4 + th_4)*(cos(th0_3 +
    th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
135 th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
    th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
    th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*cos(al_1)*cos(al_2))) - a_3*cos(th0_3 +
136 th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
    th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*cos(al_1)*cos(al_2)) + a_6*sin(th0_6 +
    th_6)*(sin(al_5)*(cos(al_4)*(cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
    th_1)*cos(th0_2
137 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
    th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
    th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 +
138 th_4)*sin(al_4)*(cos(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
    th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
    th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*(cos(th0_1
    + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
139 th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 + th_4)*sin(al_4)*(cos(th0_3 +
    th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1
    + th_1)*sin(th0_2 +
140 th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
    th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + sin(th0_5 +
    th_5)*cos(al_5)*(cos(th0_4 + th_4)*(cos(th0_3 + th_3)*(cos(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
141 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2
    + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + sin(th0_4 +
    th_4)*(cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
142 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
    sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
    sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*cos(al_1)*cos(al_2)))) +
143 cos(th0_5 + th_5)*cos(al_5)*(sin(al_4)*(cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
    th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
    sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
    sin(th0_1 +
144 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*cos(al_1)*cos(al_2)))) + sin(th0_4 + th_4)*cos(al_4)*(cos(th0_3 +
    th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
    th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
145 th_1)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 +

```

```

th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*cos(al_4)*(cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 +
146 th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + a_6*cos(th0_6
+ th_6)*(sin(th0_5 +
147 th_5)*(sin(al_4)*(cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) +
148 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 +
th_4)*cos(al_4)*(cos(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) +
149 sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 + th_4)*cos(al_4)*(cos(th0_3
+ th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) -
150 sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)))) - cos(th0_5 + th_5)*(cos(th0_4 +
th_4)*(cos(th0_3 + th_3)*(cos(th0_1 +
151 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) + sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1))) + sin(th0_4 +
152 th_4)*(cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
153 th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + a_5*sin(th0_5 +
th_5)*(sin(al_4)*(cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 +
154 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 +
th_4)*cos(al_4)*(cos(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 +
155 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2
+ th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*cos(al_4)*(cos(th0_3 +
156 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
+ cos(th0_2 + th_2)*sin(th0_1 +
157 th_1)*cos(al_1)*cos(al_2)))) - a_4*cos(th0_4 + th_4)*(cos(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +

```

```

th_1)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*(cos(th0_1 +
158 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
a_5*cos(th0_5 + th_5)*(cos(th0_4 + th_4)*(cos(th0_3 + th_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
159 th_1)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + sin(th0_4 +
th_4)*(cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) -
sin(th0_1 + th_1)*sin(th0_2 +
160 th_2)*cos(al_1)) - sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) ;

161
162 J_out(1,4)= a_6*cos(th0_6 + th_6)*(sin(th0_5 + th_5)*(cos(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) -
163 cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3
+ th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) -
164 sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
165 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)))) - cos(th0_5 + th_5)*(cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
166 sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
167 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 + th_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
168 th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) -
d_6*(cos(al_5)*(cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) -
169 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2
+ th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + sin(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 +
170 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 +
171 th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +

```



```

th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
184 th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + a_6*sin(th0_6 +
th_6)*(cos(th0_5 + th_5)*cos(al_5)*(cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 +
185 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1))) + sin(th0_4 +
186 th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
187 th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) - sin(al_5)*(cos(th0_4 +
188 th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(th0_2 + th_2) - sin(th0_1 +
189 th_1)*sin(th0_2 + th_2)*cos(al_1))) + sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 +
190 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
191 th_1)*cos(al_1)*cos(al_2))) + sin(th0_5 + th_5)*cos(al_5)*(cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) +
192 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
193 th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*(cos(th0_1 +
194 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) -
a_4*cos(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
195 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + a_4*sin(th0_4 + th_4)*(sin(th0_3
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + a_4*sin(th0_4 + th_4)*(sin(th0_3

```

```

+ th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 +
197 th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1))) + a_5*sin(th0_5 + th_5)*(cos(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
198 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
199 th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 +
200 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) ;
201
202 J_out(1,5)= d_6*(sin(th0_5 + th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 +
203 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) +
204 sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(th0_2 + th_2) -
205 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
206 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
207 th_1)*cos(al_1)*cos(al_2)))) - cos(th0_5 + th_5)*sin(al_5)*(sin(th0_4 +
th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) +
208 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
209 th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*(cos(th0_1 +

```


+ th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
 th_2)*sin(th0_1 +
 223 th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + sin(th0_5 +
 th_5)*(sin(th0_4 + th_4)*(sin(th0_3
 224 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
 th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
 th_2)*sin(th0_1 +
 225 th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
 th_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
 226 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 +
 th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
 th_2)*cos(al_1)))) -
 227 a_6*sin(th0_6 + th_6)*(sin(th0_5 +
 th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
 th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
 th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
 228 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
 th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
 + cos(th0_2 + th_2)*sin(th0_1 +
 229 th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
 th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
 th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
 230 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
 cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
 sin(al_3)*(cos(th0_1 +
 231 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
 th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
 th_1)*sin(al_1)*sin(al_2) +
 232 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) - cos(th0_5 +
 th_5)*cos(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
 sin(al_3)*(cos(th0_1 +
 233 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
 th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
 th_1)*sin(al_1)*sin(al_2) +
 234 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
 th_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
 th_1)*cos(al_1)*cos(al_2)) - cos(th0_3

```

235 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
      th_2)*cos(al_1)))));
236
237 J_out(1,6)= a_6*cos(th0_6 + th_6)*(sin(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
      cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
      th_3)*sin(al_3)*(cos(th0_1 +
238 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
      cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
      sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_1)*cos(al_2))) -
239 sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
      th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
      + th_1)*cos(th0_2 + th_2) -
240 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
      th_4)*sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
      th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
      sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
241 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
      cos(th0_2 + th_2)*sin(th0_1 +
242 th_1)*cos(al_1)*cos(al_2)))) + cos(th0_5 +
      th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
      th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
      th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
243 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
      + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
      + cos(th0_2 + th_2)*sin(th0_1 +
244 th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
      th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
      th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
245 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
      cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
      th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
      sin(al_3)*(cos(th0_1 +
246 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
      cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
      th_1)*sin(al_1)*sin(al_2) +
247 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + sin(th0_5 +
      th_5)*cos(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
      th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
      sin(al_3)*(cos(th0_1 +
248 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
      cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +

```

```

th_1)*sin(al_1)*sin(al_2) +
249 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3
250 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)))) - a_6*sin(th0_6 + th_6)*(sin(th0_5 +
th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
251 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) -
252 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
253 th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) -
254 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 +
255 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) - cos(th0_5 +
th_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 +
256 th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
257 th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3
+ th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
258 th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) ;
259
260 J_out(1,7)= 0 ;
261
262 J_out(1,8)= sin(th0_1 + th_1)*sin(al_1) ;
263
264 J_out(1,9)= cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2) ;
265
266 J_out(1,10)= cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +

```

```

th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 +
267 th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) ;
268
269 J_out(1,11)= cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2)
+ sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) -
270 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 +
271 th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(th0_2 + th_2) - sin(th0_1 +
272 th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 +
273 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
274 th_1)*cos(al_1)*cos(al_2))) ;
275
276 J_out(1,12)= cos(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
277 th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 +
278 th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(th0_2 + th_2) - sin(th0_1 +
279 th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 +
280 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
281 th_1)*cos(al_1)*cos(al_2))) - cos(th0_5 +
th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +

```

```

th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
282 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
+ cos(th0_2 + th_2)*sin(th0_1 +
283 th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
284 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 +
285 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) +
286 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) - sin(th0_5 +
th_5)*sin(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 +
287 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) +
288 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3
289 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1))) ;

290
291 J_out(1,13)= cos(th0_1 + th_1) ;
292
293 J_out(1,14)= cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1) ;
294
295 J_out(1,15)= cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
296 th_1)*cos(al_1)*cos(al_2)) ;
297
298 J_out(1,16)= - sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) +
299 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +

```

```

th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) - cos(th0_4 +
300 th_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2
+ th_2) - sin(th0_1 +
301 th_1)*sin(th0_2 + th_2)*cos(al_1)))) ;
302
303 J_out(1,17)= sin(th0_5 + th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 +
304 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)))) +
305 sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(th0_2 + th_2) -
306 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
307 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
308 th_1)*cos(al_1)*cos(al_2)))) - cos(th0_5 + th_5)*(sin(th0_4 + th_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 +
309 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
310 th_1)*cos(al_1)*cos(al_2)))) + cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*(cos(th0_1 +
311 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) ;
312
313 J_out(1,18)= sin(th0_6 + th_6)*(sin(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 +
314 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)))) -

```

```

315 sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
    th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
    th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
    + th_1)*cos(th0_2 + th_2) -
316 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
    th_4)*sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
    th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
    sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
317 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
    cos(th0_2 + th_2)*sin(th0_1 +
318 th_1)*cos(al_1)*cos(al_2)))) + cos(th0_5 +
    th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
    th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
    th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
319 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
    + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
    + cos(th0_2 + th_2)*sin(th0_1 +
320 th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
    th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
    th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
321 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
    cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
    th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
    sin(al_3)*(cos(th0_1 +
322 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
    cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
    th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
    th_1)*sin(al_1)*sin(al_2) +
323 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + sin(th0_5 +
    th_5)*cos(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
    th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
    sin(al_3)*(cos(th0_1 +
324 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
    cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
    th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
    th_1)*sin(al_1)*sin(al_2) +
325 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
    th_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
    sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*cos(al_1)*cos(al_2)) - cos(th0_3
326 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
    th_2)*cos(al_1)))) + cos(th0_6 + th_6)*(sin(th0_5 +
    th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
    sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
327 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
    th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +

```

```

th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) -
328 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
329 th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) -
330 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 +
331 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) - cos(th0_5 +
th_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 +
332 th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
333 th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3
+ th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
334 th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))));
335
336 J_out(1,19)= d_3*(sin(th0_1 + th_1)*cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(th0_1 + th_1)*sin(al_1)*sin(al_2)) + d_4*(cos(al_3)*(sin(th0_1 +
th_1)*cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(th0_1 +
th_1)*sin(al_1)*sin(al_2)) - cos(th0_3 +
337 th_3)*sin(al_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) + sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) +
d_5*(cos(al_4)*(cos(al_3)*(sin(th0_1 +
338 th_1)*cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(th0_1 +
th_1)*sin(al_1)*sin(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_2)*sin(al_1)) + sin(th0_1 + th_1)*sin(th0_2 +
339 th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) + cos(th0_3 +
th_3)*sin(th0_1 + th_1)*sin(th0_2 +
340 th_2)*sin(al_1)) - cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(sin(th0_1 +
th_1)*cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(th0_1 +
th_1)*sin(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1
341 + th_1)*cos(al_2)*sin(al_1)) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 +

```

```

th_3)*cos(al_3)*sin(al_1))) +
d_6*(cos(al_5)*(cos(al_4)*(cos(al_3)*(sin(th0_1 + th_1)*cos(al_1)*cos(al_2)
- cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_1)*sin(al_2))) - cos(th0_3
342 + th_3)*sin(al_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) + sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
343 th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_1)) - cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(sin(th0_1 +
th_1)*cos(al_1)*cos(al_2) - cos(th0_2 +
344 th_2)*sin(th0_1 + th_1)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) +
345 sin(th0_5 + th_5)*sin(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_1)) + sin(th0_4 +
346 th_4)*(sin(al_3)*(sin(th0_1 + th_1)*cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(th0_1 + th_1)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) - sin(th0_1
347 + th_1)*sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) - cos(th0_5 +
th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(sin(th0_1 + th_1)*cos(al_1)*cos(al_2)
- cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_1)*sin(al_2))) - cos(th0_3 +
348 th_3)*sin(al_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) + sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) - sin(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
349 th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_1)) + cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(sin(th0_1 +
th_1)*cos(al_1)*cos(al_2) - cos(th0_2 +
350 th_2)*sin(th0_1 + th_1)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) +
351 a_5*sin(th0_5 + th_5)*(sin(al_4)*(cos(al_3)*(sin(th0_1 +
th_1)*cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(th0_1 +
th_1)*sin(al_1)*sin(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
352 th_1)*cos(al_2)*sin(al_1)) + sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_2)*sin(al_1)) +
353 cos(th0_3 + th_3)*sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_1)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(sin(th0_1 + th_1)*cos(al_1)*cos(al_2) -
cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 +

```

```

354 th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_2)*sin(al_1)) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_3)*sin(al_1))) + a_6*sin(th0_6 +
      th_6)*(sin(al_5)*(cos(al_4)*(cos(al_3)*(sin(th0_1 +
355 th_1)*cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*sin(al_1)*sin(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
      th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_2)*sin(al_1)) + sin(th0_1 + th_1)*sin(th0_2 +
356 th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 +
      th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) +
      cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) + cos(th0_3 +
      th_3)*sin(th0_1 + th_1)*sin(th0_2 +
357 th_2)*sin(al_1)) - cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(sin(th0_1 +
      th_1)*cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*sin(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
      th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1
358 + th_1)*cos(al_2)*sin(al_1)) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_3)*sin(al_1))) - sin(th0_5 + th_5)*cos(al_5)*(cos(th0_4 +
      th_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2
      + th_2)*sin(th0_1 +
359 th_1)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_1 + th_1)*sin(th0_2 +
      th_2)*sin(al_1)) + sin(th0_4 + th_4)*(sin(al_3)*(sin(th0_1 +
      th_1)*cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*sin(al_1)*sin(al_2)) + cos(th0_3 +
360 th_3)*cos(al_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) - sin(th0_1 + th_1)*sin(th0_2 +
      th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) + cos(th0_5 +
      th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(sin(th0_1
361 + th_1)*cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*sin(al_1)*sin(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
      th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_2)*sin(al_1)) + sin(th0_1 + th_1)*sin(th0_2 +
362 th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) - sin(th0_4 +
      th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) +
      cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) + cos(th0_3 +
      th_3)*sin(th0_1 + th_1)*sin(th0_2 +
363 th_2)*sin(al_1)) + cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(sin(th0_1 +
      th_1)*cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*sin(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
      th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1
364 + th_1)*cos(al_2)*sin(al_1)) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_3)*sin(al_1))) + d_2*sin(th0_1 + th_1)*cos(al_1) +
      a_5*cos(th0_5 + th_5)*(cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
      th_1)*cos(al_1)*sin(al_2) + cos(th0_2
365 + th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_1 +
      th_1)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 + th_4)*(sin(al_3)*(sin(th0_1
      + th_1)*cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*sin(al_1)*sin(al_2)) + cos(th0_3
366 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) - sin(th0_1 + th_1)*sin(th0_2 +

```

```

th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) + a_4*cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
367 th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_1)) + a_6*cos(th0_6 + th_6)*(cos(th0_5 + th_5)*(cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
368 th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_1)) + sin(th0_4 + th_4)*(sin(al_3)*(sin(th0_1 +
th_1)*cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(th0_1 +
369 th_1)*sin(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_2)*sin(al_1)) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1))) + sin(th0_5 +
370 th_5)*(sin(al_4)*(cos(al_3)*(sin(th0_1 + th_1)*cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(th0_1 + th_1)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) +
371 sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) -
sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_1 +
372 th_1)*sin(th0_2 + th_2)*sin(al_1)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(sin(th0_1 + th_1)*cos(al_1)*cos(al_2) -
cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2
373 + th_2)*sin(th0_1 + th_1)*cos(al_2)*sin(al_1)) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) + a_3*sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_2)*sin(al_1)) +
374 a_4*sin(th0_4 + th_4)*(sin(al_3)*(sin(th0_1 + th_1)*cos(al_1)*cos(al_2) -
cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 +
375 th_1)*cos(al_2)*sin(al_1)) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) + a_2*sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_1) + a_3*cos(th0_3 + th_3)*sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_1) ;

376
377 J_out(1,20)= d_4*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) +
378 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + d_3*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) +
379 d_5*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +

```

```

th_1)*sin(th0_2 + th_2)*sin(al_2) +
380 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) - cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1
381 + th_1)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2))) + sin(th0_3 +
th_3)*sin(th0_4 +
382 th_4)*sin(al_4)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) +
d_6*(cos(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) -
383 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 +
384 th_1)*cos(al_1)*sin(al_2))) - cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
+ cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1
385 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2))) + sin(th0_3 +
th_3)*sin(th0_4 + th_4)*sin(al_4)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 +
386 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + sin(th0_5 + th_5)*sin(al_5)*(sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1
+ th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
387 th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + cos(th0_4 + th_4)*sin(th0_3 +
388 th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) - cos(th0_5 +
th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) -
389 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 +
390 th_1)*cos(al_1)*sin(al_2))) + cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
+ cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1
391 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2))) - sin(th0_3 +
th_3)*sin(th0_4 + th_4)*cos(al_4)*(cos(th0_1 + th_1)*sin(th0_2 +

```

```

th_2)*sin(al_2) + sin(th0_1 +
392 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + a_5*cos(th0_5 + th_5)*(sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1
+ th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
393 th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + cos(th0_4 + th_4)*sin(th0_3 +
394 th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + a_5*sin(th0_5 +
th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
395 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 +
396 th_1)*cos(al_1)*sin(al_2))) + cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
+ cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1
397 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2))) - sin(th0_3 +
th_3)*sin(th0_4 + th_4)*cos(al_4)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 +
398 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + a_6*sin(th0_6 +
th_6)*(sin(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
399 th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) - cos(th0_4 +
400 th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) +
401 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + sin(th0_3 + th_3)*sin(th0_4 +
th_4)*sin(al_4)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
402 th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2))) - sin(th0_5 +
th_5)*cos(al_5)*(sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) +
403 cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + cos(th0_4 + th_4)*sin(th0_3 + th_3)*(cos(th0_1
+ th_1)*sin(th0_2 +
404 th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +

```

```

th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2))) + cos(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
405 + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + cos(th0_4 +
406 th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) +
407 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) - sin(th0_3 + th_3)*sin(th0_4 +
th_4)*cos(al_4)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
408 th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)))) + a_3*sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + a_6*cos(th0_6 + th_6)*(sin(th0_5 +
409 th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) +
410 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1
411 + th_1)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2))) - sin(th0_3 +
th_3)*sin(th0_4 +
412 th_4)*cos(al_4)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + cos(th0_5 + th_5)*(sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
413 th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
414 th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2))) + cos(th0_4 + th_4)*sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + a_4*sin(th0_4 +
415 th_4)*(sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
416 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2))) + a_4*cos(th0_4 + th_4)*sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +

```

```

th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
417 th_1)*cos(al_1)*sin(al_2)) ;
418
419 J_out(1,21)= d_5*(cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
420 th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) - cos(th0_4 +
421 th_4)*sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 +
422 th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + d_4*(sin(th0_3 +
423 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 +
424 th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) +
d_6*(cos(al_5)*(cos(al_4)*(sin(th0_3 +
425 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 +
426 th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) - cos(th0_4 +
th_4)*sin(al_4)*(cos(al_3)*(cos(th0_1
427 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) +
428 cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) - cos(th0_5 +
th_5)*sin(al_5)*(sin(al_4)*(sin(th0_3 +
429 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 +
430 th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*cos(al_4)*(cos(al_3)*(cos(th0_1

```

431 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
 th_1)*sin(th0_2 + th_2)*cos(al_1)) +
 432 cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
 th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*sin(th0_5 +
 th_5)*sin(al_5)*(cos(al_3)*(cos(th0_1 +
 433 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
 th_1)*sin(th0_2 + th_2)*cos(al_1)) +
 434 cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
 th_1)*cos(al_1)*cos(al_2))) + a_6*cos(th0_6 + th_6)*(sin(th0_5 +
 th_5)*(sin(al_4)*(sin(th0_3 +
 435 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
 th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
 th_2)*sin(th0_1 +
 436 th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
 th_4)*cos(al_4)*(cos(al_3)*(cos(th0_1
 437 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
 th_1)*sin(th0_2 + th_2)*cos(al_1)) +
 438 cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
 th_1)*cos(al_1)*cos(al_2))) + cos(th0_5 + th_5)*sin(th0_4 +
 th_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
 439 th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
 th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
 th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
 440 th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
 th_1)*cos(al_1)*cos(al_2))) + a_6*sin(th0_6 +
 th_6)*(sin(al_5)*(cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
 441 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
 sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
 th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
 442 th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
 th_1)*cos(al_1)*cos(al_2))) - cos(th0_4 +
 th_4)*sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
 443 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
 th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +

```

th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
444 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + cos(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 +
445 th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
446 th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2
447 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
448 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)))) - sin(th0_4 + th_4)*sin(th0_5 +
th_5)*cos(al_5)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
449 th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
450 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)))) + a_4*sin(th0_4 + th_4)*(cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 +
451 th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 +
452 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) + a_5*sin(th0_5 + th_5)*(sin(al_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) -
453 sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
454 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
455 th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 +

```

```

456 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_1)*cos(al_2)))) + a_5*cos(th0_5 + th_5)*sin(th0_4 +
      th_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1
      + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
457 th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
      th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
      + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
458 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_1)*cos(al_2))) ;
459
460 J_out(1,22)= a_5*sin(th0_5 + th_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
      cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
      th_3)*sin(al_3)*(cos(th0_1 +
461 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
      cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
      sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_1)*cos(al_2))) -
462 sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
      th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
      + th_1)*cos(th0_2 + th_2) -
463 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
      th_4)*sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
      th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
      sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
464 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
      cos(th0_2 + th_2)*sin(th0_1 +
465 th_1)*cos(al_1)*cos(al_2)))) - d_6*(cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
      cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
      th_3)*sin(al_3)*(cos(th0_1
466 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
      cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
      sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_1)*cos(al_2))) +
467 sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
      th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
      + th_1)*cos(th0_2 + th_2) -
468 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
      th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
      th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
      sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
469 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
      th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +

```

```

cos(th0_2 + th_2)*sin(th0_1 +
470 th_1)*cos(al_1)*cos(al_2)))) + cos(th0_5 +
th_5)*sin(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
471 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
+ cos(th0_2 + th_2)*sin(th0_1 +
472 th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
473 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) +
cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 +
474 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) +
475 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) - a_6*sin(th0_6 +
th_6)*(sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 +
476 th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 +
477 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
478 th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 +
479 th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
480 th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) - cos(th0_5 +
th_5)*cos(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1)
481 + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) -
482 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +

```

```

th_1)*cos(al_1))*cos(th0_4 + th_4)*sin(al_4))*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
483 th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) +
cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) -
484 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 +
485 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) -
d_5*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
486 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) -
487 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
488 th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) -
489 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 +
490 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + a_6*cos(th0_6 +
th_6)*sin(th0_5 + th_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) +
491 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
492 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 +
493 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2
+ th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 +
494 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +

```

```

th_2)*sin(th0_1 +
495 th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) ;
496
497 J_out(1,23)= - d_6*(sin(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 +
498 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) -
499 sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(th0_2 + th_2) -
500 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
501 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
502 th_1)*cos(al_1)*cos(al_2)))) + cos(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
503 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
+ cos(th0_2 + th_2)*sin(th0_1 +
504 th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
505 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 +
506 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) +
507 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + sin(th0_5 +
th_5)*cos(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 +
508 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +

```



```

th_1)*cos(al_1)*cos(al_2)))) + sin(th0_5 +
521 th_5)*sin(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 +
th_1)*cos(al_2)*sin(al_1) +
522 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
523 th_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2
+ th_2) - sin(th0_1 +
524 th_1)*sin(th0_2 + th_2)*cos(al_1)))))) ;
525
526 J_out(1,24)= 0 ;
527
528 J_out(2,1)= d_5*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) -
529 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 +
530 th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(th0_2 + th_2) - sin(th0_1 +
531 th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 +
532 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
533 th_1)*cos(al_1)*cos(al_2)))) - d_6*(cos(th0_5 +
th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
534 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
+ cos(th0_2 + th_2)*sin(th0_1 +
535 th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
536 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -

```

```

cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 +
537 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) +
538 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) -
cos(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
539 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2)
+ cos(th0_2 + th_2)*sin(th0_1 +
540 th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
541 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) +
cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 +
542 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) +
543 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + sin(th0_5 +
th_5)*sin(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 +
544 th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) +
545 cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3
546 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)))) + a_1*cos(th0_1 + th_1) + d_3*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 +
547 th_1)*cos(al_1)*sin(al_2)) + d_4*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
548 th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +

```

```

th_1)*cos(al_1)*cos(al_2))) + a_5*sin(th0_5 +
549 th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 +
550 th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 +
551 th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2
+ th_2) - sin(th0_1 + th_1)*sin(th0_2 +
552 th_2)*cos(al_1))) - cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 +
553 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
554 th_1)*cos(al_1)*cos(al_2))) + a_2*cos(th0_1 + th_1)*cos(th0_2 + th_2) +
a_3*cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - a_5*cos(th0_5 + th_5)*(sin(th0_4 +
th_4)*(sin(th0_3 +
555 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 +
556 th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
557 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1))) +
558 a_6*cos(th0_6 + th_6)*(sin(th0_5 + th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 +
559 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) +
560 sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(th0_2 + th_2) -
561 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +

```

```

th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
562 sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
563 th_1)*cos(al_1)*cos(al_2)))) - cos(th0_5 + th_5)*(sin(th0_4 + th_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 +
564 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
565 th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*(cos(th0_1 +
566 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) +
d_2*sin(th0_1 + th_1)*sin(al_1) - a_3*sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
567 th_1)*cos(al_1)*cos(al_2)) - a_4*sin(th0_4 + th_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2
+ th_2)*sin(al_2) + sin(th0_1 +
568 th_1)*cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 +
569 th_1)*cos(al_1)*cos(al_2))) - a_4*cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
570 th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) +
a_6*sin(th0_6 + th_6)*(sin(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 +
571 th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 +
572 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) - sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
573 th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2 +
th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(th0_2 +

```

```

th_2) - sin(th0_1 +
574 th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
575 th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + cos(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1)
576 + cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(th0_2 + th_2) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) -
577 sin(th0_1 + th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 +
th_1)*sin(al_1)*sin(al_2) + cos(th0_2 +
578 th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) -
579 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 +
580 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2)))) + sin(th0_5 +
th_5)*cos(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(th0_2 + th_2) -
581 sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(al_3)*(cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + sin(th0_1 + th_1)*cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 +
582 th_1)*sin(th0_2 + th_2)*cos(al_2) - sin(th0_1 + th_1)*sin(al_1)*sin(al_2) +
cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) -
sin(th0_1 +
583 th_1)*sin(al_1)*sin(al_2) + cos(th0_2 + th_2)*sin(th0_1 +
th_1)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(th0_2
+ th_2) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) - a_2*sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_1) ;
584
585 J_out(2,2)= d_3*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_2) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) + d_4*(cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1)*sin(al_2) + cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*sin(al_2)) +
586 cos(th0_3 + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) +
cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1))) -

```

```

587 d_6*(cos(al_5)*(sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 + th_1)*sin(th0_2 +
    th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2
    + th_2) - cos(th0_1 + th_1)*cos(th0_2
588 + th_2)*cos(al_1))) - cos(al_4)*(cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*sin(al_2) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
    cos(th0_3 + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) +
    cos(th0_1 + th_1)*sin(th0_2
589 + th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
    th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) +
    cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*sin(al_2) + cos(th0_1 +
590 th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
    th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
    th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) -
591 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) + sin(th0_5 +
    th_5)*sin(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 + th_1)*sin(th0_2 +
    th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(sin(th0_1 +
592 th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) -
    sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_2)
    + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
    th_3)*cos(al_3)*(cos(th0_2 +
593 th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 + th_1)*sin(th0_2 +
    th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
    th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) +
    cos(th0_5 +
594 th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*sin(al_2) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
    cos(th0_3 + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) +
    cos(th0_1 + th_1)*sin(th0_2 +
595 th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
    th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) +
    sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
    + th_1)*cos(al_2) + cos(th0_1 +
596 th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(sin(th0_1 +
    th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) +
    cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*sin(al_2) + cos(th0_1 +
597 th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
    th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
    th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) -
598 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) - d_5*(sin(th0_4 +
    th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*cos(al_2) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) +
    cos(th0_3 + th_3)*(sin(th0_1 +
599 th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) -
    cos(al_4)*(cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_2) +

```

```

        cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
        th_3)*sin(al_3)*(cos(th0_2 +
600 th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 + th_1)*sin(th0_2 +
        th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
        th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) +
        cos(th0_4 +
601 th_4)*sin(al_4)*(sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_2) +
        cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
        th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
        th_1)*sin(th0_2 +
602 th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
        th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) +
        a_4*sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
        th_1)*sin(al_2) + cos(th0_1 +
603 th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
        th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
        th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
        th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) -
604 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) + a_6*sin(th0_6 +
        th_6)*(sin(th0_5 + th_5)*cos(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
        th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
        th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) +
605 cos(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) - cos(th0_1 +
        th_1)*cos(th0_2 + th_2)*cos(al_1))) - sin(th0_4 +
        th_4)*(sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_2) + cos(th0_1
        + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
606 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
        th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
        th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) - cos(th0_1 +
        th_1)*cos(th0_2 + th_2)*cos(al_1))) -
607 sin(al_5)*(sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_2 +
        th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 + th_1)*sin(th0_2 +
        th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2
        + th_2) - cos(th0_1 + th_1)*cos(th0_2 +
608 th_2)*cos(al_1))) - cos(al_4)*(cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
        th_1)*sin(al_2) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
        cos(th0_3 + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) +
        cos(th0_1 + th_1)*sin(th0_2 +
609 th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
        th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) +
        cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
        th_1)*sin(al_2) + cos(th0_1 +
610 th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
        th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
        th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
        th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) -
611 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) + cos(th0_5 +
        th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
        th_1)*sin(al_2) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
        cos(th0_3 + th_3)*sin(al_3)*(cos(th0_2 +

```

```

612 th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 + th_1)*sin(th0_2 +
    th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
    th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) +
    sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
613 th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(sin(th0_1
    + th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)))
    + cos(th0_4 +
614 th_4)*cos(al_4)*(sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_2) +
    cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
    th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
    th_1)*sin(th0_2 +
615 th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
    th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))))
    - a_3*cos(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) - cos(th0_1 +
    th_1)*cos(th0_2 + th_2)*cos(al_1))
616 - a_2*sin(th0_1 + th_1)*sin(th0_2 + th_2) - a_6*cos(th0_6 + th_6)*(cos(th0_5 +
    th_5)*(cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*cos(al_2) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) +
    cos(th0_3 +
617 th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1))) - sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1)*sin(al_2) + cos(th0_1 + th_1)*sin(th0_2 +
    th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
618 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
    th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) - cos(th0_1 +
    th_1)*cos(th0_2 + th_2)*cos(al_1))) - sin(th0_5 +
619 th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_2) +
    cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
    th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
    th_1)*sin(th0_2 +
620 th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
    th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) +
    sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
    + th_1)*cos(al_2) + cos(th0_1 +
621 th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(sin(th0_1 +
    th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) +
    cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
    th_1)*sin(al_2) + cos(th0_1 +
622 th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
    th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 +
    th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) -
623 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)))) + a_5*sin(th0_5 +
    th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_2) +
    cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
    th_3)*sin(al_3)*(cos(th0_2 +
624 th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 + th_1)*sin(th0_2 +
    th_2)*cos(al_1)*cos(al_2)) - sin(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +

```

```

th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) +
sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
625 th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)))
+ cos(th0_4 +
626 th_4)*cos(al_4)*(sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_2) +
cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 +
th_1)*sin(th0_2 +
627 th_2)*cos(al_1)*cos(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) -
a_3*sin(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) +
cos(th0_1 + th_1)*sin(th0_2 +
628 th_2)*cos(al_1)*cos(al_2)) - a_4*cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1)*cos(al_2) + cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2
+ th_2) - cos(th0_1 +
629 th_1)*cos(th0_2 + th_2)*cos(al_1))) - a_5*cos(th0_5 + th_5)*(cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*cos(al_2) +
cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
630 th_2) - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1))) - sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)*sin(al_2) + cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
631 th_1)*cos(al_2) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)*cos(al_2)) +
sin(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2) - cos(th0_1
+ th_1)*cos(th0_2 + th_2)*cos(al_1))) + a_2*cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1) ;
632
633 J_out(2,3)= d_4*(cos(th0_3 + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) -
634 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) - d_6*(sin(th0_5 +
th_5)*sin(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*(sin(th0_1 +
635 th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 +
th_4)*(cos(th0_3 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) +
cos(th0_1 + th_1)*sin(th0_2 +
636 th_2)*cos(al_1)) - sin(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) -
cos(al_5)*(cos(al_4)*(cos(th0_3 +
637 th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 + th_3)*sin(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)

```



```

th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*sin(al_4)*(cos(th0_3
651 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 + th_1)*cos(th0_2 +
652 th_2)*cos(al_1)*cos(al_2)))) + sin(th0_5 + th_5)*cos(al_5)*(cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) +
653 cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*(cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
654 th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)))) + cos(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(cos(th0_3 + th_3)*sin(al_3)*(cos(th0_2 +
655 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(th0_3 + th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) +
656 sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
657 th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) - cos(th0_4 +
th_4)*cos(al_4)*(cos(th0_3 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 +
658 th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) - a_4*sin(th0_4 +
th_4)*(cos(th0_3 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) +
cos(th0_1 + th_1)*sin(th0_2 +
659 th_2)*cos(al_1)) - sin(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) - a_3*sin(th0_3 +
th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1)
660 + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) + a_6*cos(th0_6 +
th_6)*(sin(th0_5 + th_5)*(sin(al_4)*(cos(th0_3 + th_3)*sin(al_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
sin(th0_3 +
661 th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) +
662 cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) - cos(th0_4 +

```

```

th_4)*cos(al_4)*(cos(th0_3
663 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 + th_1)*cos(th0_2 +
664 th_2)*cos(al_1)*cos(al_2)))) - cos(th0_5 + th_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)) + cos(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 +
665 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*(cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
666 th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)))) - a_3*cos(th0_3 + th_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
667 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + a_5*sin(th0_5 + th_5)*(sin(al_4)*(cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 +
668 th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) +
669 cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) - cos(th0_4 +
th_4)*cos(al_4)*(cos(th0_3
670 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 + th_1)*cos(th0_2 +
671 th_2)*cos(al_1)*cos(al_2)))) - a_4*cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)) + cos(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 +
672 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) - a_5*cos(th0_5 + th_5)*(cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
673 th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*(cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
674 th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) ;
675
676 J_out(2,4)= a_5*cos(th0_5 + th_5)*(sin(th0_4 + th_4)*(sin(th0_3 +

```

```

th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
677 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
cos(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_1)*sin(al_2)) +
678 sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 +
679 th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) - d_6*(cos(al_5)*(cos(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 +
680 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + sin(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 +
681 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 +
682 th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) - sin(th0_5 +
th_5)*sin(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
683 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 +
684 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) +
685 cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)))) + cos(th0_5 + th_5)*sin(al_5)*(sin(th0_4 +
686 th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 +
687 th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 +
688 th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 +

```

```

689 th_2)*cos(al_1)))) - d_5*(cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
    th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
    th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
690 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) +
    sin(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 +
    th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
    cos(th0_1 + th_1)*cos(th0_2 +
691 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
    cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
    cos(th0_1 +
692 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2)))) - a_4*cos(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 +
    th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
    cos(th0_1 + th_1)*cos(th0_2 +
693 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
    cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
    cos(th0_1 +
694 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2))) + a_5*sin(th0_5 + th_5)*(sin(th0_4 +
    th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
    sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 +
695 th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
    th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
    + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
696 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
    th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
    th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
697 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
    th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) + a_6*cos(th0_6 +
    th_6)*(sin(th0_5 + th_5)*(sin(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 +
698 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
    cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
    th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_1)) +
699 cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
    cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
    th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
700 th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
    th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
    + th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))
    + cos(th0_5 + th_5)*(sin(th0_4 +
701 th_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
    cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1

```

```

+ th_1) + cos(th0_1 +
702 th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 +
703 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)))) +
704 a_4*sin(th0_4 + th_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) + cos(th0_1
705 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - a_6*sin(th0_6 +
th_6)*(sin(al_5)*(cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 + th_1)*cos(th0_2 +
706 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + sin(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 +
707 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 +
708 th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) + sin(th0_5 +
th_5)*cos(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
709 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 +
710 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) +
711 cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)))) - cos(th0_5 + th_5)*cos(al_5)*(sin(th0_4 +
712 th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 +
713 th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 +
714 th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +

```

```

th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 +
715 th_2)*cos(al_1)))) ;
716
717 J_out(2,5)= d_6*(sin(th0_5 + th_5)*sin(al_5)*(sin(al_4)*(sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 +
718 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) +
719 sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) +
720 cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
721 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 + th_1)*cos(th0_2 +
722 th_2)*cos(al_1)*cos(al_2)))) - cos(th0_5 + th_5)*sin(al_5)*(sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
723 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 + th_1)*cos(th0_2 +
724 th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*(cos(th0_2 +
725 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) +
a_5*sin(th0_5 + th_5)*(sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 +
726 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 +
727 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
728 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) - a_6*sin(th0_6 +

```

```

th_6)*(sin(th0_5 + th_5)*cos(al_5)*(sin(al_4)*(sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_2 +
729 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
730 th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
731 th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))
- cos(th0_4 +
732 th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 +
733 th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) - cos(th0_5 +
th_5)*cos(al_5)*(sin(th0_4 +
734 th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 +
735 th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 +
th_4)*(sin(th0_3 +
736 th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 +
737 th_2)*cos(al_1)))) + a_5*cos(th0_5 + th_5)*(sin(al_4)*(sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 +
738 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) +
739 sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) +
740 cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +

```

```

741 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
    + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
    - cos(th0_1 + th_1)*cos(th0_2 +
742 th_2)*cos(al_1)*cos(al_2))) + a_6*cos(th0_6 + th_6)*(cos(th0_5 +
    th_5)*(sin(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
    th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
    cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
743 sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
    cos(th0_1 + th_1)*cos(th0_2 +
744 th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
    th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
    th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
745 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
    cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 +
    th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
    cos(th0_1 + th_1)*cos(th0_2 +
746 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
    cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
    cos(th0_1 +
747 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2))) + sin(th0_5 + th_5)*(sin(th0_4 +
    th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
    th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
748 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
    cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
    cos(th0_1 +
749 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3 +
    th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
    th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
750 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
    th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) ;
751
752 J_out(2,6)= a_6*cos(th0_6 + th_6)*(sin(al_5)*(cos(al_4)*(sin(th0_3 +
    th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
    th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 +
753 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
    cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
    cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2))) -
754 sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
    th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
    th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2

```

```

+ th_2)*sin(th0_1 + th_1) +
755 cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
756 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 + th_1)*cos(th0_2 +
757 th_2)*cos(al_1)*cos(al_2)))) + cos(th0_5 + th_5)*cos(al_5)*(sin(al_4)*(sin(th0_3
+ th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
758 th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 +
759 th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
760 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 +
761 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 +
762 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)))) + sin(th0_5 + th_5)*cos(al_5)*(sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 +
763 th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
764 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
765 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) - a_6*sin(th0_6 +
th_6)*(sin(th0_5 + th_5)*(sin(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1)
766 + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 +
767 th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -

```

```

cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 +
768 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 +
769 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) +
770 cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) - cos(th0_5 + th_5)*(sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 +
771 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) +
772 cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
773 th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1)))));
774
775 J_out(2,7)= 0 ;
776
777 J_out(2,8)= -cos(th0_1 + th_1)*sin(al_1) ;
778
779 J_out(2,9)= sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) - cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2) ;
780
781 J_out(2,10)= sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) +
cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2
782 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) ;
783
784 J_out(2,11)= cos(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 +
785 th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +

```

```

th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) - sin(th0_4 +
786 th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) + cos(th0_1 +
787 th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
788 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 + th_1)*cos(th0_2 +
789 th_2)*cos(al_1)*cos(al_2))) ;
790
791 J_out(2,12)= cos(al_5)*(cos(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) +
792 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) - sin(th0_4 +
793 th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) + cos(th0_1 +
794 th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
795 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
- cos(th0_1 + th_1)*cos(th0_2 +
796 th_2)*cos(al_1)*cos(al_2))) - cos(th0_5 + th_5)*sin(al_5)*(sin(al_4)*(sin(th0_3
+ th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
797 th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 +
798 th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
799 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 +

```

```

th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 +
800 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 +
801 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) - sin(th0_5 + th_5)*sin(al_5)*(sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 +
802 th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
803 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
804 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) ;

805
806 J_out(2,13)= sin(th0_1 + th_1) ;
807
808 J_out(2,14)= cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_1) ;

809
810 J_out(2,15)= cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_1)) - sin(th0_3 + th_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 +
811 th_2)*cos(al_1)*cos(al_2)) ;
812
813 J_out(2,16)= - sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
814 th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) - cos(th0_4 +
815 th_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 +
816 th_1)*sin(th0_2 + th_2)*cos(al_1)))) ;
817
818 J_out(2,17)= sin(th0_5 + th_5)*(sin(al_4)*(sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2)

```

```

819 + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
      th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 +
820 th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
      th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
      th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
      + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
821 th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
      th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
      sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
      + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
822 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
      + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
      - cos(th0_1 + th_1)*cos(th0_2 +
823 th_2)*cos(al_1)*cos(al_2))) - cos(th0_5 + th_5)*(sin(th0_4 +
      th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
      th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
824 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
      + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
      - cos(th0_1 + th_1)*cos(th0_2 +
825 th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(sin(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
      cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 +
      th_3)*(cos(th0_2 +
826 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) ;
827
828 J_out(2,18)= cos(th0_6 + th_6)*(sin(th0_5 + th_5)*(sin(al_4)*(sin(th0_3 +
      th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
      th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
829 th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
      cos(th0_1 + th_1)*cos(th0_2 +
830 th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
      th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
      th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
831 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
      cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 +
      th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
      cos(th0_1 + th_1)*cos(th0_2 +
832 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
      th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
      cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
      cos(th0_1 +

```



```

th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
846 th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) + sin(th0_5 +
th_5)*cos(al_5)*(sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
847 th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
848 th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) + cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 +
849 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))));
850
851 J_out(2,19)= - d_3*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) -
d_6*(cos(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2)
- cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) -
852 cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 +
853 th_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1)) - cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_1)*cos(al_2) -
854 cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 +
855 th_3)*cos(al_3)*sin(al_1))) - cos(th0_5 +
th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2)
- cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) +
856 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) - sin(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 +
857 th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1)) + cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_1)*cos(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
858 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) + sin(th0_5 +
th_5)*sin(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 +

```

```

859 th_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*sin(al_1)) + sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 +
      th_1)*cos(al_1)*cos(al_2) - cos(th0_1 +
860 th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
      th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_1 + th_1)*sin(th0_2 +
      th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1)))) -
861 d_5*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) - cos(th0_1 +
      th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
      th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
      th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
862 cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) +
      sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
      th_1)*cos(al_1)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*cos(th0_3 +
863 th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
      th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) -
      cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1
864 + th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_1 + th_1)*sin(th0_2 +
      th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) - d_4*(cos(al_3)*(cos(th0_1 +
      th_1)*cos(al_1)*cos(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
865 th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
      th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*sin(th0_2 +
      th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) - a_6*sin(th0_6 +
866 th_6)*(sin(al_5)*(cos(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) -
      cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
      th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
      th_1)*cos(th0_2 +
867 th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
      th_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) +
868 cos(th0_1 + th_1)*cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
      th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) -
      cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(cos(th0_1 +
869 th_1)*cos(al_1)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) - cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_3)*sin(al_1))) + cos(th0_5 +
      th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2)
      -
870 cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
      th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
      th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*sin(th0_2 +
      th_2)*sin(th0_3 +
871 th_3)*sin(al_1)*sin(al_3)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
      th_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +

```

```

th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1)) + cos(th0_4 +
872 th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) -
873 cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) -
sin(th0_5 + th_5)*cos(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_1
+ th_1)*cos(al_1)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 +
874 th_1)*cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) +
875 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) -
a_5*cos(th0_5 + th_5)*(cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
876 th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*cos(th0_3 +
th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1
+ th_1)*cos(al_1)*cos(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
877 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) - d_2*cos(th0_1 +
th_1)*cos(al_1) - a_5*sin(th0_5 +
878 th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
879 cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) -
sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(al_1)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*cos(th0_3 +
880 th_3)*sin(th0_2 + th_2)*sin(al_1)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1
881 + th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) - a_6*cos(th0_6 +
th_6)*(cos(th0_5 + th_5)*(cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(al_1)*sin(al_2) +
882 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 +
th_1)*cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) +
883 cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) + sin(th0_5 +
884 th_5)*(sin(al_4)*(cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) - cos(th0_1 +

```

```

th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
885 cos(th0_1 + th_1)*sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) -
sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(th0_1 +
th_1)*cos(al_1)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*cos(th0_3 +
886 th_3)*sin(th0_2 + th_2)*sin(al_1)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_1)*cos(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1
887 + th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) - a_4*cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1
+ th_1)*cos(th0_2 +
888 th_2)*cos(al_2)*sin(al_1)) + cos(th0_1 + th_1)*cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1)) - a_4*sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_1)*cos(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
889 th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) + cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1)) - a_3*sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*cos(al_1)*sin(al_2) +
890 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - a_2*cos(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_1) - a_3*cos(th0_1 + th_1)*cos(th0_3 +
th_3)*sin(th0_2 + th_2)*sin(al_1) ;
891
892 J_out(2,20)= d_4*(cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1
893 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2))) - d_6*(cos(al_5)*(cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
894 th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2))) - cos(al_4)*(cos(al_3)*(sin(th0_1
895 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) +
896 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) + sin(th0_3 +
th_3)*sin(th0_4 + th_4)*sin(al_4)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_1)*sin(al_2))) -
897 sin(th0_5 + th_5)*sin(al_5)*(sin(th0_4 + th_4)*(sin(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 +

```

```

th_3)*cos(al_3)*(cos(th0_1 +
898 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) - cos(th0_4 +
th_4)*sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 +
899 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) +
cos(th0_5 + th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 +
900 th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(sin(th0_1
901 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) +
902 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) + sin(th0_3 +
th_3)*sin(th0_4 + th_4)*cos(al_4)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_1)*sin(al_2))) -
903 d_5*(cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 +
904 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) -
cos(al_4)*(cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1
905 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2))) + sin(th0_3 + th_3)*sin(th0_4 +
906 th_4)*sin(al_4)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2))) + d_3*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
907 - cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) + a_5*cos(th0_5 +
th_5)*(sin(th0_4 + th_4)*(sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 +
908 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) - cos(th0_4 +
th_4)*sin(th0_3 + th_3)*(cos(th0_1 +
909 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) + a_5*sin(th0_5 +
th_5)*(sin(al_4)*(cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 +
910 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +

```

[illegible]

```

cos(th0_1 + th_1)*cos(th0_2 +
923 th_2)*cos(al_1)*sin(al_2))) + sin(th0_3 + th_3)*sin(th0_4 +
th_4)*sin(al_4)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2))) + sin(th0_5 +
924 th_5)*cos(al_5)*(sin(th0_4 + th_4)*(sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 +
925 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) - cos(th0_4 +
th_4)*sin(th0_3 + th_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 +
926 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) -
cos(th0_5 + th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 +
927 th_2)*cos(al_1)*cos(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(sin(th0_1
928 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) +
929 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2))) + sin(th0_3 +
th_3)*sin(th0_4 + th_4)*cos(al_4)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_1)*sin(al_2))) +
930 a_4*sin(th0_4 + th_4)*(sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2)
+ cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) -
931 sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2))) - a_4*cos(th0_4 + th_4)*sin(th0_3 +
th_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 +
th_2)*sin(al_2) + cos(th0_1 +
932 th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) ;
933
934 J_out(2,21)= d_5*(cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
+ th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) +
935 cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) - cos(th0_4 +
936 th_4)*sin(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +

```



```

th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1
962 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
+ th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
963 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) - cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
964 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) -
965 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + cos(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1)
- sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 +
th_1)*cos(th0_2 +
966 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 +
967 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
968 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) -
969 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) - sin(th0_4 +
th_4)*sin(th0_5 + th_5)*cos(al_5)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
cos(al_3)*(cos(th0_1 +
970 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) -
971 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) + a_5*cos(th0_5 +
th_5)*sin(th0_4 + th_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
cos(al_3)*(cos(th0_1 +
972 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) -
973 cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) ;
974
975 J_out(2,22)= a_5*sin(th0_5 + th_5)*(cos(al_4)*(sin(th0_3 +
th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +

```



```

988 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) +
    cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 +
    th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
    cos(th0_1 + th_1)*cos(th0_2 +
989 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
    cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
    cos(th0_1 +
990 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2)))) - a_6*sin(th0_6 +
    th_6)*(sin(al_5)*(sin(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
991 cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 +
    th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
    cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
    cos(th0_1 +
992 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
    th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
    th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
993 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
    th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) - cos(th0_4 +
    th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
    sin(th0_1 + th_1)*sin(th0_2 +
994 th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
    sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1
    + th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 +
    th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 +
995 th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
    th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) - cos(th0_5 +
    th_5)*cos(al_5)*(cos(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 +
996 th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1
    + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
    th_1)*sin(th0_2 + th_2)*cos(al_2) +
997 cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*cos(al_2))) - sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
    th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
    th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
998 th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) +
    cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 +
    th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
999 th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
    th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
    th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
    cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
1000 th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
    cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) -

```

```

d_5*(sin(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 +
th_1) + cos(th0_1 + th_1)*sin(th0_2 +
1001 th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1
+ th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) +
1002 cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
1003 th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
1004 th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
1005 th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) + a_6*cos(th0_6 +
th_6)*sin(th0_5 + th_5)*(cos(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) +
1006 cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 +
1007 th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) - sin(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 +
1008 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 +
1009 th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) +
1010 cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)))) ;
1011
1012 J_out(2,23)= - d_6*(sin(al_5)*(cos(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2)
1013 + cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) - sin(th0_4 +

```

```

1014 th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
      th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
      th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
      + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
1015 th_1)*sin(th0_2 + th_2)*cos(al_1))) + cos(th0_4 +
      th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) -
      sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2
      + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
1016 th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
      + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2)
      - cos(th0_1 + th_1)*cos(th0_2 +
1017 th_2)*cos(al_1)*cos(al_2))) + cos(th0_5 + th_5)*cos(al_5)*(sin(al_4)*(sin(th0_3
      + th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
      th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
1018 th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
      cos(th0_1 + th_1)*cos(th0_2 +
1019 th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
      th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
      th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 +
1020 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1))) -
      cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 +
      th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
      cos(th0_1 + th_1)*cos(th0_2 +
1021 th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
      th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
      cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
      cos(th0_1 +
1022 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_1)*cos(al_2))) + sin(th0_5 + th_5)*cos(al_5)*(sin(th0_4 +
      th_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
      th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 +
1023 th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
      th_3)*cos(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
      th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1
      + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
1024 th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
      th_2)*cos(al_1)*cos(al_2))) + cos(th0_4 + th_4)*(sin(th0_3 +
      th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
      th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
1025 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1 +
      th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))) - a_6*sin(th0_6 +
      th_6)*(cos(th0_5 + th_5)*sin(al_5)*(sin(al_4)*(sin(th0_3 +
      th_3)*sin(al_3)*(cos(th0_2 +
1026 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) -
      cos(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +

```

```

th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
1027 th_3)*sin(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
1028 th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)))
- cos(th0_4 +
1029 th_4)*cos(al_4)*(sin(al_3)*(cos(th0_1 + th_1)*cos(al_2)*sin(al_1) - sin(th0_1 +
th_1)*sin(th0_2 + th_2)*sin(al_2) + cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 + th_3)*cos(al_3)*(cos(th0_2 +
th_2)*sin(th0_1 + th_1) + cos(th0_1 +
1030 th_1)*sin(th0_2 + th_2)*cos(al_1)) + cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)))) -
cos(al_5)*(cos(al_4)*(sin(th0_3 +
1031 th_3)*sin(al_3)*(cos(th0_2 + th_2)*sin(th0_1 + th_1) + cos(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_1)) - cos(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 +
1032 th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*sin(al_3)*(sin(th0_1 +
th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) -
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2))) - sin(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 +
1033 th_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) + cos(th0_1 +
th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2 + th_2)*sin(th0_1
+ th_1) + cos(th0_1 + th_1)*sin(th0_2 +
1034 th_2)*cos(al_1))) + cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 +
1035 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)))) +
1036 sin(th0_5 + th_5)*sin(al_5)*(sin(th0_4 + th_4)*(sin(al_3)*(cos(th0_1 +
th_1)*cos(al_2)*sin(al_1) - sin(th0_1 + th_1)*sin(th0_2 + th_2)*sin(al_2) +
cos(th0_1 + th_1)*cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_3 +
th_3)*cos(al_3)*(cos(th0_2 +
1037 th_2)*sin(th0_1 + th_1) + cos(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_1)) +
cos(th0_3 + th_3)*cos(al_3)*(sin(th0_1 + th_1)*sin(th0_2 + th_2)*cos(al_2) +
cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 + th_1)*cos(th0_2 +
th_2)*cos(al_1)*cos(al_2))) +
1038 cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(sin(th0_1 + th_1)*sin(th0_2 +
th_2)*cos(al_2) + cos(th0_1 + th_1)*sin(al_1)*sin(al_2) - cos(th0_1 +
th_1)*cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*(cos(th0_2
+ th_2)*sin(th0_1 + th_1) + cos(th0_1 +

```

```

1039 th_1)*sin(th0_2 + th_2)*cos(al_1)))) ;
1040
1041 J_out(2,24)= 0 ;
1042
1043 J_out(3,1)= 0 ;
1044
1045 J_out(3,2)= d_4*(cos(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3) +
      sin(th0_2 + th_2)*cos(al_3)*sin(al_1)*sin(al_2) + cos(th0_3 +
      th_3)*sin(th0_2 + th_2)*cos(al_2)*sin(al_1)*sin(al_3)) +
      d_5*(cos(al_4)*(cos(th0_2 + th_2)*sin(th0_3 +
1046 th_3)*sin(al_1)*sin(al_3) + sin(th0_2 + th_2)*cos(al_3)*sin(al_1)*sin(al_2) +
      cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_2)*sin(al_1)*sin(al_3)) +
      sin(th0_4 + th_4)*sin(al_4)*(cos(th0_2 + th_2)*cos(th0_3 + th_3)*sin(al_1) -
      sin(th0_2 + th_2)*sin(th0_3 +
1047 th_3)*cos(al_2)*sin(al_1)) + cos(th0_4 + th_4)*sin(al_4)*(cos(th0_2 +
      th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1) - sin(th0_2 +
      th_2)*sin(al_1)*sin(al_2)*sin(al_3) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_2)*cos(al_3)*sin(al_1))) +
1048 d_6*(cos(al_5)*(cos(al_4)*(cos(th0_2 + th_2)*sin(th0_3 +
      th_3)*sin(al_1)*sin(al_3) + sin(th0_2 + th_2)*cos(al_3)*sin(al_1)*sin(al_2)
      + cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_2)*sin(al_1)*sin(al_3)) +
      sin(th0_4 + th_4)*sin(al_4)*(cos(th0_2 +
1049 th_2)*cos(th0_3 + th_3)*sin(al_1) - sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_2)*sin(al_1)) + cos(th0_4 + th_4)*sin(al_4)*(cos(th0_2 +
      th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1) - sin(th0_2 +
      th_2)*sin(al_1)*sin(al_2)*sin(al_3) + cos(th0_3 +
1050 th_3)*sin(th0_2 + th_2)*cos(al_2)*cos(al_3)*sin(al_1))) + sin(th0_5 +
      th_5)*sin(al_5)*(cos(th0_4 + th_4)*(cos(th0_2 + th_2)*cos(th0_3 +
      th_3)*sin(al_1) - sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_2)*sin(al_1)) -
      sin(th0_4 + th_4)*(cos(th0_2 +
1051 th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1) - sin(th0_2 +
      th_2)*sin(al_1)*sin(al_2)*sin(al_3) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_2)*cos(al_3)*sin(al_1))) + cos(th0_5 +
      th_5)*sin(al_5)*(sin(th0_4 + th_4)*cos(al_4)*(cos(th0_2 + th_2)*cos(th0_3 +
1052 th_3)*sin(al_1) - sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_2)*sin(al_1)) -
      sin(al_4)*(cos(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3) +
      sin(th0_2 + th_2)*cos(al_3)*sin(al_1)*sin(al_2) + cos(th0_3 +
      th_3)*sin(th0_2 +
1053 th_2)*cos(al_2)*sin(al_1)*sin(al_3)) + cos(th0_4 + th_4)*cos(al_4)*(cos(th0_2 +
      th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1) - sin(th0_2 +
      th_2)*sin(al_1)*sin(al_2)*sin(al_3) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_2)*cos(al_3)*sin(al_1))) -
1054 a_5*sin(th0_5 + th_5)*(sin(th0_4 + th_4)*cos(al_4)*(cos(th0_2 + th_2)*cos(th0_3
      + th_3)*sin(al_1) - sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_2)*sin(al_1))
      - sin(al_4)*(cos(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3) +
      sin(th0_2 +
1055 th_2)*cos(al_3)*sin(al_1)*sin(al_2) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_2)*sin(al_1)*sin(al_3)) + cos(th0_4 +
      th_4)*cos(al_4)*(cos(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1) -
      sin(th0_2 + th_2)*sin(al_1)*sin(al_2)*sin(al_3) + cos(th0_3 +

```

```

1056 th_3)*sin(th0_2 + th_2)*cos(al_2)*cos(al_3)*sin(al_1))) + a_2*cos(th0_2 +
      th_2)*sin(al_1) - a_6*sin(th0_6 + th_6)*(sin(th0_5 +
      th_5)*cos(al_5)*(cos(th0_4 + th_4)*(cos(th0_2 + th_2)*cos(th0_3 +
      th_3)*sin(al_1) - sin(th0_2 + th_2)*sin(th0_3 +
1057 th_3)*cos(al_2)*sin(al_1)) - sin(th0_4 + th_4)*(cos(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_3)*sin(al_1) - sin(th0_2 + th_2)*sin(al_1)*sin(al_2)*sin(al_3)
      + cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_2)*cos(al_3)*sin(al_1))) -
1058 sin(al_5)*(cos(al_4)*(cos(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3) +
      sin(th0_2 + th_2)*cos(al_3)*sin(al_1)*sin(al_2) + cos(th0_3 +
      th_3)*sin(th0_2 + th_2)*cos(al_2)*sin(al_1)*sin(al_3)) + sin(th0_4 +
      th_4)*sin(al_4)*(cos(th0_2 + th_2)*cos(th0_3
1059 + th_3)*sin(al_1) - sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_2)*sin(al_1)) +
      cos(th0_4 + th_4)*sin(al_4)*(cos(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_3)*sin(al_1) - sin(th0_2 + th_2)*sin(al_1)*sin(al_2)*sin(al_3)
      + cos(th0_3 + th_3)*sin(th0_2 +
1060 th_2)*cos(al_2)*cos(al_3)*sin(al_1))) + cos(th0_5 + th_5)*cos(al_5)*(sin(th0_4 +
      th_4)*cos(al_4)*(cos(th0_2 + th_2)*cos(th0_3 + th_3)*sin(al_1) - sin(th0_2 +
      th_2)*sin(th0_3 + th_3)*cos(al_2)*sin(al_1)) - sin(al_4)*(cos(th0_2 +
      th_2)*sin(th0_3 +
1061 th_3)*sin(al_1)*sin(al_3) + sin(th0_2 + th_2)*cos(al_3)*sin(al_1)*sin(al_2) +
      cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_2)*sin(al_1)*sin(al_3)) +
      cos(th0_4 + th_4)*cos(al_4)*(cos(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_3)*sin(al_1) - sin(th0_2 +
1062 th_2)*sin(al_1)*sin(al_2)*sin(al_3) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_2)*cos(al_3)*sin(al_1))) + a_4*cos(th0_4 + th_4)*(cos(th0_2 +
      th_2)*cos(th0_3 + th_3)*sin(al_1) - sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_2)*sin(al_1)) + a_5*cos(th0_5 +
1063 th_5)*(cos(th0_4 + th_4)*(cos(th0_2 + th_2)*cos(th0_3 + th_3)*sin(al_1) -
      sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_2)*sin(al_1)) - sin(th0_4 +
      th_4)*(cos(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1) - sin(th0_2 +
1064 th_2)*sin(al_1)*sin(al_2)*sin(al_3) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_2)*cos(al_3)*sin(al_1))) - a_4*sin(th0_4 + th_4)*(cos(th0_2 +
      th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1) - sin(th0_2 +
      th_2)*sin(al_1)*sin(al_2)*sin(al_3) + cos(th0_3 +
1065 th_3)*sin(th0_2 + th_2)*cos(al_2)*cos(al_3)*sin(al_1)) + a_6*cos(th0_6 +
      th_6)*(cos(th0_5 + th_5)*(cos(th0_4 + th_4)*(cos(th0_2 + th_2)*cos(th0_3 +
      th_3)*sin(al_1) - sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_2)*sin(al_1)) -
      sin(th0_4 + th_4)*(cos(th0_2 +
1066 th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1) - sin(th0_2 +
      th_2)*sin(al_1)*sin(al_2)*sin(al_3) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_2)*cos(al_3)*sin(al_1))) - sin(th0_5 + th_5)*(sin(th0_4 +
      th_4)*cos(al_4)*(cos(th0_2 + th_2)*cos(th0_3 +
1067 th_3)*sin(al_1) - sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_2)*sin(al_1)) -
      sin(al_4)*(cos(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3) +
      sin(th0_2 + th_2)*cos(al_3)*sin(al_1)*sin(al_2) + cos(th0_3 +
      th_3)*sin(th0_2 +
1068 th_2)*cos(al_2)*sin(al_1)*sin(al_3)) + cos(th0_4 + th_4)*cos(al_4)*(cos(th0_2 +
      th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1) - sin(th0_2 +
      th_2)*sin(al_1)*sin(al_2)*sin(al_3) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_2)*cos(al_3)*sin(al_1))) +

```

```

1069 d_3*sin(th0_2 + th_2)*sin(al_1)*sin(al_2) + a_3*cos(th0_2 + th_2)*cos(th0_3 +
      th_3)*sin(al_1) - a_3*sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_2)*sin(al_1) ;
1070
1071 J_out(3,3)= d_4*(sin(th0_3 + th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*sin(al_1)*sin(al_3)) + d_5*(cos(al_4)*(sin(th0_3 +
      th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1072 th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*sin(al_4)*(cos(th0_3 +
      th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) -
      sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)) +
1073 cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
      cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_3)*sin(al_1))) + d_6*(cos(al_5)*(cos(al_4)*(sin(th0_3 +
1074 th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
      cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)*sin(al_3)) + sin(th0_4 +
      th_4)*sin(al_4)*(cos(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) -
1075 sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)) + cos(th0_4 +
      th_4)*sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
      cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_3)*sin(al_1))) + sin(th0_5 +
1076 th_5)*sin(al_5)*(cos(th0_4 + th_4)*(cos(th0_3 + th_3)*(cos(al_1)*sin(al_2) +
      cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*sin(al_1)) - sin(th0_4 + th_4)*(sin(th0_3 +
      th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1077 th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_3)*sin(al_1))) + cos(th0_5 + th_5)*sin(al_5)*(cos(th0_4 +
      th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
      cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
1078 th_3)*sin(th0_2 + th_2)*cos(al_3)*sin(al_1)) - sin(al_4)*(sin(th0_3 +
      th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*cos(al_4)*(cos(th0_3 +
1079 th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2
      + th_2)*sin(th0_3 + th_3)*sin(al_1))) - a_4*sin(th0_4 + th_4)*(sin(th0_3 +
      th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
1080 th_3)*sin(th0_2 + th_2)*cos(al_3)*sin(al_1)) - a_5*sin(th0_5 + th_5)*(cos(th0_4
      + th_4)*cos(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
      cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_3)*sin(al_1)) -
1081 sin(al_4)*(sin(th0_3 + th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*cos(al_4)*(cos(th0_3 +
      th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1082 th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1))) -
      a_6*sin(th0_6 + th_6)*(sin(th0_5 + th_5)*cos(al_5)*(cos(th0_4 +
      th_4)*(cos(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +

```

```

th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 +
1083 th_2)*sin(th0_3 + th_3)*sin(al_1)) - sin(th0_4 + th_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*cos(al_3)*sin(al_1))) - sin(al_5)*(cos(al_4)*(sin(th0_3 +
1084 th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)*sin(al_3)) + sin(th0_4 +
th_4)*sin(al_4)*(cos(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) -
1085 sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)) + cos(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*cos(al_3)*sin(al_1))) + cos(th0_5 +
1086 th_5)*cos(al_5)*(cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*cos(al_3)*sin(al_1)) - sin(al_4)*(sin(th0_3 +
1087 th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)*sin(al_3)) + sin(th0_4 +
th_4)*cos(al_4)*(cos(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) -
1088 sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)))) + a_3*cos(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
a_4*cos(th0_4 + th_4)*(cos(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 +
1089 th_2)*sin(th0_3 + th_3)*sin(al_1)) + a_5*cos(th0_5 + th_5)*(cos(th0_4 +
th_4)*(cos(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1))
- sin(th0_4 + th_4)*(sin(th0_3 +
1090 th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_3)*sin(al_1))) + a_6*cos(th0_6 +
th_6)*(cos(th0_5 + th_5)*(cos(th0_4 + th_4)*(cos(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1091 th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)) -
sin(th0_4 + th_4)*(sin(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*cos(al_3)*sin(al_1))) -
1092 sin(th0_5 + th_5)*(cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*cos(al_3)*sin(al_1)) - sin(al_4)*(sin(th0_3 +
1093 th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)*sin(al_3)) + sin(th0_4 +
th_4)*cos(al_4)*(cos(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) -
1094 sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)))) - a_3*sin(th0_2 +
th_2)*sin(th0_3 + th_3)*sin(al_1) ;
1095
1096 J_out(3,4)= d_6*(cos(al_5)*(sin(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +

```

```

th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
1097 th_3)*cos(al_3)*sin(al_1)) + cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1))) - sin(th0_5 +
th_5)*sin(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 +
1098 th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3
+ th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
1099 th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) -
sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) + cos(th0_5 +
th_5)*sin(al_5)*(sin(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2)
- cos(th0_2 +
1100 th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) + cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) +
1101 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1))) + d_5*(sin(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
1102 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) + cos(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 +
1103 th_2)*sin(al_1))) + a_6*sin(th0_6 + th_6)*(sin(al_5)*(sin(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) -
1104 sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1)) + cos(th0_4 +
th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)))
+ sin(th0_5 +
1105 th_5)*cos(al_5)*(sin(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1)) - cos(th0_4 + th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
cos(th0_2 +
1106 th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1))) - cos(th0_5 + th_5)*cos(al_5)*(sin(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) + cos(th0_4 +
1107 th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) + cos(th0_4 +
1108 th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +

```

```

th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1))) + a_4*cos(th0_4 + th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) +
1109 cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) - a_6*cos(th0_6 + th_6)*(sin(th0_5 +
th_5)*(sin(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
1110 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) + cos(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
1111 th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3
+ th_3)*sin(th0_2 + th_2)*sin(al_1))) + cos(th0_5 + th_5)*(sin(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
1112 th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
1113 th_3)*cos(al_3)*sin(al_1))) - a_5*sin(th0_5 + th_5)*(sin(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) -
1114 sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)))
- a_4*sin(th0_4 + th_4)*(sin(th0_3 +
1115 th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3
+ th_3)*sin(th0_2 + th_2)*sin(al_1)) - a_5*cos(th0_5 + th_5)*(sin(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
1116 th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
1117 th_3)*cos(al_3)*sin(al_1))) ;
1118
1119 J_out(3,5)= d_6*(cos(th0_5 + th_5)*sin(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
1120 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1))) + sin(th0_5 +
1121 th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +

```

```

th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) +
1122 cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) -
1123 sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2
+ th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1))) + a_6*cos(th0_6 + th_6)*(cos(th0_5 +
th_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) -
1124 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
1125 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
1126 th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3
+ th_3)*sin(th0_2 + th_2)*sin(al_1))) - sin(th0_5 + th_5)*(cos(th0_4 +
th_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
1127 th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
1128 th_3)*cos(al_3)*sin(al_1))) + a_5*cos(th0_5 +
th_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 +
1129 th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 +
1130 th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 +
th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)))
- a_5*sin(th0_5 + th_5)*(cos(th0_4 +
1131 th_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1))
+ sin(th0_4 + th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
1132 th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) -
sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) - a_6*sin(th0_6 +

```

```

th_6)*(cos(th0_5 + th_5)*cos(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) +
1133 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1)) + sin(th0_4 + th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1134 th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1))) + sin(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) +
1135 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
1136 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1)))) ;
1137
1138
1139 J_out(3,6)= a_6*cos(th0_6 +
th_6)*(sin(al_5)*(cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
1140 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1))) - sin(th0_5 + th_5)*cos(al_5)*(cos(th0_4 +
th_4)*(sin(th0_3 +
1141 th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3
+ th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
1142 th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) -
sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) + cos(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) -
1143 cos(th0_3 + th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) +
1144 cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) +
1145 cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +

```

```

th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1146 th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)))) -
a_6*sin(th0_6 + th_6)*(sin(th0_5 +
th_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(cos(al_1)*sin(al_2)
1147 + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
1148 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 +
1149 th_2)*sin(al_1))) + cos(th0_5 + th_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
1150 th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)))) ;

1151
1152 J_out(3,7)= 1 ;
1153
1154 J_out(3,8)= cos(al_1) ;
1155
1156 J_out(3,9)= cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(al_1)*sin(al_2) ;
1157
1158 J_out(3,10)= cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3) ;

1159
1160 J_out(3,11)= cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) +
1161 sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2
+ th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1)) - cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
1162 th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) ;

1163
1164 J_out(3,12)= cos(al_5)*(cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +

```

```

th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
1165 th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
1166 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1))) + sin(th0_5 + th_5)*sin(al_5)*(cos(th0_4 +
th_4)*(sin(th0_3 +
1167 th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3
+ th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
1168 th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) -
sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) - cos(th0_5 +
th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) -
1169 cos(th0_3 + th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) +
1170 cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1171 th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1))) ;
1172
1173 J_out(3,13)= 0 ;
1174
1175 J_out(3,14)= sin(th0_2 + th_2)*sin(al_1) ;
1176
1177 J_out(3,15)= sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1) ;
1178
1179 J_out(3,16)= cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1)) + sin(th0_4 + th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) +
1180 cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) ;
1181
1182 J_out(3,17)= sin(th0_5 + th_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) -
cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +

```

```

1183 th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
      th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
1184 th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
      th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
      cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1))) + cos(th0_5 +
      th_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
1185 th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3
      + th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
      th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
1186 th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) -
      sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) ;
1187
1188 J_out(3,18)= sin(th0_6 +
      th_6)*(sin(al_5)*(cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
      th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
1189 th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
      th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
      cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
      th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
1190 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_3)*sin(al_1))) - sin(th0_5 + th_5)*cos(al_5)*(cos(th0_4 +
      th_4)*(sin(th0_3 +
1191 th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3
      + th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
      th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
1192 th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) -
      sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) + cos(th0_5 +
      th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) -
1193 cos(th0_3 + th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
      th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) +
1194 cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
      th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1195 th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1))) +
      cos(th0_6 + th_6)*(sin(th0_5 +
      th_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +

```

```

th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) +
1196 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
1197 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 +
1198 th_2)*sin(al_1))) + cos(th0_5 + th_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
1199 th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)))) ;

1200
1201 J_out(3,19)= d_4*(cos(th0_3 + th_3)*sin(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_1)*sin(al_3)) -
1202 d_6*(sin(th0_5 + th_5)*sin(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(sin(al_1)*sin(al_2) - cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) -
cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_1)) + sin(th0_4 +
th_4)*(sin(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 +
1203 th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(sin(al_1)*sin(al_2) -
cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_1)*cos(al_3))) - cos(al_5)*(cos(al_4)*(cos(th0_3 +
th_3)*sin(al_3)*(sin(al_1)*sin(al_2) -
1204 cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(al_3)*(cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 +
1205 th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(sin(al_1)*sin(al_2) -
cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_1)*cos(al_3)) - sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(sin(al_1)*sin(al_2) -
1206 cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(th0_2 +
th_2)*cos(al_1))) + cos(th0_5 + th_5)*sin(al_5)*(sin(al_4)*(cos(th0_3 +
th_3)*sin(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(al_3)*(cos(al_2)*sin(al_1) +
1207 cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_1)*sin(al_3)) - cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(al_1)*sin(al_2) -
1208 cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_1)*cos(al_3)) + sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +

```

```

th_3)*(sin(al_1)*sin(al_2) - cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) -
cos(th0_3 + th_3)*sin(th0_2 +
1209 th_2)*cos(al_1)))) - d_2*sin(al_1) + d_5*(cos(al_4)*(cos(th0_3 +
th_3)*sin(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
1210 th_3)*cos(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
1211 th_3)*cos(al_1)*cos(al_3)) - sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(sin(al_1)*sin(al_2) - cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) -
cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_1))) - d_3*(cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*cos(al_1)*sin(al_2))
1212 - a_6*cos(th0_6 + th_6)*(cos(th0_5 + th_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(sin(al_1)*sin(al_2) - cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) -
cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_1)) + sin(th0_4 +
th_4)*(sin(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2
1213 + th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(sin(al_1)*sin(al_2)
- cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_1)*cos(al_3))) - sin(th0_5 + th_5)*(sin(al_4)*(cos(th0_3 +
th_3)*sin(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) -
cos(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_1)*sin(al_3)) - cos(th0_4 +
1214 th_3)*sin(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) -
cos(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_1)*sin(al_3)) - cos(th0_4 +
1215 th_4)*cos(al_4)*(sin(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_1)*cos(al_3)) + sin(th0_4 +
1216 th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(th0_2 +
th_2)*cos(al_1))) - a_4*sin(th0_4 + th_4)*(sin(al_3)*(cos(al_2)*sin(al_1) +
cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
1217 cos(th0_3 + th_3)*cos(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_1)*cos(al_3)) + a_2*sin(th0_2 + th_2)*cos(al_1) + a_5*sin(th0_5
+ th_5)*(sin(al_4)*(cos(th0_3 +
1218 th_3)*sin(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) -
cos(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_1)*sin(al_3)) - cos(th0_4 +
1219 th_4)*cos(al_4)*(sin(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 +
th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_1)*cos(al_3)) + sin(th0_4 +
1220 th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_1)))
- a_4*cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(sin(al_1)*sin(al_2) - cos(th0_2
+ th_2)*cos(al_1)*cos(al_2)) -

```

```

1221 cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_1)) - a_3*sin(th0_3 +
      th_3)*(sin(al_1)*sin(al_2) - cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) -
      a_5*cos(th0_5 + th_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
      th_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
1222 th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_1)) +
      sin(th0_4 + th_4)*(sin(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 +
      th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
1223 th_2)*cos(al_1)*cos(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_1)*cos(al_3))) + a_6*sin(th0_6 +
      th_6)*(sin(al_5)*(cos(al_4)*(cos(th0_3 +
      th_3)*sin(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
      th_2)*cos(al_1)*cos(al_2)) -
1224 cos(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
      sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_1)*sin(al_3)) + cos(th0_4 +
      th_4)*sin(al_4)*(sin(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 +
      th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
1225 th_3)*cos(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) +
      sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_1)*cos(al_3)) - sin(th0_4 +
      th_4)*sin(al_4)*(sin(th0_3 + th_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
      th_2)*cos(al_1)*cos(al_2)) -
1226 cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_1))) + sin(th0_5 +
      th_5)*cos(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 + th_3)*(sin(al_1)*sin(al_2) -
      cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_1)) + sin(th0_4 +
1227 th_4)*(sin(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) +
      cos(th0_3 + th_3)*cos(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
      th_2)*cos(al_1)*cos(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_1)*cos(al_3))) + cos(th0_5 +
1228 th_5)*cos(al_5)*(sin(al_4)*(cos(th0_3 + th_3)*sin(al_3)*(sin(al_1)*sin(al_2) -
      cos(th0_2 + th_2)*cos(al_1)*cos(al_2)) - cos(al_3)*(cos(al_2)*sin(al_1) +
      cos(th0_2 + th_2)*cos(al_1)*sin(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_1)*sin(al_3)) -
1229 cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(al_2)*sin(al_1) + cos(th0_2 +
      th_2)*cos(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(sin(al_1)*sin(al_2) - cos(th0_2 +
      th_2)*cos(al_1)*cos(al_2)) + sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_1)*cos(al_3)) +
1230 sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(sin(al_1)*sin(al_2) - cos(th0_2
      + th_2)*cos(al_1)*cos(al_2)) - cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*cos(al_1))) + a_3*cos(th0_3 + th_3)*sin(th0_2 + th_2)*cos(al_1) ;
1231
1232 J_out(3,20)= d_6*(cos(al_5)*(cos(th0_4 +
      th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 +
      th_3)*cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2))) -
1233 cos(al_4)*(cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
      th_3)*sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +

```

```

th_2)*sin(al_1)*sin(al_2))) + sin(th0_3 + th_3)*sin(th0_4 +
th_4)*sin(al_4)*(cos(al_1)*cos(al_2) -
1234 cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) - sin(th0_5 + th_5)*sin(al_5)*(sin(th0_4
+ th_4)*(sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)))
1235 - cos(th0_4 + th_4)*sin(th0_3 + th_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2))) + cos(th0_5 +
th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
1236 th_3)*sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) +
cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
1237 th_2)*sin(al_1)*sin(al_2))) + sin(th0_3 + th_3)*sin(th0_4 +
th_4)*cos(al_4)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2))) - d_4*(cos(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
1238 th_3)*sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) +
d_5*(cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2
+ th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
1239 th_2)*sin(al_1)*sin(al_2))) - cos(al_4)*(cos(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2))) + sin(th0_3 + th_3)*sin(th0_4 +
1240 th_4)*sin(al_4)*(cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) -
d_3*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) -
a_5*cos(th0_5 + th_5)*(sin(th0_4 + th_4)*(sin(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 +
1241 th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*cos(al_2) -
cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) - cos(th0_4 + th_4)*sin(th0_3 +
th_3)*(cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) -
a_5*sin(th0_5 +
1242 th_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2))) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*sin(al_2) +
1243 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2))) + sin(th0_3 + th_3)*sin(th0_4 +
th_4)*cos(al_4)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2))) -
1244 a_6*cos(th0_6 + th_6)*(cos(th0_5 + th_5)*(sin(th0_4 +
th_4)*(sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2))) - cos(th0_4 + th_4)*sin(th0_3

```

```

1245 + th_3)*(cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) +
      sin(th0_5 + th_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
      th_3)*sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
1246 th_2)*sin(al_1)*sin(al_2))) + cos(th0_4 +
      th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 +
      th_3)*cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2))) + sin(th0_3 + th_3)*sin(th0_4 +
1247 th_4)*cos(al_4)*(cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(al_1)*sin(al_2)))
      + a_3*sin(th0_3 + th_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) - a_4*sin(th0_4 +
      th_4)*(sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1248 th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*cos(al_2) -
      cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) + a_6*sin(th0_6 +
      th_6)*(sin(al_5)*(cos(th0_4 +
      th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1249 th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*cos(al_2) -
      cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) -
      cos(al_4)*(cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(al_3)*(cos(al_1)*cos(al_2)
1250 - cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) + sin(th0_3 + th_3)*sin(th0_4 +
      th_4)*sin(al_4)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2))) + sin(th0_5 + th_5)*cos(al_5)*(sin(th0_4 +
      th_4)*(sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1251 th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*cos(al_2) -
      cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) - cos(th0_4 + th_4)*sin(th0_3 +
      th_3)*(cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(al_1)*sin(al_2))) -
      cos(th0_5 +
1252 th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 +
      th_3)*sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2))) + cos(th0_4 +
      th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*sin(al_2)
1253 + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - cos(th0_3 +
      th_3)*cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2))) + sin(th0_3 + th_3)*sin(th0_4 +
      th_4)*cos(al_4)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)))) +
1254 a_4*cos(th0_4 + th_4)*sin(th0_3 + th_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) ;
1255
1256 J_out(3,21)= d_6*(cos(th0_5 +
      th_5)*sin(al_5)*(sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
1257 th_3)*cos(al_3)*sin(al_1)) - cos(th0_4 +
      th_4)*cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +

```

```

th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
1258 th_3)*sin(al_1)*sin(al_3))) -
cos(al_5)*(cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
1259 th_3)*cos(al_3)*sin(al_1)) + cos(th0_4 +
th_4)*sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
1260 th_3)*sin(al_1)*sin(al_3))) + sin(th0_4 + th_4)*sin(th0_5 +
th_5)*sin(al_5)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 +
1261 th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3))) -
d_4*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 + th_2)*sin(al_1)*sin(al_2))
+ cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
1262 th_3)*cos(al_3)*sin(al_1)) - d_5*(cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
1263 th_3)*cos(al_3)*sin(al_1)) + cos(th0_4 +
th_4)*sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
1264 th_3)*sin(al_1)*sin(al_3))) + a_4*sin(th0_4 +
th_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
1265 th_3)*sin(al_1)*sin(al_3)) - a_6*sin(th0_6 +
th_6)*(sin(al_5)*(cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 +
1266 th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1)) + cos(th0_4 +
th_4)*sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 +
1267 th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3))) + cos(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) -

```

```

1268 sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1)) - cos(th0_4 +
    th_4)*cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
    th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
    th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
    th_2)*cos(al_2)*sin(al_1)) +
1269 sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3))) + sin(th0_4 +
    th_4)*sin(th0_5 + th_5)*cos(al_5)*(cos(al_3)*(cos(al_1)*cos(al_2) -
    cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
    th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
    th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
1270 th_3)*sin(al_1)*sin(al_3))) - a_5*sin(th0_5 +
    th_5)*(sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
    th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
    th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
1271 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
    th_3)*cos(al_3)*sin(al_1)) - cos(th0_4 +
    th_4)*cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
    th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
    th_3)*sin(al_3)*(cos(al_1)*sin(al_2) +
1272 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
    th_3)*sin(al_1)*sin(al_3))) - a_6*cos(th0_6 + th_6)*(sin(th0_5 +
    th_5)*(sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
    th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
    th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) -
1273 sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1)) - cos(th0_4 +
    th_4)*cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
    th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
    th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
1274 th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
    sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3))) - cos(th0_5 +
    th_5)*sin(th0_4 + th_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
    th_2)*sin(al_1)*sin(al_2)) -
1275 cos(th0_3 + th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
    th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
    th_3)*sin(al_1)*sin(al_3))) + a_5*cos(th0_5 + th_5)*sin(th0_4 +
    th_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
    th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(cos(al_1)*sin(al_2) +
1276 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
    th_3)*sin(al_1)*sin(al_3)) ;

1277
1278 J_out(3,22)= a_5*sin(th0_5 + th_5)*(cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) -
    cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
    th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
    th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
    th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
    th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
    cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
    th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
1280 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
    th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
    th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +

```

```

th_3)*cos(al_3)*sin(al_1))) - d_5*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2)
- cos(th0_2 +
1281 th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
1282 th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) +
1283 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1))) -
d_6*(cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1284 th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1285 th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1))) +
1286 cos(th0_5 + th_5)*sin(al_5)*(cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) -
cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
1287 th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
1288 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1))) - a_6*sin(th0_6 +
1289 th_6)*(sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) +
1290 cos(th0_4 + th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) -
1291 sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2
+ th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
th_2)*sin(al_1))) - cos(th0_5 +
th_5)*cos(al_5)*(cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +

```

```

1292 th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 + th_3)*sin(al_3)*(cos(al_1)*sin(al_2) +
      cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
      th_3)*(cos(al_1)*sin(al_2) +
1293 cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 +
      th_2)*sin(al_1)) - cos(th0_4 +
      th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1294 th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_3)*sin(al_1))) + a_6*cos(th0_6 + th_6)*sin(th0_5 +
      th_5)*(cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
1295 th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
      sin(th0_2 + th_2)*sin(th0_3 + th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 +
      th_4)*sin(al_4)*(sin(th0_3 + th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) +
1296 cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
      th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 +
1297 th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) ;
1298
1299 J_out(3,23)= a_6*sin(th0_6 +
      th_6)*(cos(al_5)*(cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
      th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
1300 th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
      th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
      cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) - cos(th0_4 +
      th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
1301 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
      th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*cos(al_3)*sin(al_1))) + sin(th0_5 + th_5)*sin(al_5)*(cos(th0_4 +
      th_4)*(sin(th0_3 +
1302 th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3
      + th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
      th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
1303 th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) -
      sin(th0_2 + th_2)*sin(th0_3 + th_3)*cos(al_3)*sin(al_1))) - cos(th0_5 +
      th_5)*sin(al_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
      th_2)*sin(al_1)*sin(al_2)) -
1304 cos(th0_3 + th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
      th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
      th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
      th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +

```

```

th_2)*sin(al_1)*sin(al_2)) +
1305 cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1306 th_2)*cos(al_2)*sin(al_1)) + cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1))) -
d_6*(sin(al_5)*(cos(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
1307 th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + sin(th0_4 + th_4)*sin(al_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) -
1308 cos(th0_4 + th_4)*sin(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1))) -
1309 sin(th0_5 + th_5)*cos(al_5)*(cos(th0_4 + th_4)*(sin(th0_3 +
th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) +
cos(th0_3 + th_3)*sin(th0_2 + th_2)*sin(al_1)) + sin(th0_4 +
th_4)*(sin(al_3)*(cos(al_1)*cos(al_2) - cos(th0_2 +
1310 th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 + th_3)*cos(al_3)*(cos(al_1)*sin(al_2) +
cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1))) + cos(th0_5 +
th_5)*cos(al_5)*(sin(al_4)*(cos(al_3)*(cos(al_1)*cos(al_2) -
1311 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) - cos(th0_3 +
th_3)*sin(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) + sin(th0_2 + th_2)*sin(th0_3 +
th_3)*sin(al_1)*sin(al_3)) + cos(th0_4 +
th_4)*cos(al_4)*(sin(al_3)*(cos(al_1)*cos(al_2) -
1312 cos(th0_2 + th_2)*sin(al_1)*sin(al_2)) + cos(th0_3 +
th_3)*cos(al_3)*(cos(al_1)*sin(al_2) + cos(th0_2 +
th_2)*cos(al_2)*sin(al_1)) - sin(th0_2 + th_2)*sin(th0_3 +
th_3)*cos(al_3)*sin(al_1)) - sin(th0_4 + th_4)*cos(al_4)*(sin(th0_3 +
1313 th_3)*(cos(al_1)*sin(al_2) + cos(th0_2 + th_2)*cos(al_2)*sin(al_1)) + cos(th0_3
+ th_3)*sin(th0_2 + th_2)*sin(al_1))) ;

1314
1315 J_out(3,24)= 0 ;
1316
1317 return

```