

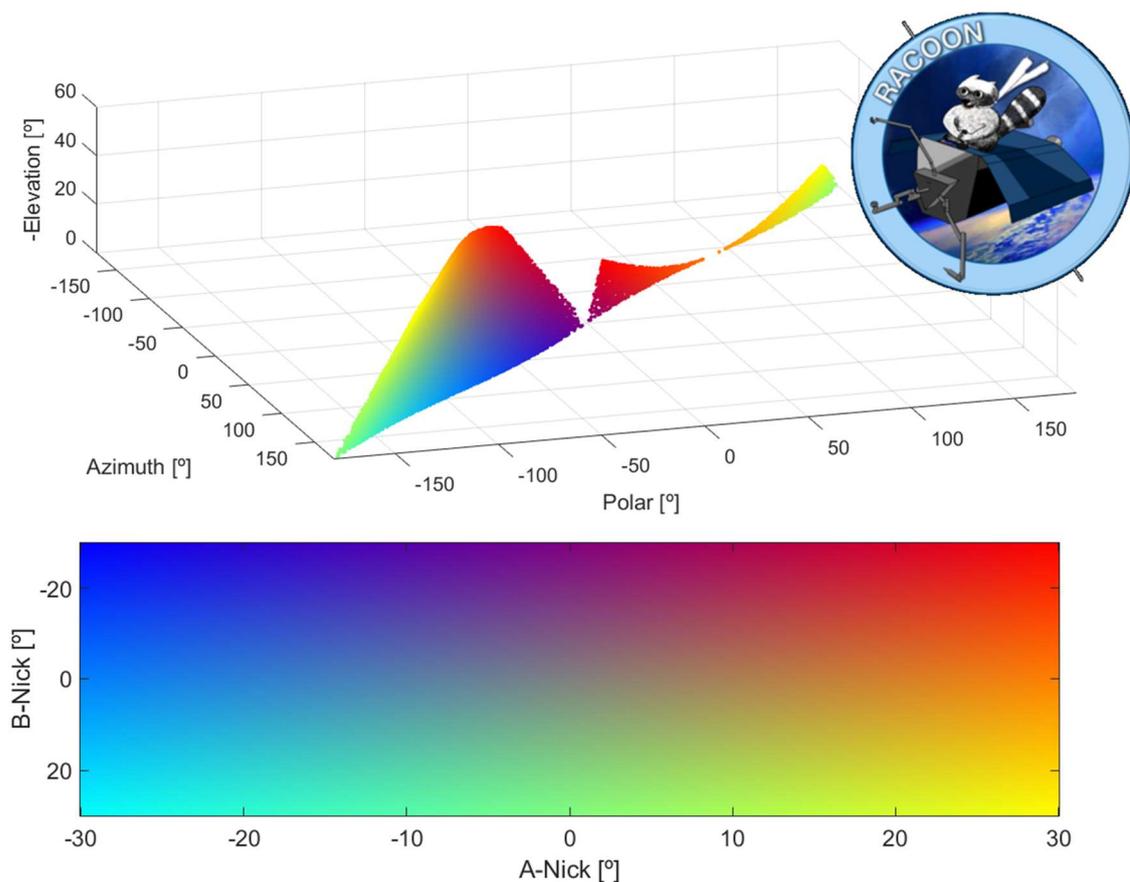
Master's Thesis

Development of an Inverse Kinematic Algorithm for Simulation of Spacecraft Attitudes in the RACOON-Lab

RT-MA 2020/06

Author:

Daniel Martínez



Supervisor:

Jonis Kiesbye
 Lehrstuhl für Raumfahrttechnik / Institute of Astronautics
 Technische Universität München

LRT-Nummer: 2020/06

Titel der Arbeit: Development of an Inverse Kinematic Algorithm for Simulation
of Spacecraft Attitudes in the RACOON-Lab

Autor: Daniel Martínez

Matrikelnummer: 03694851

Erklärung

Mir als vertraulich genannte Informationen, Unterlagen und Erkenntnisse werde ich nach meiner Tätigkeit am Lehrstuhl nicht an Dritte weitergeben.

Ich erkläre mich außerdem damit einverstanden, dass meine Bachelor-, Semester-, Master-/Diplomarbeit vom Lehrstuhl auf Anfrage fachlich interessierten Personen, auch über eine Bibliothek, zugänglich gemacht wird, und dass darin enthaltene Ergebnisse sowie dabei entstandene Entwicklungen und Programme vom Lehrstuhl für Raumfahrttechnik uneingeschränkt genutzt werden dürfen. (Rechte an evtl. entstehenden Programmen und Erfindungen müssen im Vorfeld geklärt werden.)

Ich erkläre außerdem, dass ich diese Arbeit ohne fremde Hilfe angefertigt und nur die in dem Literaturverzeichnis angeführten Quellen und Hilfsmittel benutzt habe.

Garching, den 27.05.2020



Unterschrift

Acknowledgments

The Master's Thesis represents a major milestone in one's education and is the consequence of many years of work and sacrifices. In spite of this endeavors' individual nature, it would not be achievable without the external support of people. Such help is what makes the difference in the most demanding moments and it is a hidden force that enables the realization of big projects. I would like to use these lines to express my gratitude towards the people who supported me in reaching this point.

In this direction, I would like to thank my family and friends for their continuous support during my studies. They always backed me in the most demanding moments and inspired me to continue working with renewed strength. In a more specific way, I would like to thank M.Sc. Jonis Kiesbye from the department of Astronautics of the Technical University of Munich for his guidance and providing me the opportunity to work in this challenging project. He consistently allowed me to develop my creativity, but steered me in the right the direction whenever he thought It was necessary. Without his patience and engagement, this work's quality would be much lower.

Zusammenfassung

Das RACOON-Lab ist ein Robotersystem mit überdurchschnittlich komplexer Kinematik aufgrund seiner Anforderungen an die Sichtbarkeit. Diese Anlage soll Proximity Operations im Orbit simulieren, um Bilderkennungsalgorithmen zu testen. Aus diesem Grund wurde seine Geometrie mit der Absicht konzipiert, eine ungestörte Sichtverbindung zu erreichen. Folglich ist die Kinematik der beteiligten Roboter durch einen geringen Redundanzgrad und starke Einschränkungen in den vorhandenen Gelenken gekennzeichnet. Diese starke Beschränkung der Bewegungsmöglichkeiten hat einen nicht konvexen Lösungsraum mit mehreren Singularitäten zur Folge, was die Nutzung der Anlage in den letzten Jahren eingeschränkt hat. Die vorliegende Arbeit befasste sich mit dem inversen Kinematikproblem des RACOON-Labs unter Verwendung konventioneller Ansätze und entwickelte einen neuen Algorithmus. In Bezug auf die gewöhnlichen Ansätze wurden Automatic Supervisory Control (ASC) und Weighted Least Norm (WLN) als potenzielle Lösungen evaluiert. Darüber hinaus wurde ein neuartiger Ansatz entwickelt, der auf der analytischen Lösung der inversen Kinematik basiert. In diesem Fall wird die isokinematische Mannigfaltigkeit an jedem Punkt der Bahn berechnet und zur Erzeugung der Trajektorie des Roboters verwendet. Alle Algorithmen wurden auf zwei Fälle angewendet. Zunächst wurde ein reines Rollen untersucht. Diese Bewegung ist besonders relevant, da sie sich in früheren RACOON-Lab-Studien als äußerst problematisch erwiesen hat. Die andere betrachtete Flugbahn war das Taumeln eines im All befindlichen Satelliten: MOVE-II. Die Rotation dieses Raumfahrzeugs wurde rekonstruiert, nachdem die von seinen Gyroskopen gesammelten Daten über einen Zeitraum verarbeitet wurden, in dem MOVE-II ohne Aktuierung taumelte. Die konventionellen Ansätze lieferten einen begrenzten Erfolg bei der Ausführung der Rollbewegung. In diesem Fall war es notwendig, auf hohe Verstärkungskoeffizienten zurückzugreifen, um eine realisierbare Bewegung zu erreichen. Diese hohen Verstärkungskoeffizienten erwiesen sich als wirksam, schienen jedoch gleichzeitig den Wirkungsgrad zu beeinträchtigen und Schwingungen auszulösen. Darüber hinaus konnte die Trajektorie von MOVE-II mit keiner dieser Methoden unabhängig von der Verstärkung vollständig nachempfunden werden. Der auf der analytischen Lösung basierende Ansatz übertraf in beiden Fällen die herkömmlichen Methoden. Für die Rollbewegung wurde ein Wirkungsgrad von 50% erzielt, verglichen mit ungefähr 10%, die durch ASC und WLN erreicht wurden. In Bezug auf die Simulation des Taumelns von MOVE-II konnte der auf der analytischen Lösung basierende Algorithmus diese mit einem Wirkungsgrad von 70% ausführen. Ergänzend zur Betrachtung des Trajektorienproblems wurde die entwickelte inverse Kinematiklösung zur Untersuchung der Fähigkeiten von RACOON-Lab angewandt. Insbesondere wurde die Effizienz des Roboters in Abhängigkeit der geometrischen Randbedingungen untersucht. Dies ermöglichte eine objektive Bestimmung der kinematischen Fähigkeiten des Roboters.

Abstract

The RACOON lab is a robotic system with extremely complex kinematics due to its requirements on visibility. This facility intends to recreate on-orbit proximity operations in order to test visual recognition algorithms. Due to this, its geometry was conceived with the intention of promoting a diaphanous framework. Consequently, the kinematics of the involved robots is characterized by a low degree of redundancy and strong constraints in the existing joints. This severe narrowing of the motion possibilities determines a non-convex solution space with several singularities, which has limited the utilization of the facility over the last years.

The present work addressed the inverse kinematics problem of the RACOON Lab using conventional approaches and by proposing a new algorithm. Regarding the ordinary approaches, Automatic Supervisory Control (ASC) and Weighted Least Norm (WLN) were evaluated as potential solvers. In addition, a novel approach based on the inverse kinematics was developed. In this case, the isokinematic manifold at each point of the trajectory is computed and used for guiding the robot's trajectory.

All these algorithms were applied to two trajectory problems. First, a pure roll rotation was studied. This motion is particularly relevant since it had proven to be significantly problematic in past RACOON-Lab studies. The other considered trajectory was the tumbling of an actual orbiting satellite: MOVE-II. This spacecraft's attitude history was reconstructed after processing the data collected by its gyroscopes over a period of time when it orbited unactuated.

The conventional approaches provided a limited success in the execution of the roll rotation. In this case, in order to achieve feasible motion it was necessary to resort to high gain coefficients. These high gain coefficients proved to be effective, but they simultaneously seemed to be detrimental for the efficiency and triggered oscillations. Furthermore, the execution of MOVE-II's trajectory could not be completely achieved with any of these methods regardless of the gain.

The approach based on the analytical solution outperformed the conventional methodologies in both cases. For the roll rotation, an efficiency of 50% was obtained, compared to the rough 10% achieved by ASC and WLN. Regarding the simulation of MOVE-II's trajectory, the algorithm based on the analytical solution proved able to execute it with an efficiency of 70%.

Complementary to the research in the trajectory problem, the developed inverse kinematics solution was applied for studying the facility's capabilities. In particular, the robot's efficiency was studied as a function of the geometrical constraints. This enabled an objective assessment of the robot's kinematical capabilities.

Table of Contents

1	MOTIVATION	1
2	STATE OF THE ART	2
2.1	Solution in the Velocity Domain	3
2.1.1	Resolved Motion Rate Control and Automatic Supervisory Control	3
2.1.2	Automatic Supervisory Control	4
2.1.3	Weighted Least Norm Method	5
2.1.4	Further Upgrades	5
2.2	Topological Approach	6
3	GOAL OF THIS THESIS	8
4	APPROACH	9
5	ENVIRONMENT DESCRIPTION	10
5.1	System Architecture	10
5.2	System Kinematics	12
5.2.1	Basic Definitions	12
5.2.2	Chaser Kinematics	16
5.2.3	Rotator Kinematics	17
6	SOLUTION OF THE INVERSE KINEMATICS	22
6.1	Analytical Solution	22
6.1.1	Problem Statement	22
6.1.2	Strategy Overview	22
6.1.3	Individual Solutions	24
6.1.4	Complete Solution	24
6.1.5	Solution Representation	31
6.1.6	Algorithm Testing	34
6.2	Solution in the Velocity Domain	35
6.2.1	Problem Statement	36
6.2.2	Strategy Overview	36
6.2.3	Problem Solution	36
6.2.4	Singularity Handling	38
6.2.5	Solution Example	38
6.2.6	Algorithm Testing	39
7	TRAJECTORY PLANNING	41
7.1	Definitions	41
7.1.1	Trajectory Problem Statement	41
7.1.2	Trajectory Efficiency	42
7.1.3	Attitude Error	44
7.1.4	Static and Dynamic Trajectories	44

7.1.5	Rotation Example	44
7.2	Conventional Algorithms	45
7.2.1	Automatic Supervisory Control	45
7.2.2	Weighted Least Norm	50
7.3	Algorithms based on the Analytical Solution	53
7.3.1	Minimum Instantaneous Control Effort	54
7.3.2	Isokinematic Centroid Tracking	56
7.3.3	Relaxed Isokinematic Centroid Tracking	61
7.3.4	Automatic Supervisory Control with Distance to an Isokinematic Centroid as Performance Criterion	63
7.4	Summary	64
8	SYSTEM EVALUATION	65
8.1	System Capabilities	65
8.2	Singularity Analysis	67
8.2.1	Singularity Definition	67
8.2.2	Singularity Identification	68
8.2.3	Examples of Singularities Across a Trajectory	71
9	PRACTICAL CASE	73
9.1	MOVE-II Description	73
9.2	Attitude Data	74
9.3	Attitude Simulation within the RACOON-Lab	76
9.3.1	Automatic Supervisory Control	76
9.3.2	Relaxed Isokinematic Centroid Tracking	77
10	DISCUSSION	80
11	CONCLUSIONS	81
12	OUTLOOK	83

List of Figures

FIG. 2-1: EXAMPLE OF SOLUTIONS FOR THE INVERSE KINEMATICS FOR DIFFERENT DESIRED COORDINATES. IMAGE COURTESY OF [4]	2
FIG. 2-2. SLAVE MA-23 MANIPULATOR USED IN LIÉGOIS STUDY. FIGURE FROM [8].....	4
FIG. 2-3. COMPARISON OF THE TRAJECTORY SOLUTIONS FOR THE RMRC (LABELLED AS LN), ACS (LABELLED AS GPM) AND WLN METHODS [10].....	5
FIG. 2-4. EXAMPLE OF OF AN ISOKINEMATIC TOPOLOGY FOR A CONSTRAINED ROBOT AND THE CONECTIVITY BUNDLE OF ITS ELEMENTS [16].	6
FIG. 5-1. ARTIST IMPRESSION OF ATV DOCKING WITH THE ISS [19].....	10
FIG. 5-2. DEUTSCHE ORBITALE SERVICING MISSION (DEOS) [20]	10
FIG. 5-3: PHOTOGRAPHY OF THE RACOON-LAB WITH ALL ITS DEGREES OF FREEDOM [2]	11
FIG. 5-4. SCHEME OF RACOON-LAB'S ELEMENTS [22] AND ILLUMINATION INSTRUMENTATION [18].....	11
FIG. 5-5: REFERENCE SYSTEM IN RACOON-LAB WITH CHASER AND ROTATOR IN ITS DEFAULT POSITIONS	14
FIG. 5-6: CHAIN MULTIBODY SYSTEM EXAMPLE [25].....	15
FIG: 5-7: EXAMPLES OF REVOLUTE AND PRISMATIC JOINTS [26]	15
FIG. 5-8: BOUNDARIES OF THE ELEVATION ANGLE	17
FIG. 5-9: LIMITS OF THE NICK ANGLES	18
FIG. 5-10: POSSIBLE ROTATION SEQUENCE AROUND THE X AXIS FOR RACOON-LAB'S TARGET	18
FIG. 5-11. EXAMPLE OF A LOCKING CONFIGURATION FOR THE RACOON-LAB'S ROTATOR	19
FIG. 5-12. PROBABILITY DENSITY FUNCTION OF THE ROBOTS COORDINATES AS A FUNCTION OF THE ROLL AND PITCH ANGLES.....	21
FIG. 6-1: $K1^2 + K2^2$ FOR THE DETERMINATION OF THE ELEVATION ANGLE.....	26
FIG. 6-2: MINIMUM ANGLE BETWEEN THE END EFFECTOR'S VERTICAL AXES AND THE HORIZONTAL PLANE FOR MAKING A COMBINATION OF γ AND δ FEASIBLE.	26
FIG. 6-3: MANIFOLD OF THE CONSTRAINED JOINTS FOR DIFFERENT VALUES OF X.....	28
FIG. 6-4. DIAGRAM OF POSSIBILITIES FOR CALCULATING THE AZIMUTH AND POLAR JOINTS	29
FIG. 6-5: INVERSE KINEMATICS SOLUTION EXAMPLE (I)	32
FIG. 6-6: INVERSE KINEMATICS SOLUTION EXAMPLE (II)	33
FIG. 6-7: PROBABILITY DENSITY FUNCTION OF THE TIME PER SOLUTION IN THE RANDOM TEST FOR THE ANALYTICAL SOLUTION OF THE INVERSE KINEMATICS.....	34
FIG. 6-8: PROBABILITY DENSITY FUNCTION OF THE TIME PER SOLUTION IN THE SINGULARITY TEST FOR THE ANALYTICAL SOLUTION OF THE INVERSE KINEMATICS.....	35
FIG. 6-9. TIME PER SOLUTION OF THE ALGORITHM AS A FUNCTION OF THE ANGLE BETWEEN THE END EFFECTOR'S VERTICAL AXIS AND THE HORIZONTAL PLANE.	35
FIG. 6-10: SOLUTION EXAMPLE OF THE INVERSE KINEMATICS IN THE VELOCITY DOMAIN.....	39
FIG. 6-11: PROBABILITY DENSITY FUNCTION OF THE TIME PER SOLUTION IN THE RANDOM TEST FOR THE SOLUTION OF THE INVERSE KINEMATICS IN THE VELOCITY DOMAIN	40
FIG. 6-12: PROBABILITY DENSITY FUNCTION OF THE TIME PER SOLUTION IN THE RANDOM TEST FOR THE SOLUTION OF THE INVERSE KINEMATICS IN THE VELOCITY DOMAIN	40
FIG. 7-1: EXAMPLE TRAJECTORY USED FOR THE ALGORITHM TESTING.....	45
FIG. 7-2: RESULTING TRAJECTORY FOR AUTOMATIC SUPERVISORY CONTROL ($\alpha k = 24$ S/RAD).....	47
FIG. 7-3: OSCILLATIONS IN THE B-NICK JOINT WITH AUTOMATIC SUPERVISORY CONTROL	47
FIG. 7-4: OSCILLATIONS IN THE ELEVATION JOINT WITH AUTOMATIC SUPERVISORY CONTROL	47
FIG. 7-5: SENSITIVITY COEFFICIENTS OF THE ROLL MOTION WITH RESPECT TO THE CONFIGURATION PARAMETERS	48
FIG. 7-6: RESULTING EFFICIENCY FOR AUTOMATIC SUPERVISORY CONTROL ($\alpha k = 24$).....	48
FIG. 7-7: FILTERING OF THE ELEVATION ANGLE BY FOURIER ANALYSIS.....	49
FIG. 7-8: ERROR INTRODUCED BY THE FILTERING THE ASC TRAJECTORY FOR DIFFERENT CUT-OFF FREQUENCIES.....	49
FIG. 7-9: EFFICIENCY OF THE FILTERED ASC TRAJECTORY FOR DIFFERENT CUT-OFF FREQUENCIES.....	50
FIG. 7-10: RESULTING TRAJECTORY FOR WEIGHTED LEAST NORM ($\alpha k = 25$ S ² /RAD ²)	51
FIG. 7-11: RESULTING TRAJECTORY FOR WEIGHTED LEAST NORM ($\alpha k = 50$ S ² /RAD ²)	51

FIG. 7-12: RESULTING EFFICIENCY FOR WEIGHTED LEAST NORM ($A_K=25 \text{ S}^2/\text{RAD}^2$)	52
FIG. 7-13: RESULTING EFFICIENCY FOR WEIGHTED LEAST NORM ($A_K=50 \text{ S}^2/\text{RAD}^2$)	52
FIG. 7-14: FILTERING OF THE ELEVATION ANGLE BY FOURIER ANALYSIS.....	52
FIG. 7-15: ERROR INTRODUCED BY THE FILTERING THE WLN TRAJECTORY FOR DIFFERENT CUT-OFF FREQUENCIES.....	53
FIG. 7-16: EFFICIENCY OF THE FILTERED ASC TRAJECTORY FOR DIFFERENT CUT-OFF FREQUENCIES.....	53
FIG. 7-17: TRAJECTORY OBTAINED BY THE MINIMUM INSTANTANEOUS CONTROL EFFORT ALGORITHM	54
FIG. 7-18: TRAJECTORY EFFICIENCY OBTAINED BY THE MINIMUM INSTANTANEOUS CONTROL EFFORT ALGORITHM	54
FIG. 7-19: INVERSE KINEMATICS MANIFOLD FOR TRAJECTORY'S POINTS AROUND THE LOCKING INSTANT	55
FIG. 7-20: PROBABILITY DENSITY FUNCTION OF THE AZIMUTH JOINT THROUGH THE EXAMPLE TRAJECTORY	56
FIG. 7-21: PROBABILITY DENSITY FUNCTION OF THE ELEVATION JOINT THROUGH THE EXAMPLE TRAJECTORY	57
FIG. 7-22: PROBABILITY DENSITY FUNCTION OF THE A-NICK JOINT THROUGH THE EXAMPLE TRAJECTORY	57
FIG. 7-23: PROBABILITY DENSITY FUNCTION OF THE B-NICK JOINT THROUGH THE EXAMPLE TRAJECTORY	57
FIG. 7-24: PROBABILITY DENSITY FUNCTION OF THE POLAR JOINT THROUGH THE EXAMPLE TRAJECTORY	58
FIG. 7-25: PROBABILITY DENSITY FUNCTION FOR THE AZIMUTH FOR DIFFERENT COORDINATES	59
FIG. 7-26: MOST PROBABLE VALUES FOR THE DIFFERENT JOINTS OVER THE EXAMPLE TRAJECTORY	60
FIG. 7-27: TRAJECTORY WITH THE ISOKINEMATIC CENTROID TRACKING	60
FIG. 7-28: EFFICIENCY WITH THE ISOKINEMATIC CENTROID TRACKING	61
FIG. 7-29: TRAJECTORY OBTAINED WITH THE RELAXED ISOKINEMATIC CENTROID TRACKING WITH A MINIMUM EFFICIENCY OF A 50%.....	62
FIG. 7-30: EFFICIENCY OBTAINED WITH THE RELAXED ISOKINEMATIC CENTROID TRACKING WITH A MINIMUM EFFICIENCY OF A 50%.....	62
FIG. 7-31: ASC WITH DISTANCE TO THE MINIMUM DISTANCE TO AN ISOKINEMATIC CENTROID AS THE PERFORMANCE CRITERIUM	63
FIG. 7-32: EFFICIENCY OBTAINED WITH ASC USING THE PDF AS THE PERFORMANCE CRITERIUM	63
FIG. 8-1: LINEARIZATION OF INVERSE KINEMATICS CENTROID MOTION AROUND $\varphi=180^\circ$	65
FIG. 8-2: INVERSE KINEMATICS CENTROID GRADIENT IN A ROLL ROTATION FOR THE RACOON-LAB	66
FIG. 8-3: EXAMPLES OF THE ARGUMENT OF THE MINIMUM DETERMINANT FOR THE REDUCED JACOBIAN	68
FIG. 8-4: AZIMUTH-POLAR COUPLING SINGULARITY	69
FIG. 8-5: SINGULARITY MAP WITH THE POLAR-AZIMUTH COUPLING VISIBLE IN THE UPPER AREA	69
FIG. 8-6: AZIMUTH-A-NICK COUPLING SINGULARITY	70
FIG. 8-7: ELEVATION-B-NICK COUPLING SINGULARITY	70
FIG. 8-8: ABSOLUTE VALUES OF THE REDUCED JACOBIANS ACROSS A TRAJECTORY PERFORMED WITH ASC	71
FIG. 8-9: LOCKING OF ASC FOR PERFORMING A PITCH ROTATION	72
FIG. 8-10: RMPV PERFORMING A PITCH ROTATION FROM THE DEFAULT CONFIGURATION	72
FIG. 9-1: COMPUTER GENERATED GRAPHIC (CGI) OF MOVE-II WITH DEPLOYED FLAPPANELS [30].....	73
FIG. 9-2: BMX055 ADCS SENSOR	74
FIG. 9-3: VISUALIZATION OF MOVE-II ORBITING ON MAY 13TH AT 10:04 GMT	74
FIG. 9-4: SPECTRAL RELATIVE ENERGY OF THE HARMONICS OBTAINED BY THE REGRESSION.....	75
FIG. 9-5: RECONSTRUCTION OF MOVE-II GYROSCOPIC DATA.....	75
FIG. 9-6: MOVE II-A EULER ANGLES RATES OVER THE STUDY TIME.....	76
FIG. 9-7: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH ASC.....	77
FIG. 9-8: VIOLATION OF LIMITS IN ELEVATION AND B-NICK WITH ASC AND $A_K=60 \text{ S}/\text{RAD}$	77
FIG. 9-9: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $\eta_{\text{MIN}}=180\%$	77

FIG. 9-10: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $\eta_{\text{MIN}}=70\%$ (I)	78
FIG. 9-11: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $\eta_{\text{MIN}}=70\%$ (II)	78

Appendices

FIG-C 1: ASC FOR A ROLL ROTATION WITH $A_K= 2 \text{ S/RAD}$	90
FIG-C 2: ASC FOR A ROLL ROTATION WITH $A_K= 4 \text{ S/RAD}$	90
FIG-C 3: ASC FOR A ROLL ROTATION WITH $A_K= 12 \text{ S/RAD}$	90
FIG-C 4: ASC FOR A ROLL ROTATION WITH $A_K= 16 \text{ S/RAD}$	91
FIG-C 5: ASC FOR A ROLL ROTATION WITH $A_K= 24 \text{ S/RAD}$	91
FIG-C 6: ASC FOR A ROLL ROTATION WITH $A_K= 32 \text{ S/RAD}$	91
FIG-C 7: ASC FOR A ROLL ROTATION WITH $A_K= 40 \text{ S/RAD}$	92
FIG-C 8: WLN FOR A ROLL ROTATION WITH $A_K= 5 \text{ S}^2/\text{RAD}^2$	92
FIG-C 9: WLN FOR A ROLL ROTATION WITH $A_K= 10 \text{ S}^2/\text{RAD}^2$	92
FIG-C 10: WLN FOR A ROLL ROTATION WITH $A_K= 15 \text{ S}^2/\text{RAD}^2$	93
FIG-C 11: WLN FOR A ROLL ROTATION WITH $A_K= 25 \text{ S}^2/\text{RAD}^2$	93
FIG-C 12: WLN FOR A ROLL ROTATION WITH $A_K= 35 \text{ S}^2/\text{RAD}^2$	93
FIG-C 13: WLN FOR A ROLL ROTATION WITH $A_K= 50 \text{ S}^2/\text{RAD}^2$	94
FIG-C 14: WLN FOR A ROLL ROTATION WITH $A_K= 100 \text{ S}^2/\text{RAD}^2$	94
FIG-C 15: WLN FOR A ROLL ROTATION WITH $A_K= 200 \text{ S}^2/\text{RAD}^2$	94
FIG-C 16: WLN FOR A ROLL ROTATION WITH $A_K= 300 \text{ S}^2/\text{RAD}^2$	95
FIG-C 17: WLN FOR A ROLL ROTATION WITH $A_K= 2 \text{ S/RAD}$	95
FIG-D 1: SOLUTIONS FROM $\Phi=15^\circ$ TO $\Phi=45^\circ$	96
FIG-D 2: SOLUTIONS FROM $\Phi=60^\circ$ TO $\Phi=105^\circ$	97
FIG-D 3: SOLUTIONS FROM $\Phi=120^\circ$ TO $\Phi=175^\circ$	98
FIG-D 4: SOLUTIONS FROM $\Phi=180^\circ$ TO $\Phi=225^\circ$	99
FIG-D 5: SOLUTIONS FROM $\Phi=240^\circ$ TO $\Phi=285^\circ$	100
FIG-D 6: SOLUTIONS FROM $\Phi=300^\circ$ TO $\Phi=345^\circ$	101
FIG-D 7: SOLUTION FOR $\Phi=360^\circ$	102
FIG-E 1: RELAXED ISOKINEMATIC CENTROID TRACKING TRAJECTORY WITH $H_{\text{MIN}}=20\%$	103
FIG-E 2: EFFICIENCY OF RELAXED ISOKINEMATIC CENTROID TRACKING ALGORITHM WITH $H_{\text{MIN}}=20\%$	103
FIG-E 3: RELAXED ISOKINEMATIC CENTROID TRACKING TRAJECTORY WITH $H_{\text{MIN}}=25\%$	103
FIG-E 4: EFFICIENCY OF RELAXED ISOKINEMATIC CENTROID TRACKING ALGORITHM WITH $H_{\text{MIN}}=25\%$	104
FIG-E 5: RELAXED ISOKINEMATIC CENTROID TRACKING TRAJECTORY WITH $H_{\text{MIN}}=30\%$	104
FIG-E 6: EFFICIENCY OF ISOKINEMATIC CENTROID TRACKING TRACKING ALGORITHM WITH $H_{\text{MIN}}=30\%$	104
FIG-E 7: RELAXED ISOKINEMATIC CENTROID TRACKING TRAJECTORY WITH $H_{\text{MIN}}=35\%$	104
FIG-E 8: EFFICIENCY OF RELAXED ISOKINEMATIC CENTROID TRACKING ALGORITHM WITH $H_{\text{MIN}}=35\%$	105
FIG-E 9: RELAXED ISOKINEMATIC CENTROID TRACKING TRAJECTORY WITH $H_{\text{MIN}}=40\%$	105
FIG-E 10: EFFICIENCY OF RELAXED ISOKINEMATIC CENTROID TRACKING ALGORITHM WITH $H_{\text{MIN}}=40\%$	105
FIG-E 11: RELAXED ISOKINEMATIC CENTROID TRACKING TRAJECTORY WITH $H_{\text{MIN}}=45\%$	105
FIG-E 12: EFFICIENCY OF RELAXED ISOKINEMATIC CENTROID TRACKING ALGORITHM WITH $H_{\text{MIN}}=45\%$	106
FIG-E 13: RELAXED ISOKINEMATIC CENTROID TRACKING TRAJECTORY WITH $H_{\text{MIN}}=50\%$	106
FIG-E 14: EFFICIENCY OF RELAXED ISOKINEMATIC CENTROID TRACKING ALGORITHM WITH $H_{\text{MIN}}=50\%$	106

FIG-F 1: ROLL ANGLE OF MOVE-II OVER THE STUDY TIME.....	107
FIG-F 2: ROLL RATE OF MOVE-II OVER THE STUDY TIME	107
FIG-F 3: PITCH ANGLE OF MOVE-II OVER THE STUDY TIME	107
FIG-F 4: PITCH RATE OF MOVE-II OVER THE STUDY TIME.....	108
FIG-F 5: YAW ANGLE OF MOVE-II OVER THE STUDY TIME	108
FIG-F 6: YAW RATE OF MOVE-II OVER THE STUDY TIME.....	108
FIG-F 7: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH ASC AND $\dot{A}_K=60$ S/RAD (I)	109
FIG-F 8: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH ASC AND $\dot{A}_K=60$ S/RAD (II)	109
FIG-F 9: ANGULAR VELOCITIES OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH ASC AND $\dot{A}_K=60$ S/RAD.....	109
FIG-F 10: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=180\%$	110
FIG-F 11: ANGULAR VELOCITIES OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=180\%$	110
FIG-F 12: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=150\%$	110
FIG-F 13: ANGULAR VELOCITIES OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=150\%$	111
FIG-F 14: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=27\%$ (I).....	111
FIG-F 15: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=27\%$ (II).....	111
FIG-F 16: ANGULAR VELOCITIES OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=27\%$ (I).....	112
FIG-F 17: ANGULAR VELOCITIES OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=27\%$ (II).....	112
FIG-F 18: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=50\%$ (I).....	112
FIG-F 19: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=50\%$ (II).....	113
FIG-F 20: ANGULAR VELOCITIES OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=50\%$ (I).....	113
FIG-F 21: ANGULAR VELOCITIES OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=50\%$ (II).....	113
FIG-F 22: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=70\%$ (I).....	114
FIG-F 23: CONFIGURATION OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=70\%$ (I).....	114
FIG-F 24: ANGULAR VELOCITIES OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=70\%$ (I).....	114
FIG-F 25: : ANGULAR VELOCITIES OF THE RACOON-LAB'S JOINTS FOR SIMULATING MOVE-II ATTITUDE HISTORY WITH RIKCT AND $H_{MIN}=70\%$ (II).....	115



List of Tables

TABLE 5-1: SUMMARY OF JOINTS OF THE ROTATOR	19
TABLE 6-1: SUMMARY OF POSSIBILITIES FOR CALCULATING THE AZIMUTH AND POLAR JOINTS.....	29
TABLE 7-1: SUMMARY OF PERFORMANCE FOR THE INVERSE KINEMATIC ALGORITHMS TESTED FOR A ROLL ROTATION	64
TABLE 8-1: LIST OF REDUCED JACOBIANS FOR THE RACOON-LAB.....	67

Symbols

Symbol	Unit	Euclidean Space	Description
c_0	[-]	\mathbb{R}	Linearized kinematic centroid for no nick and elevation range
c_n	rad ⁻¹	\mathbb{R}	Linearized sensitivity of kinematic centroid with respect to nick range
c_β	rad ⁻¹	\mathbb{R}	Linearized sensitivity of kinematic centroid with respect to elevation range
D_{AB}	[-]	$\mathbb{R}^{4 \times 4}$	Homogeneous transformation of space coordinates from frame B to frame A
f	several	several	Generic function
H	s or s ² /rad or s ² /m	\mathbb{R}	Performance Criterion
i	[-]	\mathbb{N}	Row index
I_n	[-]	$\mathbb{R}^{n \times n}$	n-dimensional identity matrix
j	[-]	\mathbb{N}	Column index
j	[-]	\mathbb{C}	Imaginary unit
K_1	[-]	\mathbb{R}	Sinusoidal component of a matrix entry with respect to ω
K_2	[-]	\mathbb{R}	Cosinusoidal component of a matrix entry with respect to ω
J_{AB}	Unit(A)/Unit(B)	$\mathbb{R}^{n_{DoF,A} \times m_{DoF,B}}$	Jacobian of A's derivative with respect to B
$J_{xy,ab}^\circ$	[-]	$\mathbb{R}^{3 \times 3}$	Reduced Jacobian after removing the rows a and b
m	[-]	\mathbb{R}	Value of a given entry from a rotation matrix
M_{AB}	[-]	$\mathbb{R}^{3 \times 3}$	Rotation matrix for containing coordinates in frame B to coordinates in frame A
M_d	[-]	$\mathbb{R}^{3 \times 3}$	Desired attitude matrix

p_E	rad/s	\mathbb{R}	Rotation component in x's axis in body frame
q	rad and/or m	\mathbb{R}^{DoF}	Vector of Robot's Configuration
q_E	rad/s	\mathbb{R}	Rotation component in y's axis in body frame
q_{μ_L}	rad	\mathbb{R}^{n_L}	Vector of the isokinematic Centroids for the Limited Joints
q_{μ_C}	rad	$\mathbb{R}^{2 \times n_C}$	Vector of the isokinematic Centroids for the Cyclic Joints
r_E	rad/s	\mathbb{R}	Rotation component in z's axis in body frame
S	rad/s	$\mathbb{R}^{3 \times 3}$	Rotation Tensor
Q	rad and/or m	\mathbb{R}^{DoF}	Infinite set of physically feasible configurations
R_{AB}	m	\mathbb{R}^3	Offset between Frame B and Frame A
r_0	rad	\mathbb{R}	Lower bound of the solution region that contains the highest concentration of configurations
r_1	rad	\mathbb{R}	Upper bound of the solution region that contains the highest concentration of configurations
t	s	\mathbb{R}	Time
w_d	rad and m	\mathbb{R}^6	Vector of desired coordinates
W	rad ² / s ² and/or m ² / s ²	$\mathbb{R}^{DoF \times DoF}$	Weighting matrix
W_0	rad ² / s ² and/or m ² / s ²	$\mathbb{R}^{DoF \times DoF}$	Reference weighting matrix
x	m	\mathbb{R}	Coordinate in the x axis
y	m	\mathbb{R}	Coordinate in the y axis
z	m	\mathbb{R}	Coordinate in the z axis
Z_F	rad and m	\mathbb{R}^6	Space and angular coordinates in frame F

α	rad	\mathbb{R}	Azimuth angle
α_k	s/ rad or s/ m or s ² / rad ² or s ² / m ²	\mathbb{R}	Coefficient of the performance criterion
β	rad	\mathbb{R}	Elevation angle
γ	rad	\mathbb{R}	A-Nick angle
Γ	[-]	$\mathbb{R}^{3 \times 3}$	Argument of the reduced determinant closest to zero
δ	rad	\mathbb{R}	B-Nick angle
ε	rad	\mathbb{R}	Polar angle
$\epsilon_{\mathcal{E}}$	rad	\mathbb{R}	Attitude error
ζ	rad ²	\mathbb{R}	Minimum distance to a an isokinematic manifold's centroid
η	[-]	\mathbb{R}	Trajectory Efficiency
η_{min}	[-]	\mathbb{R}	Minimum Allowable efficiency
η_M	[-]	\mathbb{R}	Maneuver efficiency
θ	rad	\mathbb{R}	Pitch angle
κ	rad	\mathbb{R}	Control effort
μ	rad	\mathbb{R}	Euler angles' norm
ρ	rad	\mathbb{R}	Rotation angle
σ	rad	\mathbb{R}^{DoF}	Isokinematic manifold
φ	rad	\mathbb{R}	Roll angle
Φ	[-]	\mathbb{R}	Cost function
χ	rad	\mathbb{R}	Angle between the horizontal plane and the end-effectors vertical axis
ψ	rad	\mathbb{R}	Yaw angle
ω	rad	\mathbb{R}	Angle of a revolute join
Ω	rad	\mathbb{R}	Maximum
Ω_E	rad	\mathbb{R}^3	Vector of



Operators

Symbol	Unit	Euclidean Space	Description
C_{xy}	Unit(x)/Unit(y)	\mathbb{R}	$\frac{\partial x}{\partial y}$
\dot{x}	Unit(x)/s	\mathbb{R}	$\frac{\partial x}{\partial t}$

Abbreviations

ACDS	Attitude Determination and Control System	NMPC	Nonlinear Model Predictive Control
ASC	Automatic Supervisory Control	PDF	Probability Distribution Function
ATV	Automated Transfer Vehicle	RACOON-Lab	Robotic Actuation and On-Orbit Navigation Laboratory
BFS	Breath First Search		
CDR	Critical Design Review	RIKCT	Relaxed Isokinematic Centroid Tracking
COF	Cut-off Frequency	RMRC	Resolved Motion Rate Control
CIG	Computer Generated Graphic	TPS	Time per Solution
DoF	Degree of Freedom	WLN	Weighted Least Norm
DEOS	Deutsche Orbitale Servicing Mission	WARR	Wissenschaftliche Arbeitsgemeinschaft für Raketentechnik und Raumfahrt
DK	Inverse Kinematics		
ESA	European Space Agency		
IKCT	Isokinematic Centroid Tracking		
ISS	International Space Station		
LEO	Low Earth Orbit		
MEO	Medium Earth Orbits		
MICE	Minimum Instantaneous Control Effort		
MOVE	Munich Orbital Verification Experiment		

1 Motivation

The proliferation of defunct spacecraft and space debris, especially in Low Earth Orbit (LEO), poses an unprecedented challenge for the future of space operations. In order to overcome this problem, it is necessary to explore novel technologies able to cope with this issue. Among the different proposed alternatives, many require the mastering of proximity operations involving un-cooperative targets [1]. Facilities such as RACOON-Lab provide a realistic environment for extensive testing and simulation of on-orbit operations.

To maximize the test's validity, it is necessary to establish a framework as diaphanous as possible. A clear line of sight between the involved robots is essential for testing algorithms based on visual recognition techniques. These characteristics are achieved by limiting the number of actuators and imposing strong geometrical constraints on them. However, such designs cause convoluted kinematics with a great number of singularities.

Such is the case for the RACOON-Lab. The utilization of this facility has been limited through its first years of operation due to its challenging inverse kinematics. Several past studies have tried to solve this problem but no robust and generalized solution has yet been found [2].

In order to maximize the capabilities of this research tool, it is necessary to develop algorithms capable of solving the inverse kinematics of the involved robots. Removing this hindrance would enable the use of the RACOON-Lab for the recreation of complex orbital trajectories and provide a functional framework for testing on-orbit proximity operations.

2 State of the Art

The Rotator is the most challenging element within the RACOON-Lab from a kinematic perspective and it is the device on which the majority of this Thesis will focus. This mechanism can be described as a robot, since it is composed of several individual links that are interconnected through joints.

The state of a robot is characterized through its configuration vector q . In this array each element represents the position of one of its joints. The robot's configuration determines spatial and orientation coordinates for each joint. These coordinates are summarized in the coordinates vector w , in which each element is associated with one of the robot's degrees of freedom in space.

The most relevant of a robot's elements is its end effector. This link contains the robot's termination point and it is responsible for interacting with the environment. Therefore, the robot's behavior is usually defined by the end effector's trajectory. The inverse kinematics problem consists of finding feasible configuration(s) vector(s) q , which lead to the desired coordinates of the end effector w_d [3, 4]:

$$q = f(w_d) \forall q \in Q \quad \text{Eq. (2-1)}$$

Where Q denotes the set of configurations which comply with the robot's physical limitations and are therefore feasible. This problem is only well posed for the extremely limited number of cases in which a unique configuration provides the desired coordinates. In the most common scenario, the problem is ill-posed either because there are several or infinite solutions or because there is no solution at all [3, 4]. The variety of possibilities of the inverse kinematics is illustrated with an example in Fig. 2-1. In this figure $nsol$ denotes the number of existing solutions which are available for the same robot depending on the desired coordinates.

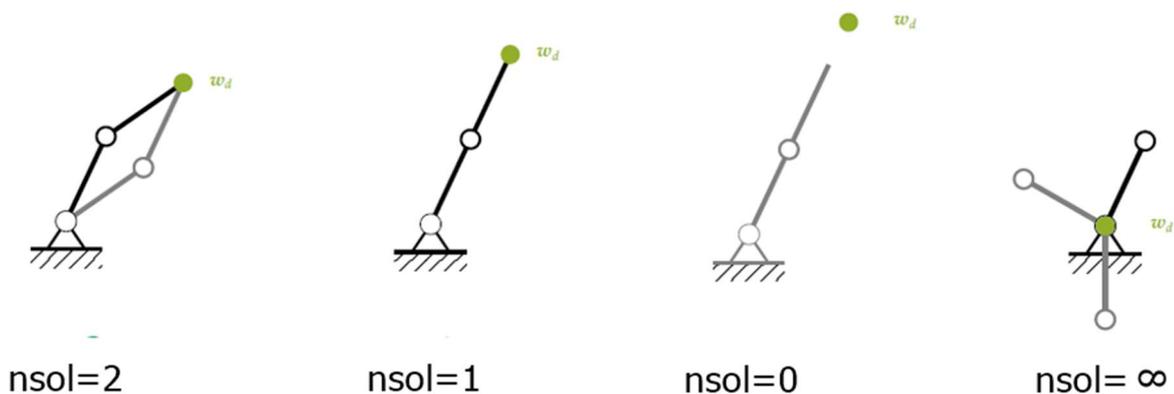


Fig. 2-1: Example of Solutions for the Inverse Kinematics for Different Desired Coordinates. Image Courtesy of [4]

Most of the scientific literature has studied the inverse kinematics with a focus on its implementation for robotic arms, characterized by a high degree of redundancy. The common points between robotic arms and the RACOON-Lab's Rotator is their redundancy (albeit in different degrees) and the geometrical constraints. The majority of the existing approaches originate from robotics, so this field will be the one considered for the identification of the top performing methodologies. Nonetheless, the RACOON-Lab's rotator presents some remarkable differences from the conventional

robotic arms that should be noted. This multibody presents a freedom of motion restricted to the attitude domain, with extraordinarily constrained joints and a low degree of redundancy. Furthermore, its exotic kinematic chain is characterized by a significant number of singularities and a solution space that is not convex at all points.

In the last decades, several algorithms have been proposed for addressing the inverse kinematics of redundant constrained robots. The wide diversity of these methodologies enables a myriad of possible ways for sorting them. In the context of the present project, the following classification will be used:

- Solution in the velocity domain. In this group of methods, the problem is approached using the time derivatives of the coordinates and the result is integrated afterwards. Since the kinematic equations in the velocity domain constitute linear systems, its numerical solution is easier to obtain.
- Analytical approach. In this case the kinematic equations are solved for each point of the trajectory. This implies the resolution of the non-linear geometrical equations. Such procedures are much more demanding from the point of view of mathematical complexity.

2.1 Solution in the Velocity Domain

The solution in the velocity domain requires solving the following undetermined linear system of equations for each point of the trajectory:

$$\dot{w}_d = J_{\dot{w}q} \dot{q} \quad \text{Eq. (2-2)}$$

Where \dot{w}_d denotes the desired time derivate of the the robot's end effector's coordinates, \dot{q} is the time derivative of the robot's configuration and $J_{\dot{w}q}$ is the Jacobian tensor which relates both vectors. In this system of equations, the number of columns of $J_{\dot{w}q}$ is greater to its number of rows due to the robot's redundancy. In order to handle this lack of conditions, additional tasks must be defined. These tasks are formulated as cost functions, which intend to minimize the possibility of reaching a terminal position while executing smooth trajectories. Therefore, the main distinctive feature of each algorithm is the philosophy used for the determination of the cost function.

2.1.1 Resolved Motion Rate Control and Automatic Supervisory Control

One of the first algorithms proposed for solving the inverse kinematics of redundant systems was the Resolved Motion Rate Control (RMRC). This method was introduced by Whitney in 1969 [5] and establishes an optimization problem with a cost function while simultaneously accounting for the compliance of the kinematic equations of the robot. The cost function is defined as:

$$\Phi = \frac{1}{2} \dot{q}^T W \dot{q} \quad \text{Eq. (2-3)}$$

With W being a weighting matrix and \dot{q} the time derivative of the robot's configuration. The goal is to minimize the value of Φ while simultaneously complying with the kinematic condition i.e. Eq. (2-1). This problem can be adressed efficiently using the Lagrange multipliers method [4, 6]. The RMRC enables a linear formulation of the problem, which can be highly advantageous for introducing additional dependencies

such as momentum conservation [7]. Nevertheless, this method presents the disadvantage of not being able to incorporate geometrical restrictions of the joints. Its application is thus limited to unconstrained robots.

2.1.2 Automatic Supervisory Control

The Automatic Supervisory Control (ASC) is an extension of the RMRC proposed by Liégeois in 1977 [8]. This approach includes the dependency on the current configuration in the cost function using a smooth function $H(q)$ in the following fashion:

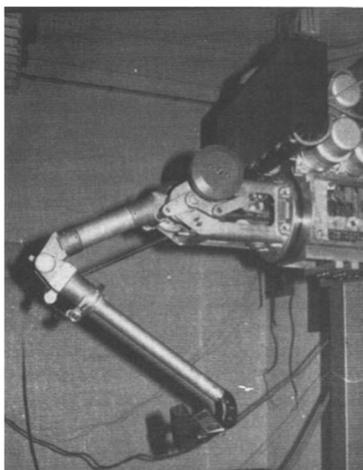
$$\Phi = \frac{1}{2} \dot{q}^T W \dot{q} + \alpha_k \frac{\partial H(q)}{\partial q} \dot{q} \quad \text{Eq. (2-4)}$$

With $\alpha > 0$ being an auxiliary parameter which can be tuned in order to variate the global authority of the current configuration in the cost. If one keeps the boundary kinematic condition of Eq. (2-1) as in RMRC it is possible to solve the problem using the Lagrange Multipliers method.

This procedure enables strategies able to address the joints' limitations problematic. For example, the function H can be shaped in a way that tends to infinite when the joint's configuration tends to its limit values. In particular, Liégeois proposed the use of quadratic hyperbolae with its singularities placed in the joint's limits. The robot that Liégeois used to test his method and its constrains are displayed in Fig. 2-2.

This approach provides a good starting point for incorporating the joints' limits into the control system but it has several shortcomings:

- Only the homogeneous space is used for the task of optimization. The outcome of this limitation can be a low efficiency of the trajectory.
- The value of the constant α is not trivial and must be chosen for each robot and for each application. This is often done by trial and error, choosing a very high value and gradually decreasing it until the physical limitations are violated.
- If the value of α_k is too high, the trajectory tends to be attached to the intermediate position between the joints' limits, which results in highly unefficient trajectories.
- High values of the constant α_k can produce important oscillations [9].



Joint	q min [°]	q max [°]	Limitation [%]
1	-60	60	67
2	-70	57	65
3	-135	-30	71
4	-179	179	1
5	-29	109	62
6	-180	180	0

Fig. 2-2. Slave MA-23 Manipulator used in Liégeois Study. Figure from [8].

2.1.3 Weighted Least Norm Method

The Weighted Least Norm (WLN) method is modification of the ASC Method proposed by Chan et.al in 1995 [10] in order to overcome some of its disadvantages.

In this method, the optimization function Φ has the same form as in the Resolved Motion Rate Control in Eq. (2-3). This is achieved by incorporating the function H into the weighting Matrix W so that it depends on the robot's configuration:

$$W = W(q) = W_0 + \frac{\partial H(q)}{\partial q} \tag{ 2-5 }$$

This approach uses the space in a more efficient way and can avoid the problem with oscillations. In Chan's study, a trajectory was simulated with the RMRC, ASC and the WLN Methods. The results are displayed in Fig. 2-3. As it can be seen, the RMRC ends up violating the joint's limitation. Both the WLN and ASC methods are able to remain within the joint's limit but the WLN algorithm results in a more efficient utilization of the space. This feature is evident if one observes the smoother trajectory of the WLN solution, which is able to fulfill the task without resorting to abrupt deviations as its joints approach to their limits. This is not the case for the ASC's output trajectory, which has a much step response once it is close to a terminal position resulting in a less efficient motion.

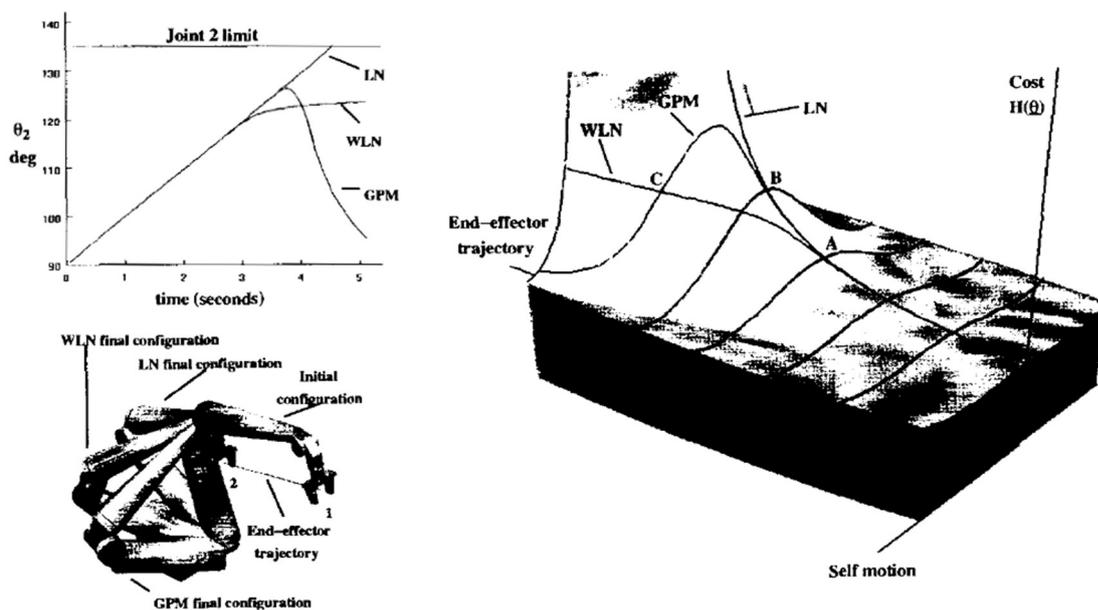


Fig. 2-3. Comparison of the Trajectory Solutions for the RMRC (Labelled as LN), ACS (Labelled as GPM) and WLN Methods [10].

2.1.4 Further Upgrades

Departing from the previously presented algorithms, several extensions and modifications have been proposed. Nevertheless, the general philosophy of incorporating a penalization which increases as the configuration moves towards one of its limits prevails. An inherent problem to this sort of algorithms is its strong tendency

to generate trajectories which move around the intermediate position between the joint's limits without benefiting from all the solution space.

This problematic can be addressed by certain workarounds. For example, *Lee et al. 2007* proposed a definition of a range of areas which proved able to improve the performance with respect to other conventional methods [11]. In the recent years, other approaches such as Nonlinear Model Predictive Control (NMPC) [12] or space transformations [13] have been proposed.

However, the tendency of a space's underutilization always remains due to the complexity of coupling the cost function's definition with a meaningful assessment of the configuration's optimality.

2.2 Topological Approach

The topological approach addresses the inverse kinematics from an analytical standpoint. This perspective aims the study of the solution space for a given input desired coordinates. Such consideration enables a more comprehensive understanding of the robot's kinematic capabilities.

Most of the researches of the inverse kinematics focus on the solution to the point problem [14, 15]. Works that addressed trajectory planning strategies based on an analytical solution to the inverse kinematics are rather scarce. The main studies in this direction were conducted by Lücke et.al in the 1990s decade [16, 17]. In these works, the self-motion topologies within the configuration space were studied. These regions are the set of configurations which result in the same coordinates for the end effector.

Using the example of a planar arm, the solution space for the point problem considering joints' limitations was studied. An example of these isokinematic regions is displayed in Fig. 2-4.

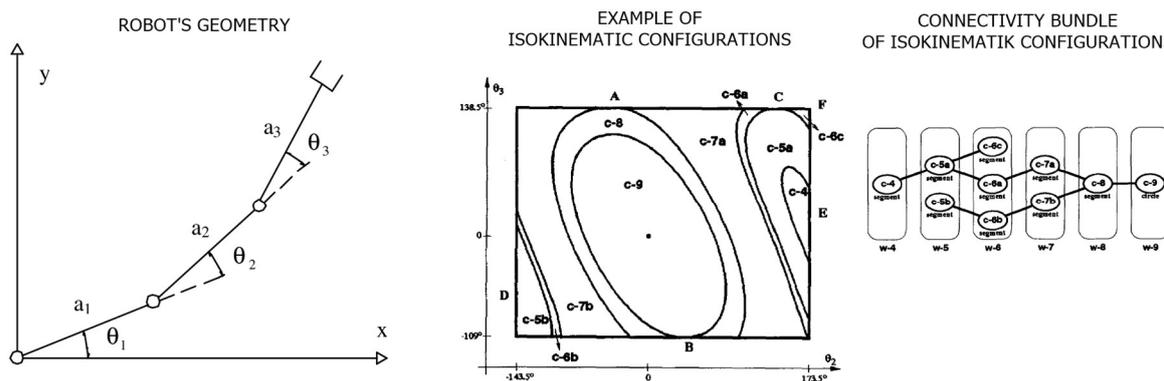


Fig. 2-4. Example of of an Isokinematic Topology for a Constrained Robot and the Conectivity Bundle of its Elements [16].

This Figure illustrates the topology of the self motion space for a given desired coordinates of the robot and for a given value of θ_1 . The actual solution space is a surface in a tridimensional space but the value of θ_1 was fixed in order to produce a slice of this surface which provides a more intuitive visualization. As it can be seen, the solution space is composed of different segments, between which transitions can take



place by varying the value of θ_1 . The associated graph which illustrates these feasible transitions is presented in Fig. 2-4 as well.

Lück proposed a global trajectory planning strategy by discretization and the use of a Breadth First Search (BFS) heuristics considering the connectivity bundle at each point of the discretized trajectory. Its approach proved to improve the execution time of the task by a factor of 50 when compared with conventional methods [17].

The main shortcoming of these methodologies is the complexity of the inverse kinematics' analytic solution even in eminently simple systems. This difficulty grows exponentially when the trajectory planning comes into play.

3 Goal of this Thesis

In order to maximize the capabilities of the RACOON-Lab it is necessary to enable its use for the recreation of complex trajectories. This requires the implementation of an algorithm capable of solving this robotic systems' inverse kinematics.

Since this facility is highly constrained and presents a low degree of redundancy, the general approaches discussed in the previous section may result in an underuse of the system's potential. Due to this, it becomes necessary to design a specific methodology which enables an optimal utilization of the facility. Therefore, the main goal of the present thesis is the development of an algorithm capable of solving the inverse kinematics problem for the RACOON-Lab. This implies to produce a methodology capable of generating real time and offline trajectories that simultaneously avoid unnecessary motions and forbidden states.

Parallel to this objective, it will be aimed to assess the feasibility of optimal execution of trajectories within the use of the present robotic system. This involves the evaluation of the suitability of the RACOON-Lab's design concept from the point of view of its kinematics. From this evaluation task it will be possible to determine the performance limits of the facility. This will provide an adequate framework for the appraisal of the developed algorithm's aptness.

Therefore, the main objective of the present thesis will be the implementation of a functional inverse kinematics algorithm and it will be coupled with the study of the facilities suitability for representing convoluted maneuvers.

4 Approach

The robot whose inverse kinematics will be studied has three five rotational degrees of freedom for achieving the three attitude angles. This provides a degree of redundancy of two, which can be regarded as low. In addition, many of the joints are severely limited. Such disposition can be excessively challenging for the application of conventional methodologies. Furthermore, the task of evaluating the system's suitability for the representation of complex maneuvers calls for a holistic approach. It will be necessary to address the problem with an integral strategy, capable of providing a comprehensive perspective of the facility's potentials.

The analytical approach is a suitable methodology for fulfilling such requirements. Furthermore, the low degree of redundancy of the involved robotic systems is an advantageous feature when considering this procedure. Due to this, the analytical approach will be the strategy followed for the development of the present project.

Therefore, the work will begin with an extensive analysis of the facility's robotic systems and their forward kinematics. This will provide a solid background for the following tasks

Departing from this study, the inverse kinematics of the facility will be addressed from an analytical standpoint in order to provide a consistent mathematical expression of the robot's configuration as a function of the desired coordinates.

The next step will be the development of algorithms for implementing trajectory planning based on the developed formulations. These results will be compared with the outcome of some of the conventional algorithms present in the literature. This will provide an appropriate foundation in order to assess the achievements of the followed approach.

The analytical approach will provide a set of tools that enable the analysis of the facility suitability's through sensitivity and singularity analyses. This study will provide a deeper understanding of the facility's capabilities and will be used for evaluating the goodness of the developed inverse kinematics algorithms.

Subsequently, the elaborated methodologies will be applied for the trajectory planning of actual orbiting spacecraft within the RACOON-Lab's framework. This will provide an actual example for a realistic scenario where the developed algorithms can be implemented.

Finally, the achieved work will be summarized and discussed, stressing its comparison with the initially presented goals and next steps and possible improvements will be proposed in an outlook chapter

5 Environment Description

The Real-Time Attitude Control and On-Orbit Navigation laboratory (RACOON-Lab) was conceived in order to provide a flexible test bed for the research of new on-orbit servicing and space debris removal technologies [18]. Since the beginning of its operations, a decade ago, several research projects have benefitted from its use. This laboratory's framework consists on several modules, but this text will pay special attention to the mechanical hardware facility.

This chapter is devoted to the comprehensive description of the RACOON-Lab with a special focus on its multibody kinematics. This exposition will be structured in two different sections. In the first one, the RACOON-Lab environment will be illustrated from an architectural standpoint. In the following section, the kinematics of the robots present in the RACOON-Lab will be described in order to provide a solid background for addressing the system's inverse kinematics in the following chapters.

5.1 System Architecture

The experimental setup of RACOON-Lab is composed of two independent robots: A Chaser and a Target. The Chaser is the element which intends to represent the controlled spacecraft, that attempts to perform on-orbit operations requiring the interaction with another cooperative or uncooperative spacecraft. This secondary element is referred as the Target and within the RACOON-Lab it is represented by the other robot: the Rotator. This element receives this name due to the fact that it can only change its orientation. Therefore, any relative translational motions between both elements is executed by the Chaser. These two elements can be used for testing new technologies and algorithms in proximity operations. In Fig. 5-1 and Fig. 5-2 examples of possible scenarios that could be represented within the RACOON-Lab are displayed.



Fig. 5-1. Artist Impression of ATV Docking with the ISS [19]

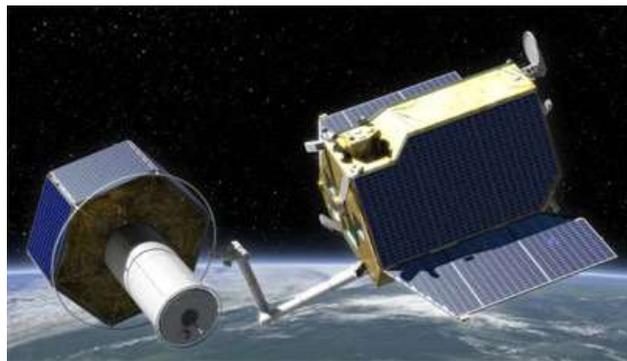


Fig. 5-2. Deutsche Orbitale Servicing Mission (DEOS) [20]

In order to reproduce the trajectories, typical for on-orbit operations, it is necessary to provide these robots with the necessary degrees of freedom that enable seamless relative motion in three-dimensional space. In total, the facility has eleven degrees of freedom, five of them are associated to the Target and the remaining six to the Chaser. Both elements and its degrees of freedom can be observed in Fig. 5-3. This disposition enables the simulation of proximity operations with real-time capabilities. A particularly

relevant simulation case is the interaction with uncooperative targets for de-orbiting purposes. In these cases, the rotator represents an uncontrolled, often defunct spacecraft which has to be captured by the chaser. Many of the novel technologies which attempt to perform such tasks rely in visual recognition algorithms [21].

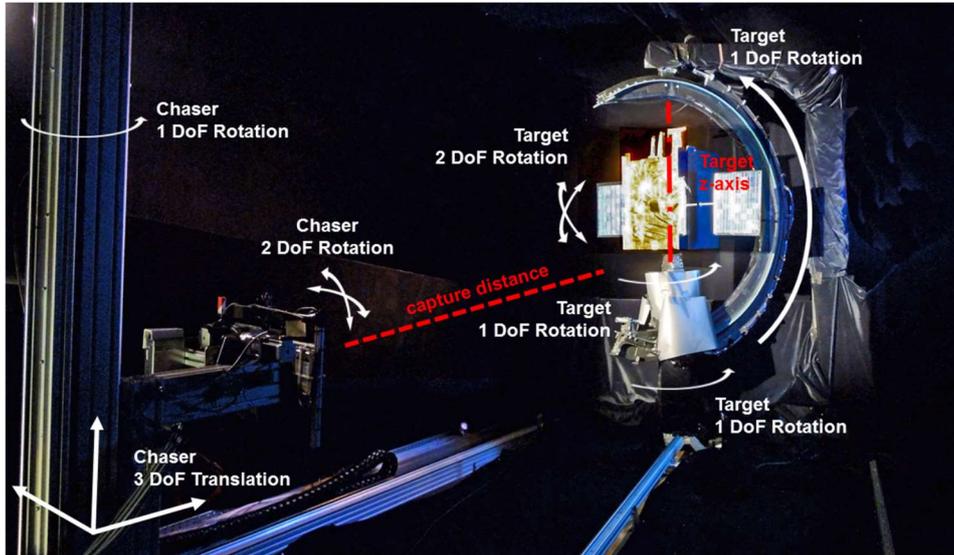


Fig. 5-3: Photography of the RACOON-Lab with all its degrees of Freedom [2]

Due to this, it is essential to recreate the same illumination conditions that the robots would experience during an actual Space mission. With the exception of interstellar missions, the two main light sources that the spacecraft receives are the Sunlight and the Earth's albedo. The former source is relevant for all missions, whilst the latter can be very significant for missions which take place in Low Earth Orbits (LEO) and MEO (Medium Earth Orbits). These lighting conditions are achieved with the use of Earth's albedo and Sun simulators, which are capable of accurately mimicking the electromagnetic spectrum of these sources of light [18].

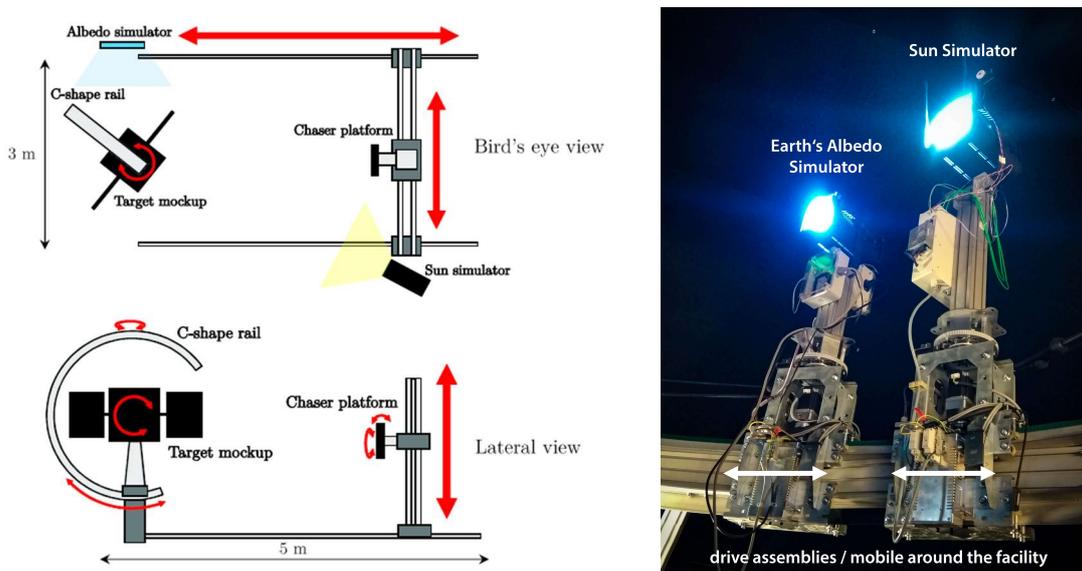


Fig. 5-4. Scheme of RACOON-Lab's Elements [22] and Illumination Instrumentation [18]

The goal of maximizing the simulations' realism influences the kinematic disposition of the robots too. The main example is the C-Shape rail which enables partial rotations of the satellite mockup around axes contained in the horizontal plane. This disposition was chosen in detriment to a closed O-Ring in order to create a clear line of sight free of interruptions between the Chaser and the Rotator. Although this concept is highly advantageous from this perspective it difficults the solution of the inverse kinematics, especially when considering continuous rotations around axes contained in the horizontal plane. This problem will be discussed in further detail in the following sections.

5.2 System Kinematics

In this section the multibody systems present in the RACOON-Lab and its kinematics are presented. In order to provide the appropriate background for this description, an initial section will introduce the basic notions of robotic kinematics. Subsequently, the direct kinematics of the robots will be described.

5.2.1 Basic Definitions

The field of kinematics is the study of the movement's geometry and its task is the description of the coordinates and configuration of bodies over time and space [23]. For the scope of the present section only the definition over space is relevant and therefore we will restrict to this aspect of the kinematics.

5.2.1.1 Coordinates in Space

The coordinates of a body in space describe six degrees of freedom, three of them refer to the bodies' position and the other three are related to the bodies' orientation. There are infinite valid ways of representing this information. For the present work, the spatial coordinates and the Euler angles with respect to the inertial frame will be the ones used, as displayed in Fig. 5-1.

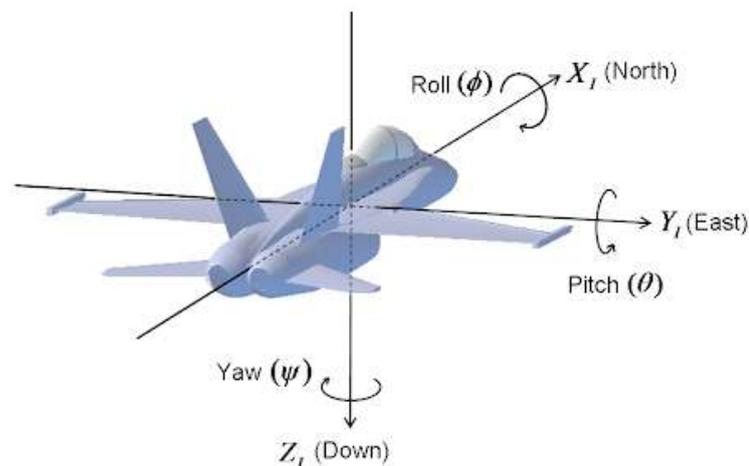


Figure 5-1. Euler Angles in the Inertial Frame [24]

Therefore, the coordinates of a given body can be expressed as:

$$W_F = (Z_F \quad \Xi)^T = (x \quad y \quad z \quad \varphi \quad \theta \quad \psi)_F^T \quad \text{Eq. (5-1)}$$

Where Z_F is the 3x1 vector of the space coordinates and Ξ is the 3x1 vector containing the Euler angles between the frame F and the inertial frame. In this notation, the sub-index F denotes the framework in which these coordinates are provided. A central aspect of the kinematics is the expression of coordinates in different frames. A frame in a tridimensional space is defined by the coordinates of its origin and by the orientation of its main axes.

If two different frames present a common origin, it is possible to transform the coordinates from one frame to another using a rotation matrix in the following fashion:

$$Z_A = M_{AB}Z_B \quad \text{Eq. (5-2)}$$

Where M_{AB} is the rotation matrix which is a function of the Euler angles between the frames A and B . The rotation matrix depends on the axis, in which the rotation takes place and on the rotation angle ρ . The rotation matrices that represent these elemental rotations are the following:

$$M_x(\rho) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\rho) & \sin(\rho) \\ 0 & -\sin(\rho) & \cos(\rho) \end{pmatrix} \quad \text{Eq. (5-3)}$$

$$M_y(\rho) = \begin{pmatrix} \cos(\rho) & 0 & -\sin(\rho) \\ 0 & 1 & 0 \\ \sin(\rho) & 0 & \cos(\rho) \end{pmatrix} \quad \text{Eq. (5-4)}$$

$$M_z(\rho) = \begin{pmatrix} \cos(\rho) & \sin(\rho) & 0 \\ -\sin(\rho) & \cos(\rho) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{Eq. (5-5)}$$

If several elemental rotations take place with a given order it is possible to summarize all of them with a single rotation matrix which is the product of all the previous rotations:

$$Z_1 = M_{12}M_{23}M_{34} \dots M_{(N-2)(N-1)}M_{(N-1)N}Z_N = M_{1N}Z_N \quad \text{Eq. (5-6)}$$

The other differential element between two different frames is the offset between their origin. If we consider the simplest case of an offset in the origin between two frames that maintain the parallelism between their axes, the change of coordinates from one frame to another can be performed with the following operation:

$$Z_A = R_{AB} + Z_B \quad \text{Eq. (5-7)}$$

Where R_{AB} represents the 3x1 vector connecting the origin of frame A and B. In the most common scenario, there is both an offset in position and orientation between two frames. In this case, the manipulation of coordinates can be expressed as:

$$Z_A = R_{AB} + M_{AB}Z_B \quad \text{Eq. (5-8)}$$

Which can be formulated in a more compact manner using the so-called homogeneous transformation:

$$\begin{pmatrix} Z_A \\ 1 \end{pmatrix} = \begin{pmatrix} M_{AB} & R_{AB} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} Z_B \\ 1 \end{pmatrix} = D_{AB} \begin{pmatrix} Z_B \\ 1 \end{pmatrix} \quad \text{Eq. (5-9)}$$

As in the other two cases, it is possible to concatenate several transformations by multiplying the matrices D :

$$D_{1N} = D_{12}D_{23}D_{34} \dots D_{(N-2)(N-1)}D_{(N-1)N} \quad \text{Eq. (5-10)}$$

In the scope of the present thesis, whenever the space coordinates Z are mentioned, they will be assumed to refer to the end effector's coordinates expressed in the inertial frame. The reference system used as the inertial one for the RACOON-Lab is displayed in Fig. 5-5. In this image the two robots present its default configuration. The following sections will discuss with further detail the meaning of these configurations and its possible variations.

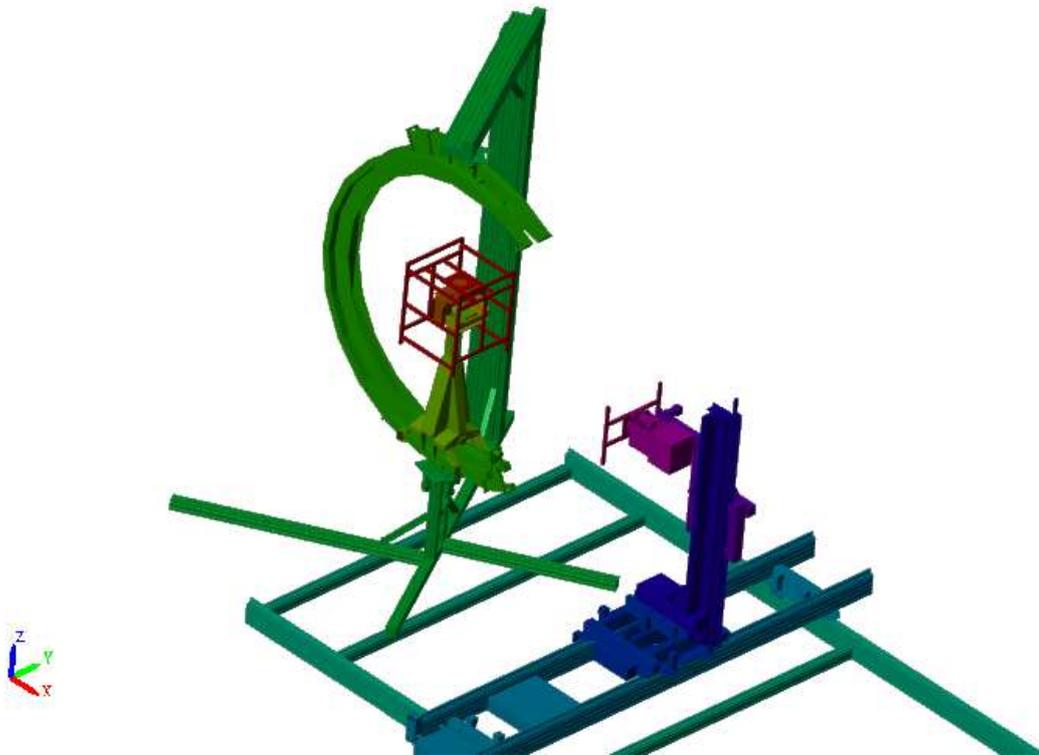


Fig. 5-5: Reference System in RACOON-Lab with Chaser and Rotator in its default positions

5.2.1.2 Robot's Configuration

Generally, robots are multibody systems, this means that they are the composition of a finite number of individual bodies. These bodies are connected with the use of joints. Depending of how these interconnections between the bodies are organized it is possible to distinguish different structures. For the scope of the present work, only chain structures are relevant. These structures are characterized by the fact that each body presents exactly one predecessor and one successor, with the exception of two bodies, that only present a neighbor element. These bodies are the base and the end effector, the base has no predecessor and the end effector has no successor [4]. Fig. 5-6 illustrates an example of a typical chained multibody system.

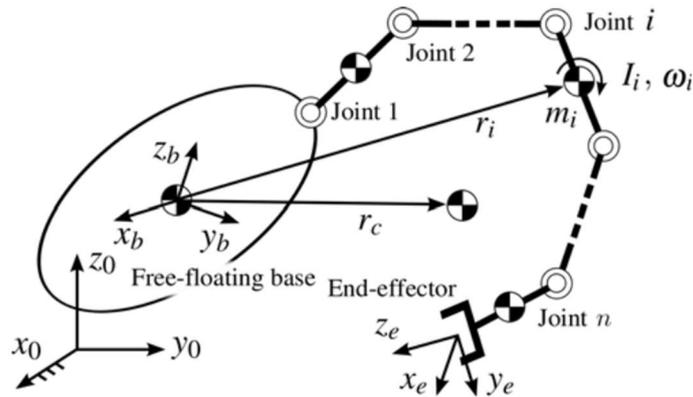


Fig. 5-6: Chain Multibody System Example [25]

The joints are critical elements in robotics as they are the responsible for the movement of the multibody system. Generally, there are two main kind of joints: revolute and prismatic. The first type allows a linear motion whilst the latter enable rotations around a certain axis. These different types of joints can be visualized in Fig: 5-7.

Revolute Joints

Prismatic Joints

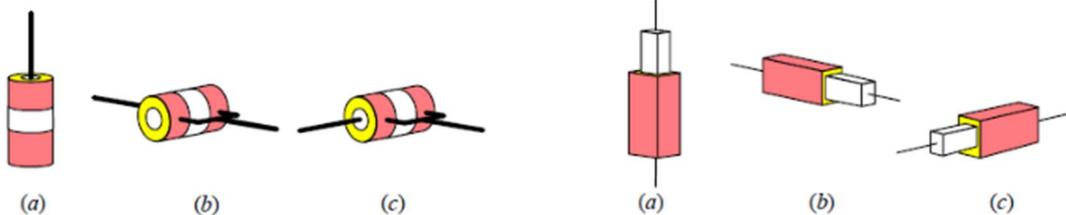


Fig: 5-7: Examples of Revolute and Prismatic Joints [26]

For the scope of the current work, we will restrain to these two kinds of joints since they are the only ones present at the RACOON-Lab. Every joint provides an additional Degree of Freedom (DoF) to the multibody that can be characterized either as an angular value or as a displacement for revolute and prismatic joints respectively. The conjunction of all these values expressed in a column vector q is what constitutes a robot's configuration. This vector suffices to describe the current state of the robot and it is possible to determine the space coordinates of every joint as a function of it. This can be done using the formulations introduced in the previous section, generating successive homogeneous transformations. For the particular case of the RACOON-Lab, the Chaser presents three prismatic joints and three revolute joints whilst the Rotator is equipped with five revolute joints. Therefore, the configuration vectors for these robots can be expressed as follows:

$$\begin{cases} q_c = (x_c \ y_c \ z_c \ \varphi_c \ \theta_c \ \psi_c)^T \\ q_R = (\alpha \ \beta \ \gamma \ \delta \ \varepsilon)^T \end{cases} \quad \text{Eq. (5-11)}$$

There are several methods available in literature which allow to automate the generation of the homogeneous transformation matrices as a function of a robot's configuration. However, since the robots used in the present work are rather simple it will not be necessary to resort to them.

5.2.1.3 Direct and Inverse Kinematics

Once the concepts of space coordinates and robot's configuration has been introduced, it is possible to state robust definitions of the direct and inverse kinematics.

The direct kinematics refers to the physical problem of obtaining the space coordinates w as a function of the robot's configuration:

$$w = f_{DK}(q) | f_{DK} : \mathbb{R}^{DoF} \rightarrow \mathbb{R}^6 \quad \text{Eq. (5-12)}$$

This task can be fulfilled easily using the equations presented in 5.2.1.1. The direct kinematics is a well-posed problem as there is a unique solution for a given robot's configuration.

The inverse kinematics refers to the opposite task and it is significantly more challenging. In this case what one wants to know is the robot configuration (or the set of robot's configurations) that can lead to a given space coordinates. This can be expressed mathematically as follows:

$$q = f_{IK}(w) | f_{IK} : \mathbb{R}^6 \rightarrow \mathbb{R}^{DoF} \quad \text{Eq. (5-13)}$$

Unlike the direct kinematics, the inverse kinematics is only well posed for the very limited number of cases where there is a bidirectional univocal relationship between a robot's configuration and its coordinates i.e. f_{IK} and f_{DK} are bijective. Very often, a given coordinate can be achieved by several configurations. Furthermore, it is also possible that due to the geometrical restrictions of a robot there is no configuration that provides a certain coordinate to the end effector.

5.2.2 Chaser Kinematics

The Chaser is equipped with six joints which provide the necessary degrees of freedom for operating in the space. This robot has three prismatic joints which can provide a linear motion across the three main axes and three revolute joints which are identified with roll, pitch and yaw angles. The transformations take place in the following order:

1. Displacement in X through a prismatic joint: x_C
2. Displacement in Y through a prismatic joint: y_C
3. Displacement in Z through a prismatic joint: z_C
4. Rotation around X through a revolute joint: φ_C
5. Rotation around Y through a revolute joint: θ_C
6. Rotation around Z through a revolute joint: ψ_C

Considering all the previously stated transformations, it is possible to obtain the homogeneous transformation D_{IE_C} which relates the inertial frame to the end-effector's frame:

$$\begin{pmatrix} x_I \\ y_I \\ z_I \\ 1 \end{pmatrix} = \begin{pmatrix} c(\theta_C) c(\psi_C) & c(\varphi_C) s(\psi_C) + c(\psi_C) s(\varphi_C) s(\theta_C) & s(\varphi_C) s(\psi_C) - c(\varphi_C) c(\psi_C) s(\theta_C) & x_C \\ -c(\theta_C) s(\psi_C) & c(\varphi_C) c(\psi_C) - s(\varphi_C) s(\theta_C) s(\psi_C) & c(\psi_C) s(\varphi_C) + c(\varphi_C) s(\theta_C) s(\psi_C) & y_C \\ s(\theta_C) & -c(\theta_C) s(\varphi_C) & c(\varphi_C) c(\theta_C) & z_C \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_E \\ y_E \\ z_E \\ 1 \end{pmatrix} \quad \text{Eq. (5-14)}$$

Where $c(x)$ and $s(x)$ stand for $\cos(x)$ and $\sin(x)$ respectively.

Although the Chaser has more degrees of freedom, its direct and inverse kinematics are significantly simpler than the ones of the Rotator. The reason for this is that both the revolute and prismatic joints are directly coupled to the spatial coordinates. Therefore, this robot belongs to the exceptionally reduced group of multibodies which have a bijective function that links configuration to space coordinates. Due to this, the formulation of the inverse kinematics is extremely simple.

5.2.3 Rotator Kinematics

Unlike the Chaser, the Rotator does not have any prismatic joint and thus cannot experience displacements. Therefore, its only freedom in space is with respect to its orientation. Another important distinction of the chaser is that three out of its five revolute joints are limited in space. This is a critical aspect when addressing the inverse kinematics, but it does not make a significant difference regarding the direct kinematics.

The first joint is the azimuth and it is referred with the angle α . It provides a rotation around the vertical axis. Since this rotation is the first one performed with respect to the inertial frame it is directly translated into yaw of the mockup spacecraft.

The Second Joint is the elevation angle β and it provides a rotation about axes contained in the horizontal plane. This actuator is limited due to the C-Ring Design. Its approximate boundaries are from -180° to 0° as it can be seen in Fig. 5-8. The robot in the actual facility is able to accommodate a range slightly larger. However, there are several uncertainties surrounding the actual value. First it is necessary to determine the exact values experimentally. Second, currently there is no safety lock that can prevent the robot from exiting the track and generating a dangerous situation. Therefore, even if the actual limits were known, it would be necessary to assume a slightly smaller range for the sake of safety. Due to these reasons, the range of $[-180^\circ - 0^\circ]$ will be assumed for the present work since it provides a conservative approach.

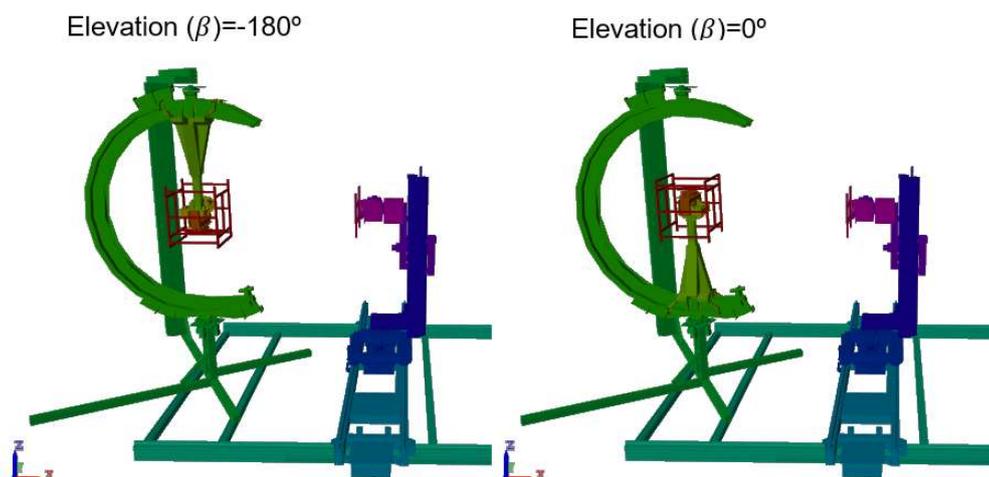


Fig. 5-8: Boundaries of the Elevation Angle

The third and fourth rotations are performed by the A and B Nick angles which will be referred in future derivations as γ and δ respectively. These angles are strongly constrained since they can only rotate with a range of approximately $\pm 30^\circ$ as it can be seen in Fig. 5-9.

Too large angle name, as big as text, symbol

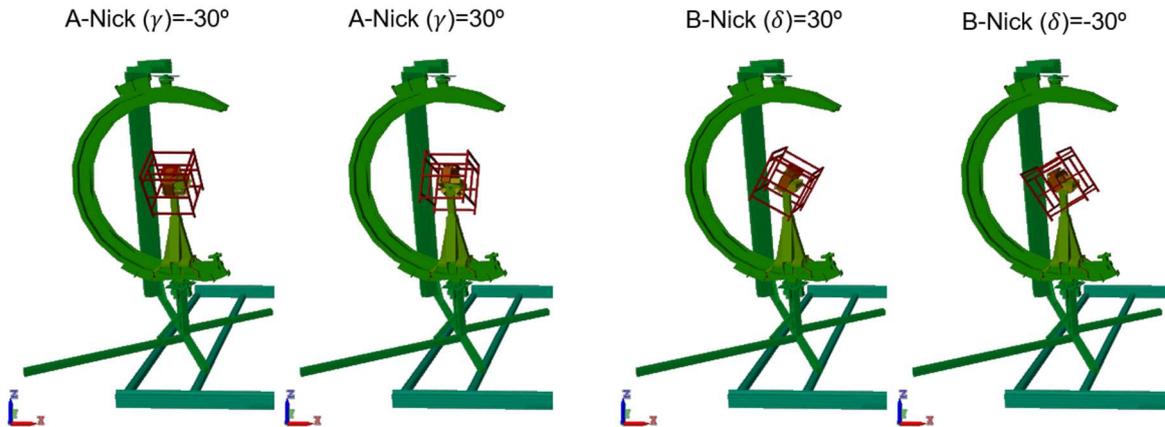


Fig. 5-9: Limits of the Nick Angles

The last rotation takes place around the Z axis and it is referred as the polar angle, ϵ . This rotational degree of freedom is unrestricted as in the case of the azimuthal joint.

These five degrees of freedom are theoretically enough to enable the full freedom of orientation in space. Nevertheless, this task might be strongly hindered due to the strong imitations in the nick and elevation angles. As it was previously introduced, the motivation behind the elevation angle's limitation was to provide a diaphanous space between the Chaser and the Rotator. Such feature is highly beneficial if algorithms based on visual recognition are to be tested within the facility. However, this greatly increases the difficulty of performing rotations around axes contained in the XY plane. In these circumstances, when the elevation angle is close to its minimum, it is necessary that the azimuth and polar angles flip 180° degrees so that the rotator can continue the rotation. This sequence is illustrated in Fig. 5-10.

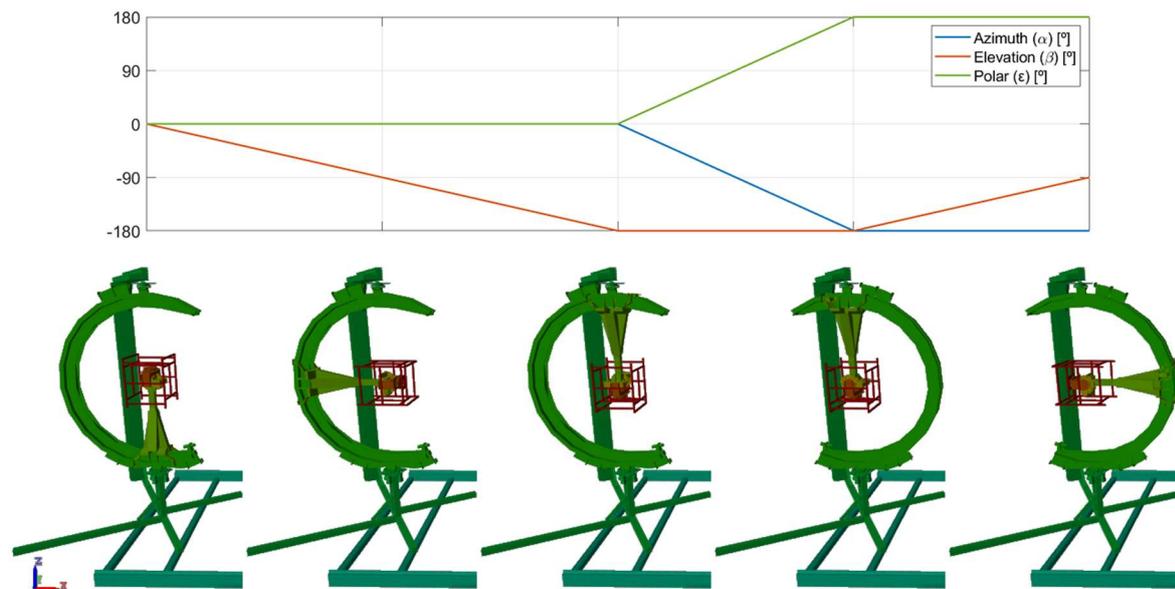


Fig. 5-10: Possible Rotation Sequence Around the X Axis for RACOON-Lab's Target

If the rotations of the azimuth and polar joints take place completely just after the minimum elevation angle has been reached as displayed in Fig. 5-10 it would require a sudden abort of the simulation as an instantaneous infinite angular velocity would be required. Therefore, a suitable trajectory would gradually rotate the azimuth and polar angles as the elevation angle approaches to its minimum value.

This sort of movement can be addressed from an intuitive perspective very easily, but it has proven extremely difficult to implement such behavior in trajectory control routines. Most algorithms do not execute the rotations of azimuth and polar in time. This often results with the robot ending in a locking position and performing an abrupt transition. Such condition takes place when several joints of the robot have reached its limit value and the only possible way of the robot to keep its movement requires to surpass these limits or execute an abrupt transition. An example of such scenario is displayed in Fig. 5-11 where the elevation and one of the nick angles are saturated and it is not possible to continue a rotation around the X axis in a seamless way. The whole degrees of freedom that characterize the Rotator's configuration are summarized in Table 5-1. In this table, the limits presented for the constrained joints are approximated and rather conservative. Nevertheless, these values ensure the validity of the work as it was experimentally determined that they are close to match the actual amounts.

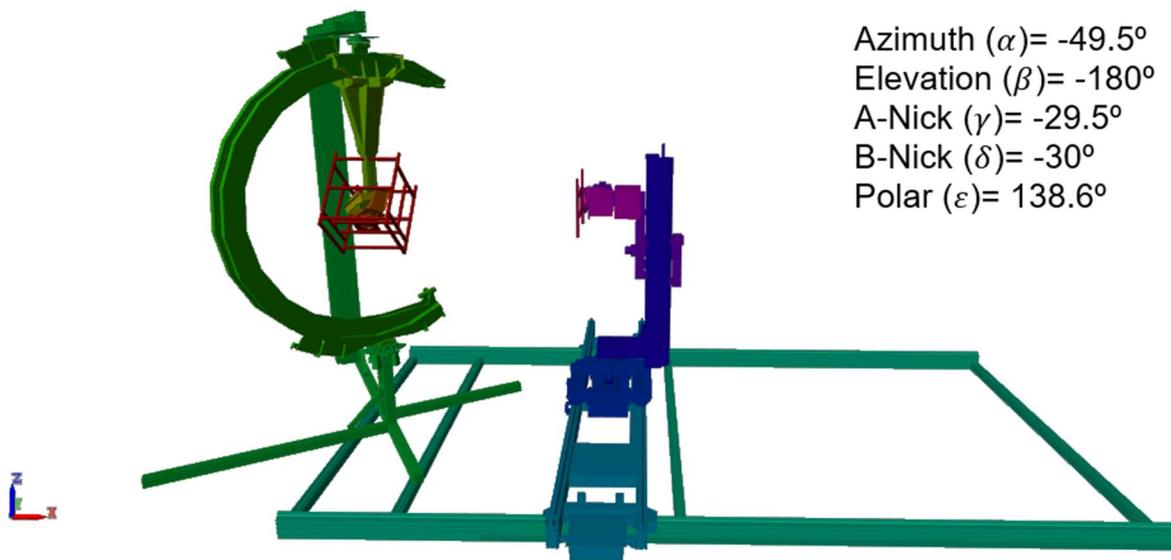


Fig. 5-11. Example of a Locking Configuration for the RACOON-Lab's Rotator

Table 5-1: Summary of Joints of the Rotator

Joint Order	Name	Symbol	Axis	Minimum	Maximum
1	Azimuth	α	Z	-180°	+180°
2	Elevation	β	Y	-180°	0°
3	A-Nick	γ	X	-30°	+30°
4	B-Nick	δ	Y	-30°	+30°
5	Polar	ϵ	Z	-180°	+180°

As in the previous case, it is possible to obtain the homogeneous transformation matrix considering the transformations that take place over all the robot's joints. However, since there are no prismatic joints, it suffices the use of a rotation matrix, which converts the vectors in the rotator's end effector to the inertial frame:

$$\begin{pmatrix} x_I \\ y_I \\ z_I \end{pmatrix} = M_Z(\alpha)M_Y(\beta)M_X(\gamma)M_Y(\delta)M_Z(\varepsilon) \begin{pmatrix} x_{E_R} \\ y_{E_R} \\ z_{E_R} \end{pmatrix} = M_{I_{E_R}} \begin{pmatrix} x_{E_R} \\ y_{E_R} \\ z_{E_R} \end{pmatrix} \quad \text{Eq. (5-15)}$$

With:

$$\begin{aligned} M_{I_{E_R}}(1,1) &= c(\varepsilon) \left(s(\delta)(s(\alpha)s(\gamma) - c(\alpha)c(\gamma)s(\beta)) + c(\alpha)c(\beta)c(\delta) \right) - s(\varepsilon)(c(\gamma)s(\alpha) + c(\alpha)s(\beta)s(\gamma)) \\ M_{I_{E_R}}(1,2) &= c(\varepsilon) \left(s(\delta)(c(\alpha)s(\gamma) + c(\gamma)s(\alpha)s(\beta)) - c(\beta)c(\delta)s(\alpha) \right) - s(\varepsilon)(c(\alpha)c(\gamma) - s(\alpha)s(\beta)s(\gamma)) \\ M_{I_{E_R}}(1,3) &= c(\varepsilon)(c(\delta)s(\beta) + c(\beta)c(\gamma)s(\delta)) + c(\beta)s(\gamma)s(\varepsilon) \\ M_{I_{E_R}}(2,1) &= c(\varepsilon)(c(\gamma)s(\alpha) + c(\alpha)s(\beta)s(\gamma)) + s(\varepsilon) \left(s(\delta)(s(\alpha)s(\gamma) - c(\alpha)c(\gamma)s(\beta)) + c(\alpha)c(\beta)c(\delta) \right) \\ M_{I_{E_R}}(2,2) &= c(\varepsilon)(c(\alpha)c(\gamma) - s(\alpha)s(\beta)s(\gamma)) + s(\varepsilon) \left(s(\delta)(c(\alpha)s(\gamma) + c(\gamma)s(\alpha)s(\beta)) - c(\beta)c(\delta)s(\alpha) \right) \\ M_{I_{E_R}}(2,3) &= s(\varepsilon)(c(\delta)s(\beta) + c(\beta)c(\gamma)s(\delta)) - c(\beta)c(\varepsilon)s(\gamma) \\ M_{I_{E_R}}(3,1) &= c(\delta)(s(\alpha)s(\gamma) - c(\alpha)c(\gamma)s(\beta)) - c(\alpha)c(\beta)s(\delta) \\ M_{I_{E_R}}(3,2) &= c(\delta)(c(\alpha)s(\gamma) + c(\gamma)s(\alpha)s(\beta)) + c(\beta)s(\alpha)s(\delta) \\ M_{I_{E_R}}(3,3) &= c(\beta)c(\gamma)c(\delta) - s(\beta)s(\delta) \end{aligned}$$

This matrix is enough for describing the direct kinematics of the Rotator. Since this multibody has no freedom of motion around position, its space coordinates only consist on the Euler angles:

$$W_R = (\varphi_R \quad \theta_R \quad \psi_R)^T \quad \text{Eq. (5-16)}$$

Departing from the equations for the direct kinematics, it is possible to do a first assessment of the Robot's capabilities. By generating random configurations within the physical capabilities, the resulting coordinates can be calculated and its Probability Distribution Function (PDF) can be plotted. This function is illustrated in Fig. 5-12. Since the yaw angle can be variated freely with the azimuthal joint, this function is independent of this coordinate.

This graph reveals several patterns of the robot's kinematics. It can be seen that the pitch and roll multiples of 180° present a higher probability density compared to the other combinations. These areas correspond to the case in which the end effector's vertical axis is quasi aligned with the vertical axis in the inertial frame. These regions with easy access extend with a radius of approximately 30° , which corresponds to the range of the nick angles. This is a logical result as the higher probability densities are a consequence of the easier reachability. For situations in which the end effector's vertical axes are almost aligned, it is possible to resort either to the nick angles or the other joints in order to achieve the desired attitude. However, as the angle between the vertical axes increases, the possibilities of using the nick angles are narrowed.

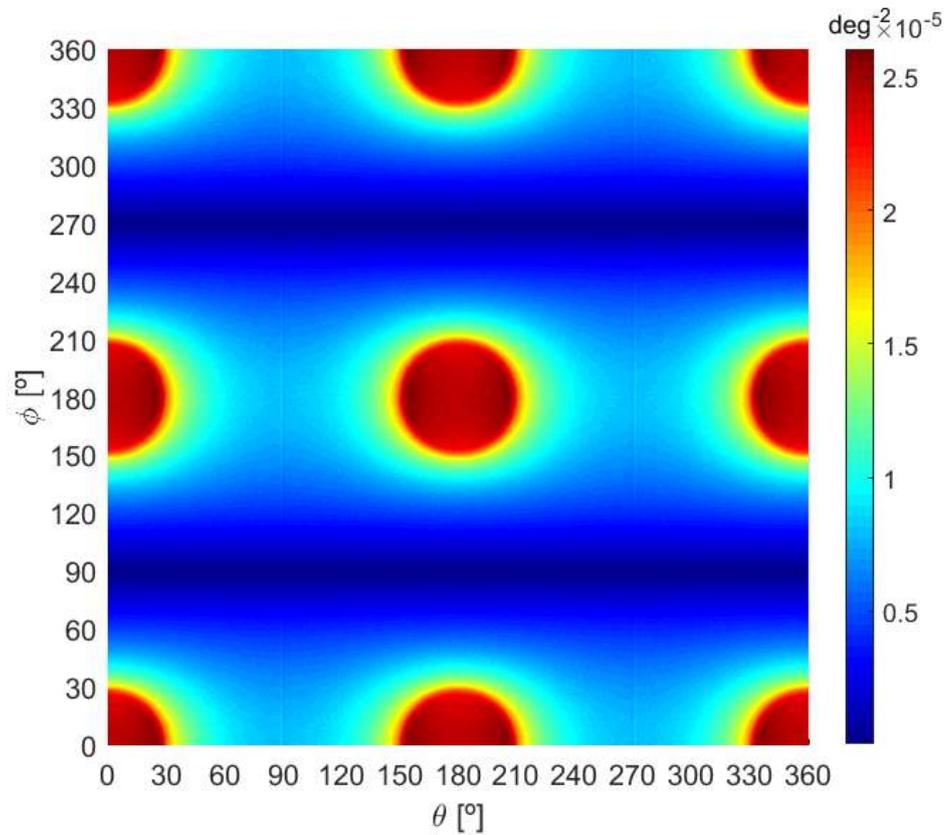


Fig. 5-12. Probability Density Function of the Robots Coordinates as a Function of the Roll and Pitch Angles

6 Solution of the Inverse Kinematics

Once the direct kinematics of the Rotator have been introduced, it is possible to address the inverse kinematics problem in a robust way. Unlike most of the usual works, here the inverse kinematics is approached analytically. This means, that the manifold of solutions is determined for a given desired pose instead of for its time derivative. Furthermore, the usual approach is followed too, and in the second section of this chapter the inverse kinematics will be solved in the velocity domain. This will be useful for comparing the performance of conventional algorithms with respect to the one developed in this thesis.

6.1 Analytical Solution

The analytical approach intends to find a robust formulation that links the space coordinates to the manifold of configurations that leads to such coordinates. In this section, the process for reaching such expression for the case of the RACOON-Lab's Rotator is presented and the performance of the described method is assessed.

In order to do this, the problem is initially stated stressing the particularities of the considered framework. Subsequently, an overview of the solution strategy is presented in an intuitive way. Finally, this strategy is executed detailing the involved steps. During this exposition, some examples of the inverse kinematics are computed in order to provide a mean for an initial assessment of the proposed methodology. These first evaluations are complemented with the final sub-section, where the developed algorithm is tested with random inputs in order to prove its robustness.

6.1.1 Problem Statement

As it was introduced in the previous chapter, the inverse kinematics problem for the RACOON-Lab is restricted to the attitude coordinates. Therefore, the inputs will be the Euler angles, and the output the robot configurations that lead to such angles. Therefore, the problem can be stated as follows:

Given the desired Euler angles $\varphi_d, \theta_d, \psi_d$, which provide the rotation matrix:

$$M_d(\Xi) = M_d(\varphi_d, \theta_d, \psi_d) = M_z(\psi_d)M_y(\theta_d)M_x(\varphi_d) \quad \text{Eq. (6-1)}$$

Find the manifold σ , consistent of all configurations $q = (\alpha, \beta, \gamma, \delta, \varepsilon)^T$ that fullfills the following equality:

$$q \in \sigma(\Xi) \leftrightarrow \{M_{IER}(q) = M_d(\Xi)\} \wedge \{q_{i,min} \leq q_i \leq q_{i,max} \forall i\} \quad \text{Eq. (6-2)}$$

With M_{IER} being the rotation matrix from the Rotator's end effector to the inertial frame and $q_{i,min/max}$ denotes the minimum/maximum configuration of the i^{th} entry of the configuration's vector.

6.1.2 Strategy Overview

The matrixial equality presented in the problem statement can be split into the following nine individual equations:

$$M_{IER}(1,1) = c(\varepsilon) \left(s(\delta)(s(\alpha)s(\gamma) - c(\alpha)c(\gamma)s(\beta)) + c(\alpha)c(\beta)c(\delta) \right) - s(\varepsilon)(c(\gamma)s(\alpha) + c(\alpha)s(\beta)s(\gamma)) = c(\theta_d)c(\psi_d) \quad \text{Eq. (6-3)}$$

$$M_{IER}(1,2) = c(\varepsilon) \left(s(\delta)(c(\alpha)s(\gamma) + c(\gamma)s(\alpha)s(\beta)) - c(\beta)c(\delta)s(\alpha) \right) - s(\varepsilon)(c(\alpha)c(\gamma) - s(\alpha)s(\beta)s(\gamma)) = c(\varphi_d)s(\psi_d) + c(\psi_d)s(\varphi_d)s(\theta_d) \quad \text{Eq. (6-4)}$$

$$M_{IER}(1,3) = c(\varepsilon)(c(\delta)s(\beta) + c(\beta)c(\gamma)s(\delta)) + c(\beta)s(\gamma)s(\varepsilon) = s(\varphi_d)s(\psi_d) - c(\varphi_d)c(\psi_d)s(\theta_d) \quad \text{Eq. (6-5)}$$

$$M_{IER}(2,1) = c(\varepsilon)(c(\gamma)s(\alpha) + c(\alpha)s(\beta)s(\gamma)) + s(\varepsilon) \left(s(\delta)(s(\alpha)s(\gamma) - c(\alpha)c(\gamma)s(\beta)) + c(\alpha)c(\beta)c(\delta) \right) = -c(\theta_d)s(\psi_d) \quad \text{Eq. (6-6)}$$

$$M_{IER}(2,2) = c(\varepsilon)(c(\alpha)c(\gamma) - s(\alpha)s(\beta)s(\gamma)) + s(\varepsilon) \left(s(\delta)(c(\alpha)s(\gamma) + c(\gamma)s(\alpha)s(\beta)) - c(\beta)c(\delta)s(\alpha) \right) = c(\varphi_d)c(\psi_d) - s(\varphi_d)s(\theta_d)s(\psi_d) \quad \text{Eq. (6-7)}$$

$$M_{IER}(2,3) = s(\varepsilon)(c(\delta)s(\beta) + c(\beta)c(\gamma)s(\delta)) - c(\beta)c(\varepsilon)s(\gamma) = c(\psi_d)s(\varphi_d) + c(\varphi_d)s(\theta_d)s(\psi_d) \quad \text{Eq. (6-8)}$$

$$M_{IER}(3,1) = c(\delta)(s(\alpha)s(\gamma) - c(\alpha)c(\gamma)s(\beta)) - c(\alpha)c(\beta)s(\delta) = s(\theta_d) \quad \text{Eq. (6-9)}$$

$$M_{IER}(3,2) = c(\delta)(c(\alpha)s(\gamma) + c(\gamma)s(\alpha)s(\beta)) + c(\beta)s(\alpha)s(\delta) = -c(\theta_d)s(\varphi_d) \quad \text{Eq. (6-10)}$$

$$M_{IER}(3,3) = c(\beta)c(\gamma)c(\delta) - s(\beta)s(\delta) = c(\varphi_d)c(\theta_d) \quad \text{Eq. (6-11)}$$

Since the Euler angles φ_d , θ_d , ψ_d are given as an input, the values of each entry can be assumed to be known. Therefore, the set of equations that must be fulfilled can be re-written as:

$$M_{IER} = \begin{pmatrix} f(\alpha, \beta, \gamma, \delta, \varepsilon) & f(\alpha, \beta, \gamma, \delta, \varepsilon) & f(\beta, \gamma, \delta, \varepsilon) \\ f(\alpha, \beta, \gamma, \delta, \varepsilon) & f(\alpha, \beta, \gamma, \delta, \varepsilon) & f(\beta, \gamma, \delta, \varepsilon) \\ f(\alpha, \beta, \gamma, \delta) & f(\alpha, \beta, \gamma, \delta) & f(\beta, \gamma, \delta) \end{pmatrix} = \begin{pmatrix} M_d(1,1) & M_d(1,2) & M_d(1,3) \\ M_d(2,1) & M_d(2,2) & M_d(2,3) \\ M_d(3,1) & M_d(3,2) & M_d(3,3) \end{pmatrix} \quad \text{Eq. (6-12)}$$

At a first glimpse, it could be concluded that the problem is an overdetermined system with five unknowns and nine equations. However, this is far from being accurate. Due to the properties of the rotation matrices, they already include several constrains. Indeed, it can be shown that a rotation matrix in the tridimensional space only has three degrees of freedom. This means, that it is possible to use three known entries in order to determine the other six.

However, not every freely chosen trio of entries leads unambiguously to the other six entries. Some unfavorable combinations may lead to two possible values for some or all the undefined entries or even to infinite possibilities. This problem will be addressed at the end of the section, but at this stage it will remain ignored. Under this assumption, it is possible to choose any three of the nine equalities provided for solving the inverse kinematics problem.

The RACOON Lab's rotator has five degrees of freedom that can be used for achieving the three chosen equalities. This means that our problem is indeed undetermined with two degrees of freedom.

Therefore, a possible procedure for determining the whole domain of solutions could be to define two degrees of freedom of the robot: q_a and q_b as parameters, and then determine the value of the remaining angles as a function of the parametrized variables and the input desired attitude matrix:

$$q_i = f_i(M_d, q_a, q_b) \text{ with } \begin{cases} i \neq a \\ i \neq b \end{cases} \quad \text{Eq. (6-13)}$$

6.1.3 Individual Solutions

At this point, it is necessary to introduce a useful result which will be applied repeatedly in this approach for solving the inverse kinematics.

Let us consider an entry of a rotation matrix $M(i, j)$ with n degrees of freedom. Each of these degrees of freedom corresponds to the angle of a joint. We want to find a combination of all these angles that make $M(i, j)$ take the value of m . If we constrain $n - 1$ of all the possible angles only one angle ω remains free, then it is possible to rewrite the equation of our problem as:

$$K_1 \sin(\omega) + K_2 \cos(\omega) = m \quad \text{Eq. (6-14)}$$

Where K_1 and K_2 are a function of the degrees of freedom that we have constrained. This equation has two solutions:

$$\omega = \begin{cases} -\log\left(\frac{mj - \sqrt{K_1^2 + K_2^2 - m^2}}{K_1 + K_2j}\right) \cdot j \\ -\log\left(\frac{mj + \sqrt{K_1^2 + K_2^2 - m^2}}{K_1 + K_2j}\right) \cdot j \end{cases} \quad \text{Eq. (6-15)}$$

And the condition for these two solutions to be within the real domain is:

$$\omega \in \mathbb{R} \Leftrightarrow \frac{K_1^2}{m^2} + \frac{K_2^2}{m^2} \geq 1 \quad \text{Eq. (6-16)}$$

If the result of ω lays within the imaginary domain, then it is physically impossible to reach the input desired attitude with the given configuration of the other degrees of freedom. With this result, it is possible to individually address each equation associated to every entry of the attitude matrix, in which only one of its angles remains unknown.

6.1.4 Complete Solution

Under the previously enunciated assumptions it is possible to sketch a general blueprint for solving the inverse kinematics for a given desired attitude. First of all, $n - 3$ angles must be predefined. Then, it is necessary to find entries that only present an undefined angle and determine it with Eq. (6-15). This procedure has to be executed thrice in order to compute the 3 non-predefined degrees of freedom. The following sections describe this method in detail for the case of the RACOON-Lab.

6.1.4.1 Determination of the Constrained angles: β, γ, δ

The first step consists on choosing which $n - 3$ angles must be pre-defined and choose which entry must be the first one used to begin the calculation of the remaining undefined 3 angles. Since in this case two angles must be pre-defined, the ideal starting point is an entry that has three degrees of freedom.

If one has a closer look to the individual entries of the attitude matrix in Eq. (6–12), it can be observed how the degrees of freedom of each entry vary. For example, the entries in the XY plane (rows and columns from one to two) have five degrees of freedom, whilst the entry $M(3,3)$ has only three. This is an ideal starting point. Therefore, the initial step will be to solve the following equation:

$$\cos(\beta) \cos(\gamma) \cos(\delta) - \sin(\beta) \sin(\delta) = M(3,3) \quad \text{Eq. (6–17)}$$

At this point, there are three different pathways to follow:

1. Define γ and δ ($q_a = \gamma, q_b = \delta$) and then calculate β with Eq. (6–15)
2. Define β and δ ($q_a = \beta, q_b = \delta$) and then calculate γ with Eq. (6–15)
3. Define β and γ ($q_a = \beta, q_b = \gamma$) and then calculate δ with Eq. (6–15)

In order to decide which strategy is the most suitable it is necessary to consider the physical limitations present at the Rotator. If all the degrees of freedom could range from 0 to 360 °, it wouldn't make a difference to choose any of the three different possibilities. However, this is not the case. The angles that are defined at the beginning q_a and q_b can be easily set with physically possible values. However, there is no control over the three remaining degrees of freedom. This implies, that they may take any value, even if it is outside its physical capabilities. Therefore, it is desirable that the degrees of freedom that we don't pre-define are the less constrained ones.

Since the nick angles are the ones which are more strictly constrained, they are the best candidates for being the ones defined at the beginning of the algorithm. Therefore, the optimal strategy at this first stage will be to define the values of the nick angles γ and δ and then calculate the elevation angle β , since this angle has a much wider range. If this procedure is followed, it is necessary to consider the following equation:

$$K_1 \sin(\beta) + K_2 \cos(\beta) = m \quad \text{Eq. (6–18)}$$

With:

$$K_1 = -\sin(\delta)$$

$$K_2 = \cos(\gamma) \cos(\delta)$$

$$m = M_d(3,3) = \cos(\varphi_d) \cos(\theta_d)$$

Although it is theoretically possible to choose any desired combination for the nick angles, it is important to argument the implication of these values in the global solution to the robot's configuration. The reason for this is that it is possible that for some combinations of γ and δ there may not exist a solution to the inverse kinematics. Indeed, it will be shown that in certain extreme cases there is a unique value of one of the nick angles that can lead to a physically meaningful solution. In such cases, the

manifold of solutions to the inverse kinematics would be confined to a differential part of the configuration's domain. The aforesaid situation could significantly hinder the goal of computing the solution to the inverse kinematics, as it would not exist in the majority of the domain. Therefore, it is crucial to determine the regions where the solution to the inverse kinematics is circumscribed.

In order to assess this condition, it is necessary to study the relationship of the solution of the elevation angle β with the values chosen for the nick angles. The nick angles determine the coefficients K_1 and K_2 Eq. (6–18) and the squared sum of this values $K_1^2 + K_2^2$ must be higher than the square of the entry $M_d(3,3)$. Fig. 6-1 illustrates the value of this squared sum for the range of values considered for the nick angles.

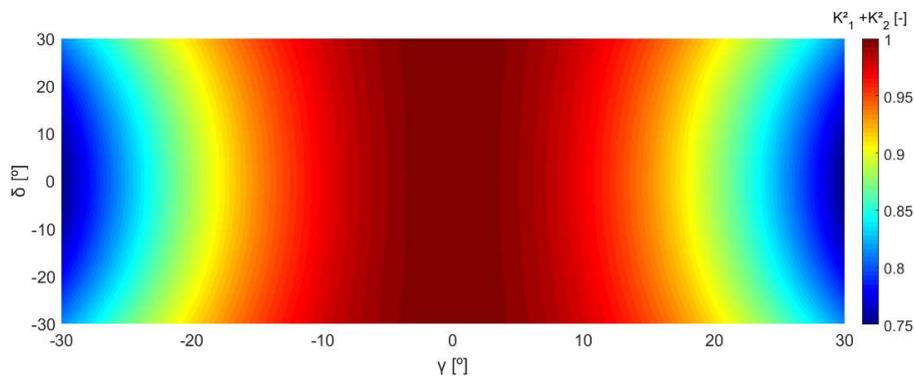


Fig. 6-1: $K_1^2 + K_2^2$ for the Determination of the Elevation angle

As it can be seen, if the value of $M_d^2(3,3)$ is below 0.75 any combination of γ and δ can provide a real value for the elevation angle β . However, as the value of $M_d^2(3,3)$ increases, the range of valid combinations decreases. For the extreme case of $M_d^2(3,3) = 1$, the A-Nick angle can only take the value of 0° . This case corresponds to the situation in which the Rotator's Z axis is aligned with the inertial frame's Z axis. If this orientation is visualized, it becomes absolutely obvious that in this case any slight rotation in the A-Nick does not allow a complete alignment between these two vertical axes. The value of $M_d(3,3)$ corresponds to the projection of the end effector's vertical component into the vertical. Therefore, the angle between the end effector's vertical component and the horizontal plane can be calculated as $\chi = \arcsin(M_d(3,3))$. This enables a more intuitive interpretation of the previously presented results, as it is possible to plot the minimum value of χ for which a given combination of Nick angles is physically possible. This can be visualized in Fig. 6-2.

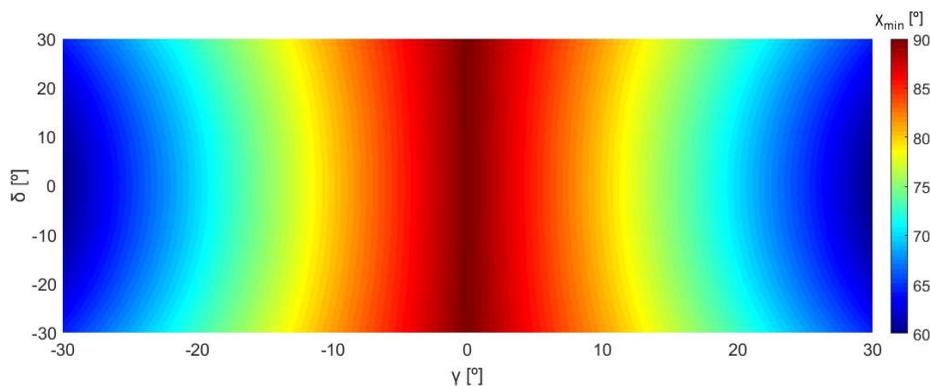


Fig. 6-2: Minimum Angle between the End Effector's Vertical Axes and the Horizontal Plane for Making

a Combination of γ and δ feasible.

Now that the relationship between the nick angles and the solution's feasibility has been introduced, it is possible to calculate the ranges of nick angles that are valid for solving the inverse kinematics.

For a given $m^2 = M_d^2(3,3)$, it is necessary that $K_1^2 + K_2^2$ presents a higher value. It can be seen, that for a given δ and m there is a range of γ which will lead to a real solution for the elevation angle β . The lower and upper bounds of this range can be computed easily with the condition $K_1^2(\gamma, \delta) + K_2^2(\gamma, \delta) > m^2$:

$$\begin{aligned} \gamma_{low} &= -\operatorname{acos}\left(\frac{\sqrt{(m - \sin(\delta))(m + \sin(\delta))}}{\cos(\delta)}\right) \\ \gamma_{up} &= \operatorname{acos}\left(\frac{\sqrt{(m - \sin(\delta))(m + \sin(\delta))}}{\cos(\delta)}\right) \end{aligned} \quad \text{Eq. (6-19)}$$

Therefore, for a given m and δ it is possible to determine the range of γ where the combination of γ and δ will lead to a non-imaginary value for β . Finally, it is necessary to introduce the geometrical restrictions. Under this consideration, it is possible to state that for a given δ and $m = M(3,3)$, the physically meaningful values of γ will range from γ_0 to γ_1 :

$$\begin{aligned} \gamma_0(\delta, m) &= \max\left\{-\operatorname{acos}\left(\frac{\sqrt{(m - \sin(\delta))(m + \sin(\delta))}}{\cos(\delta)}\right), \gamma_{min}\right\} \\ \gamma_1(\delta, m) &= \min\left\{\operatorname{acos}\left(\frac{\sqrt{(m - \sin(\delta))(m + \sin(\delta))}}{\cos(\delta)}\right), \gamma_{max}\right\} \end{aligned} \quad \text{Eq. (6-20)}$$

Then, for any values for $\delta \in [\delta_{min}, \delta_{max}]$ and $\gamma \in [\gamma_{min}, \gamma_{max}]$ it is possible to compute two solutions for β using Eq. (6-15). However, there is the possibility that $\beta_{rand} \notin [\beta_{min}, \beta_{max}]$. In this case, there is no point in continuing and the procedure and it should be aborted. Nevertheless, this is rather unlikely since the range of values that the elevation angle β can take is rather large. With all the results presented in this section it is now possible to determine the manifold of the constrained angles for a given value of the entry $M(3,3)$. This is illustrated in Fig. 6-3, where the obtained manifolds have been plotted for increasing values of the angle χ . These graphs clearly illustrate the tendency of decreasing the solution space in the nick angles as the horizontal axes tend to be aligned. A more detailed observation reveals that the manifold tends to converge to the line $\beta = -\delta$ as the pitch angle tends to 90° , which is also in good agreement with the intuitive approach.

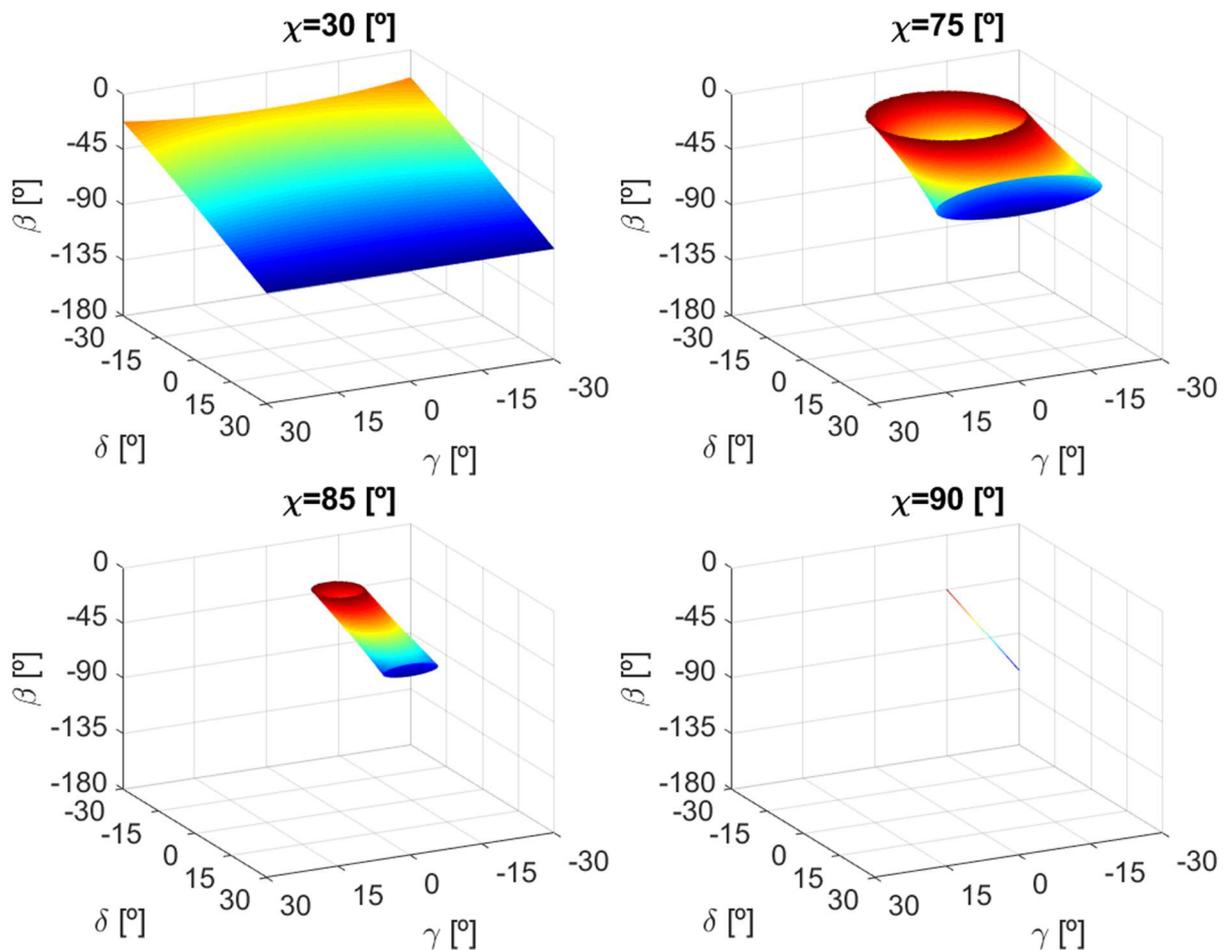


Fig. 6-3: Manifold of the Constrained Joints for Different Values of χ

6.1.4.2 Calculation of the Unconstrained Joints: α, ε

At this point, three out of the five degrees of freedom are known. The next step is to determine the values of the remaining angles α and ε in order to fulfill Eq. (6–20). Leaving these variables as the last ones to determine is highly convenient since they are not constrained and can take values ranging from 0 to 360°. A closer look to the rotation matrix reveals that there are several possibilities for proceeding. With β, γ and δ known it is possible to use any of the entries $M(1,3)$ and $M(2,3)$ in order to determine ε or any of the entries $M(3,1)$ and $M(3,2)$ in order to determine the value of α . This provides 4 possibilities for determining the fourth degree of freedom of the robot. Afterwards, the remaining degree of freedom can be obtained using the other two lateral entries or the four central entries of the matrix $M(1/2,1/2)$. This provides a total of six possible procedures for determining the fifth and last degree of freedom once the other four are known.

There, there are is a total of 24 possibilities for executing the second and third steps that will lead to the values of α and ε . These paths are illustrated in Fig. 6-4 and summarized in Table 6-1.

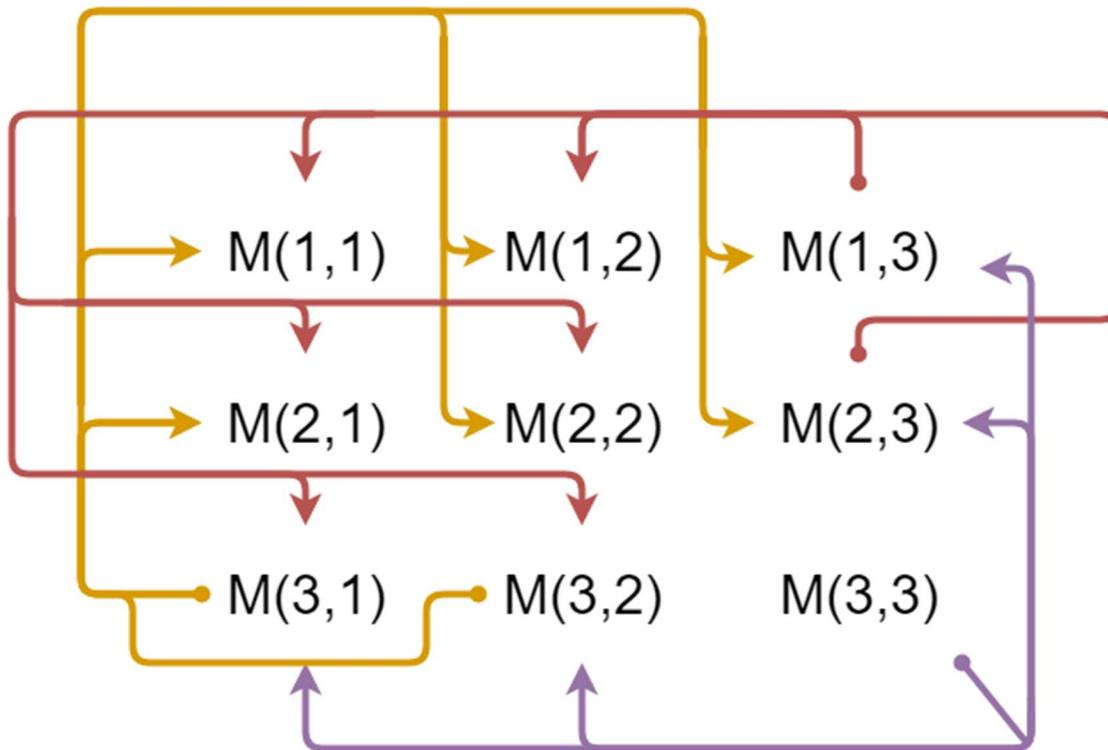


Fig. 6-4. Diagram of Possibilities for Calculating the Azimuth and Polar Joints

Table 6-1: Summary of Possibilities for Calculating the Azimuth and Polar Joints

Possibility	Step 2	Step 3
1	$\alpha = f(\beta, \gamma, \delta, M(3,1))$	$\varepsilon = f(\alpha, \beta, \gamma, \delta, M(1,1))$
2	$\alpha = f(\beta, \gamma, \delta, M(3,1))$	$\varepsilon = f(\alpha, \beta, \gamma, \delta, M(1,2))$
3	$\alpha = f(\beta, \gamma, \delta, M(3,1))$	$\varepsilon = f(\beta, \gamma, \delta, M(1,3))$
4	$\alpha = f(\beta, \gamma, \delta, M(3,1))$	$\varepsilon = f(\alpha, \beta, \gamma, \delta, M(2,1))$
5	$\alpha = f(\beta, \gamma, \delta, M(3,1))$	$\varepsilon = f(\alpha, \beta, \gamma, \delta, M(2,2))$
6	$\alpha = f(\beta, \gamma, \delta, M(3,1))$	$\varepsilon = f(\beta, \gamma, \delta, M(2,3))$
7	$\alpha = f(\beta, \gamma, \delta, M(3,2))$	$\varepsilon = f(\alpha, \beta, \gamma, \delta, M(1,1))$
8	$\alpha = f(\beta, \gamma, \delta, M(3,2))$	$\varepsilon = f(\alpha, \beta, \gamma, \delta, M(1,2))$
9	$\alpha = f(\beta, \gamma, \delta, M(3,2))$	$\varepsilon = f(\beta, \gamma, \delta, M(1,3))$
10	$\alpha = f(\beta, \gamma, \delta, M(3,2))$	$\varepsilon = f(\alpha, \beta, \gamma, \delta, M(2,1))$
11	$\alpha = f(\beta, \gamma, \delta, M(3,2))$	$\varepsilon = f(\alpha, \beta, \gamma, \delta, M(2,2))$

12	$\alpha = f(\beta, \gamma, \delta, M(3,2))$	$\varepsilon = f(\beta, \gamma, \delta, M(2,3))$
13	$\varepsilon = f(\beta, \gamma, \delta, M(1,3))$	$\alpha = f(\beta, \gamma, \delta, \varepsilon, M(1,1))$
14	$\varepsilon = f(\beta, \gamma, \delta, M(1,3))$	$\alpha = f(\beta, \gamma, \delta, \varepsilon, M(1,2))$
15	$\varepsilon = f(\beta, \gamma, \delta, M(1,3))$	$\alpha = f(\beta, \gamma, \delta, \varepsilon, M(2,1))$
16	$\varepsilon = f(\beta, \gamma, \delta, M(1,3))$	$\alpha = f(\beta, \gamma, \delta, \varepsilon, M(2,2))$
17	$\varepsilon = f(\beta, \gamma, \delta, M(1,3))$	$\alpha = f(\beta, \gamma, \delta, \varepsilon, M(3,1))$
18	$\varepsilon = f(\beta, \gamma, \delta, M(1,3))$	$\alpha = f(\beta, \gamma, \delta, M(3,2))$
19	$\varepsilon = f(\beta, \gamma, \delta, M(2,1))$	$\alpha = f(\beta, \gamma, \delta, M(1,1))$
20	$\varepsilon = f(\beta, \gamma, \delta, M(2,3))$	$\alpha = f(\beta, \gamma, \delta, \varepsilon, M(1,2))$
21	$\varepsilon = f(\beta, \gamma, \delta, M(2,3))$	$\alpha = f(\beta, \gamma, \delta, \varepsilon, M(2,1))$
22	$\varepsilon = f(\beta, \gamma, \delta, M(2,3))$	$\alpha = f(\beta, \gamma, \delta, \varepsilon, M(2,2))$
23	$\varepsilon = f(\beta, \gamma, \delta, M(2,3))$	$\alpha = f(\beta, \gamma, \delta, M(3,1))$
24	$\varepsilon = f(\beta, \gamma, \delta, M(2,3))$	$\alpha = f(\beta, \gamma, \delta, M(3,2))$

6.1.4.3 Singularity Handling

The procedure exposed before is able to provide a valid configuration for the wide majority of input Euler angles. However, as it was previously introduced, in some cases not every combination of three entries can deliver a solution. Therefore, it is necessary to recognize these unfavorable scenarios in order to handle them and avoid spurious solutions. A very simple example of such adverse scheme is any case in which one or several entries presents a value of 1, for example:

$$M_d = \begin{pmatrix} M_d(1,1) & M_d(1,2) & M_d(1,3) \\ M_d(2,1) & M_d(2,2) & M_d(2,3) \\ M_d(3,1) & M_d(3,2) & M_d(3,3) \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad \text{Eq. (6-21)}$$

In this case, if one uses the entry $M_d(3,1)$ for calculating α , it is not possible to use the entry $M_d(1,1)$ or $M_d(2,1)$ for calculating ε afterwards. The reason behind this is related to the properties of the rotation matrices: the modulus of each column must equal the unit. Therefore, if the condition of $M_d(3,1) = 1$ has been applied then $M_d(1,1) = M_d(2,1) = 0$ by definition. Hence, using any of these conditions does not provide any new information to the system. Nevertheless, there are many remaining possibilities that are valid for solving the problem. For example, ε can be calculated using $M_d(2,2)$ (Possibility 5 in Table 6-1). Therefore, although some of the 24 possibilities can

become meaningless for certain configurations, several will remain valid and it will always be possible to find the solutions.

Besides this problematic, there is one extra condition which requires a deviation from the nominal procedure. This is the case in which $M_d(3,3) = 1$. In such cases $M_d(3,1) = M_d(3,2) = M_d(1,3) = M_d(2,3) = 0$ by definition, which makes all these entries unsuitable for determining either α or ε . In these specific cases, there is a singularity that reduces the degrees of freedom of the robot and the value of either α or ε can be determined freely. This can be deduced from the results of 6.1.4.1, where it was demonstrated that for the case of $M_d(3,3) = 1$ γ must be zero, this can be interpreted as the loss of one degree of freedom into the system. Indeed, it can be demonstrated that $M_d(3,1) = M_d(3,2) = M_d(1,3) = M_d(2,3) = 0$ regardless of the value that we choose for α or ε . For example, if we consider the entry $M_d(3,2)$, with the conditions of $\gamma = 0$, then $\beta = -\delta$, therefore:

$$M_d(3,2) = \cos(-\beta)(\cos(\alpha)\sin(\gamma = 0) + \sin(\alpha)\sin(\beta)) + \cos(\beta)\sin(\alpha)\sin(-\beta) = 0 \quad \text{Eq. (6-22)}$$

This equality holds regardless of the value of α . The same can be demonstrated for the other three lateral entries. Therefore, for the case of $M_d(3,3) = 1$ the values of β , γ and δ are determined as presented in section 6.1.4.1 and afterwards, either either α or ε can be chosen randomly and finally ε or α are calculated using the central entries of the horizontal plane in M_d .

6.1.5 Solution Representation

With the exposed methodology it is now possible to calculate the manifold of the inverse Kinematics for a given Euler angles. These manifolds are five-dimensional hypersurfaces, whose visualization is not intuitive. In order to plot them, the three spatial dimensions have been used for three of the angles (Azimuth, Elevation and Polar) and the colorcode represents the two Nick angles. Fig. 6-5 and Fig. 6-6 illustrate the hypergeometries that result for the configurations that can lead to the coordinates $\varphi_d = -30^\circ$, $\theta_d = 0^\circ$ and $\psi_d = 0^\circ$. As it can be seen, the solutions presents several regions, all of which lead to the same attitude as it can be seen explicitly in Fig. 6-6. This sort of graphs provides a comprehensive overview of the solution to the inverse kinematics and several insights can be extracted from them. For example, it can be seen that the solution manifold presents several unconnected regions in a similar way as it was identified in Lück et.al 1999 [16]. It can be seen that the manifold's borders correspond to limit values of at least two of the constrained joints. The color gradient can of how strongly. For example, for the region of solutions located at a negative polar and positive azimuth, it can be seen that having extra freedom in the A-Nick joint would not provide a significant increase in the manifold's hypervolume. However, if one focuses in the area with a positive polar and a negative azimuth it can be seen that the color gradient is much lower, which indicates that additional freedom around the A-Nick would be highly beneficial. This rather abstract deductions can be verified in a more intuitive way by comparing the point F and B from Fig. 6-6.

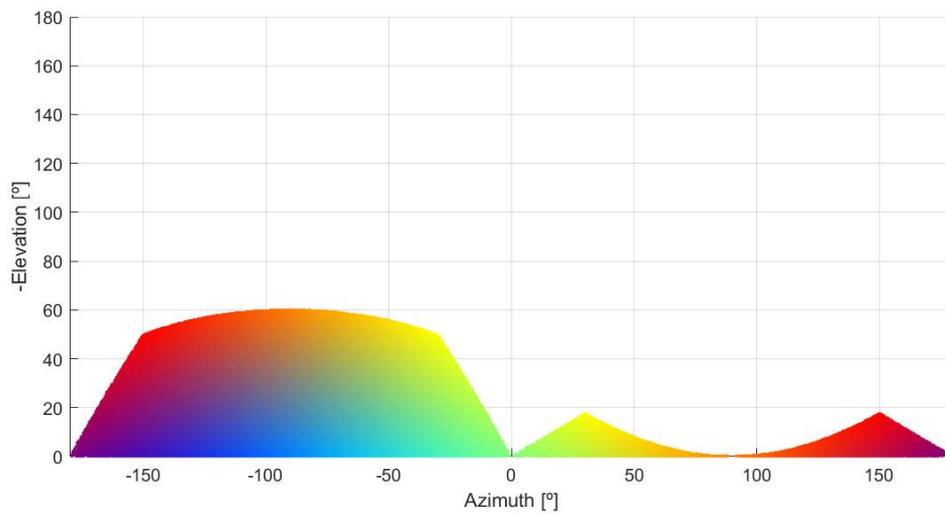
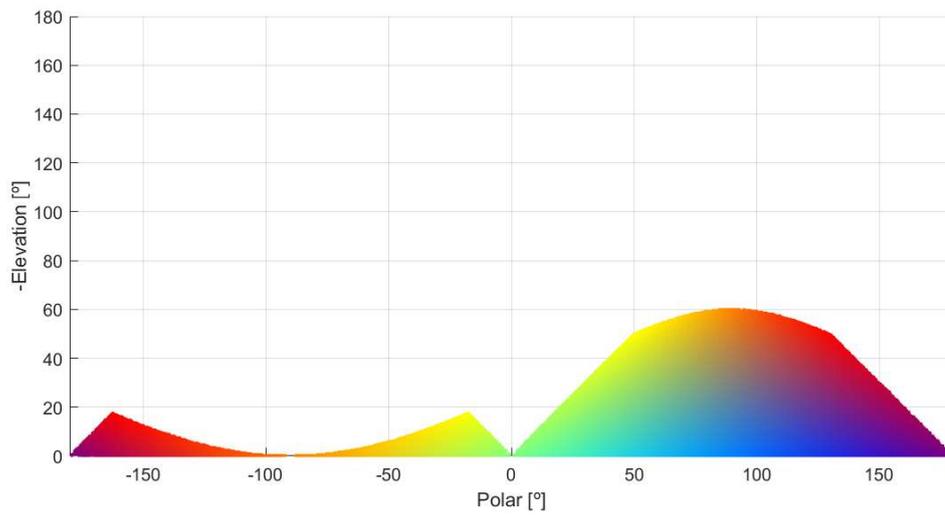
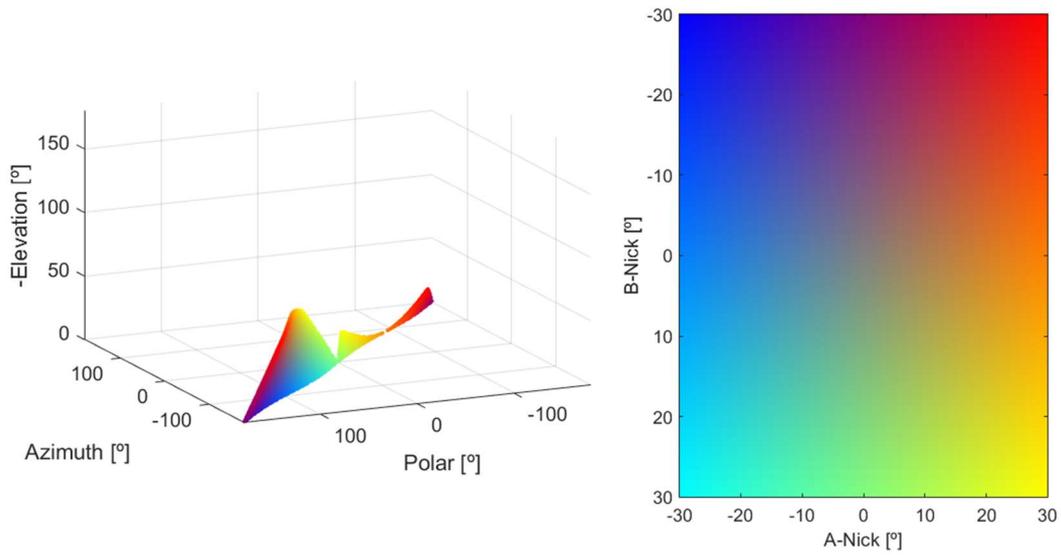


Fig. 6-5: Inverse Kinematics Solution Example (I)

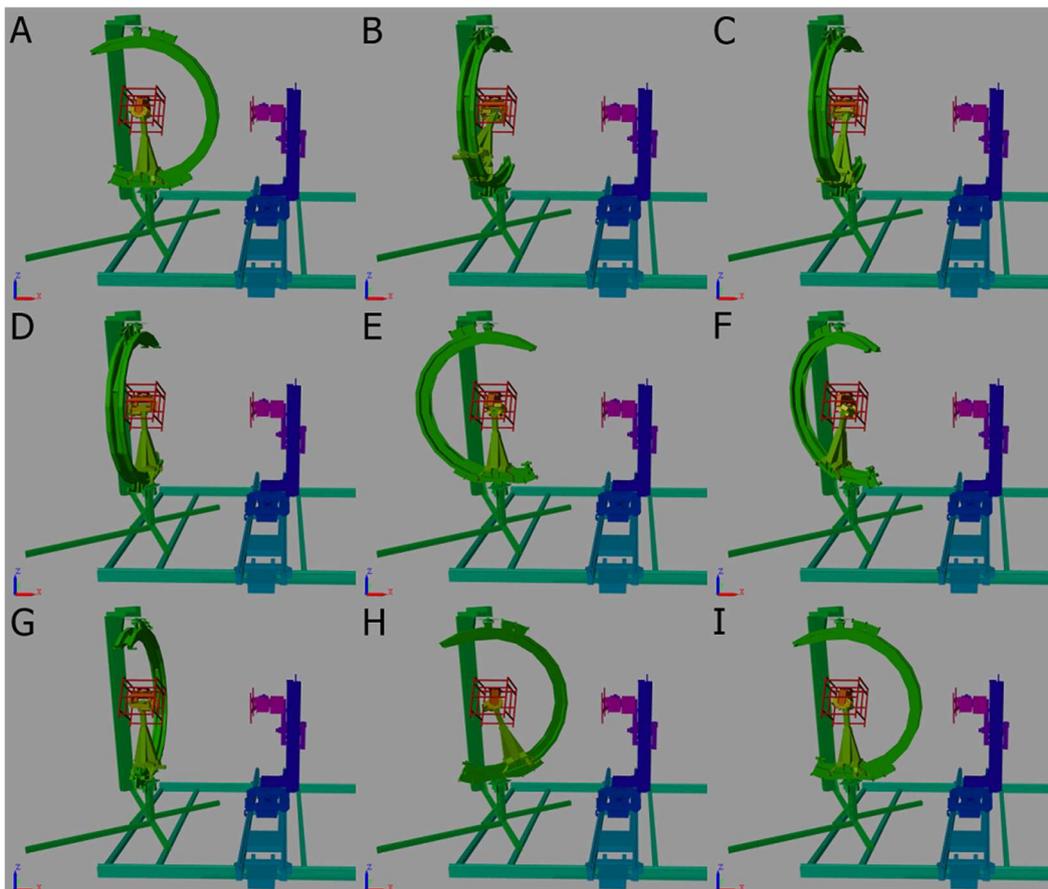
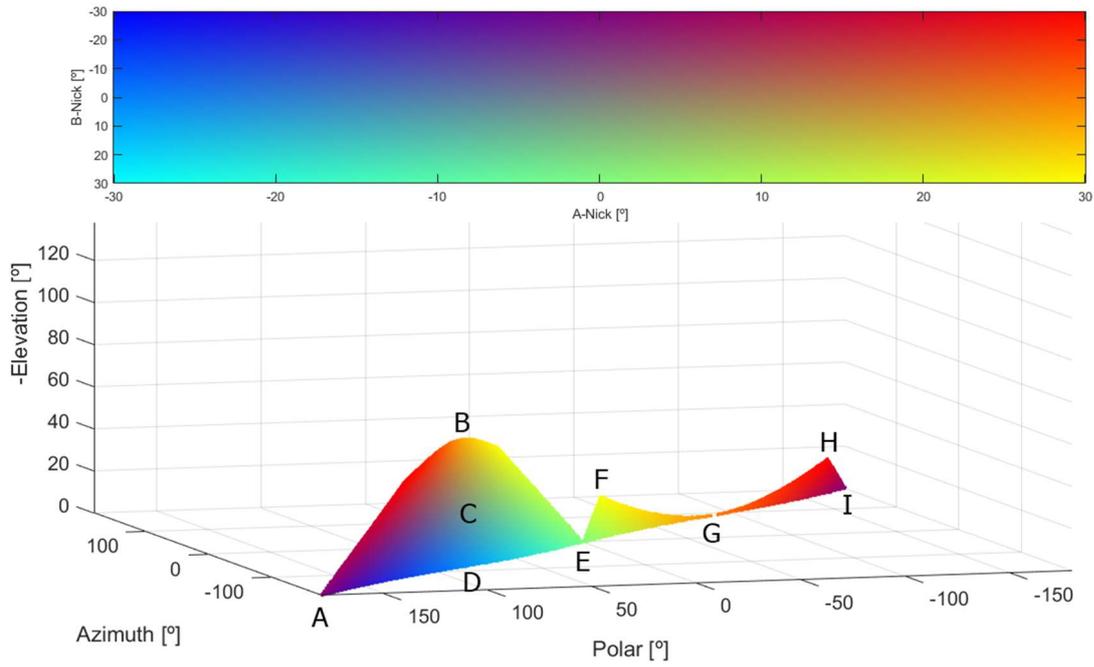


Fig. 6-6: Inverse Kinematics Solution Example (II)

6.1.6 Algorithm Testing

In order to proof the robustness of the algorithm it is necessary to test the behavior of its implementation. The algorithm has been programmed in a Matlab script which receives as input the desired Euler angles and number of points of the manifold and provides as output an $n \times 5$ Array containing n points of the solution manifold. The critical parameter that it is evaluated in the tests is the Time per Solution (TPS) which refers to the average time needed for finding a point of the manifold for a given Euler Angles. This variable has been statistically evaluated with the random and singularity tests. Although these results are strongly dependent on the machine capabilities, they enable the detection of problems when the TPS exceeds its average value. Furthermore, they provide an orientation of the time that the algorithm will require when it is implemented in the embedded system. This will be a crucial aspect when addressing aspects such as delay and latency in practical applications of the RACOON-Lab.

6.1.6.1 Random Test

The random test evaluates the TPS for random Euler angles. A total 100000 random Euler angles have been tested. In order to calculate the TPS of each of them, 1000 points have been requested and the total execution time has been measured. The average TPS has been determined to be $356.3 \mu\text{s}$ with a standard deviation of $45.4 \mu\text{s}$. The experimentally obtained Probability Density Function (PDF) can be visualized in Fig. 6-7.

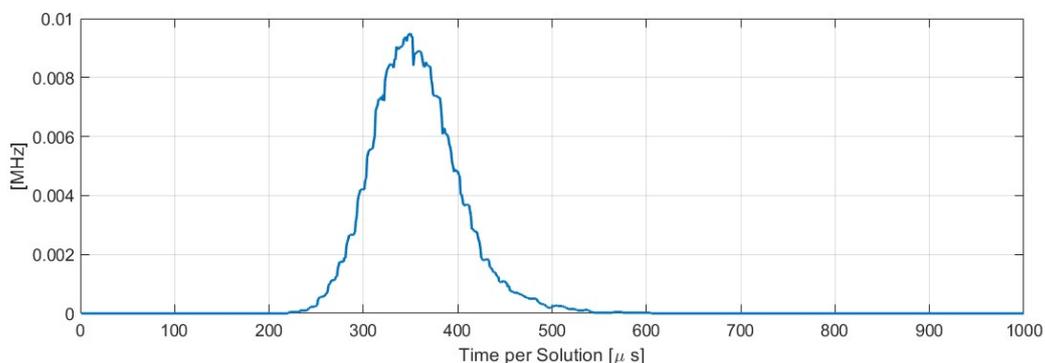


Fig. 6-7: Probability Density Function of the Time per Solution in the Random Test for the Analytical Solution of the Inverse Kinematics

6.1.6.2 Singularity Test

The singularity test follows the same scheme of the random test with the particularity that in this case Euler angles are always multiples of 15° , which provides a total of $24^3 = 13824$ possibilities. The goal of this is intending to trigger issues that the algorithm may present when approaching to singular coordinates, which are much more unlikely to take place if random Euler angles are considered. In the case of this test, the average TPS was determined to be $272.6 \mu\text{s}$ and the standard deviation $71.7 \mu\text{s}$. The associated PDF to this test is displayed in Fig. 6-8. These results indicate that the performance in singular scenarios is slightly better than that of normal cases.

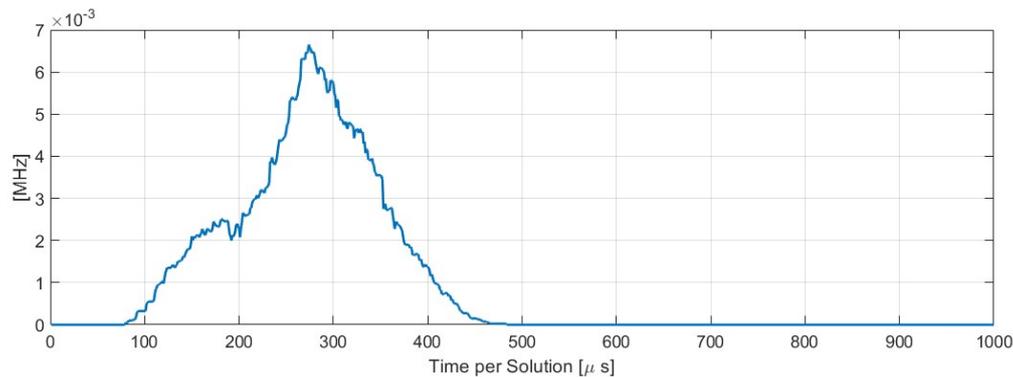


Fig. 6-8: Probability Density Function of the Time per Solution in the Singularity Test for the Analytical Solution of the Inverse Kinematics

The reason behind this performance improvement is likely to be related to the fact that the singularity handling strategy presented in 6.1.4.3 does not require any additional computational effort. Indeed, for the cases when the Z axes are aligned, the singularity reduces the problem's complexity as $\gamma = 0^\circ$ by definition and it is only necessary to determine two angles instead of the usual three because the system has only four degrees of freedom. In order to visualize this, the performance of the algorithm was studied as a function of how far it is from singularities. This proximity is assessed with the angle $\chi = \text{asin}(M_d(3,3))$ which is the angle between the end effector's vertical plane and the horizontal plane in the inertial frame. Therefore, the cases of $\chi = \pm 90^\circ$ correspond to the scenario in which the inertial and end effector's vertical axes are aligned. Fig. 6-9 illustrates the relationship between this angle and the algorithm's performance and it verifies the hypothesis stated before.

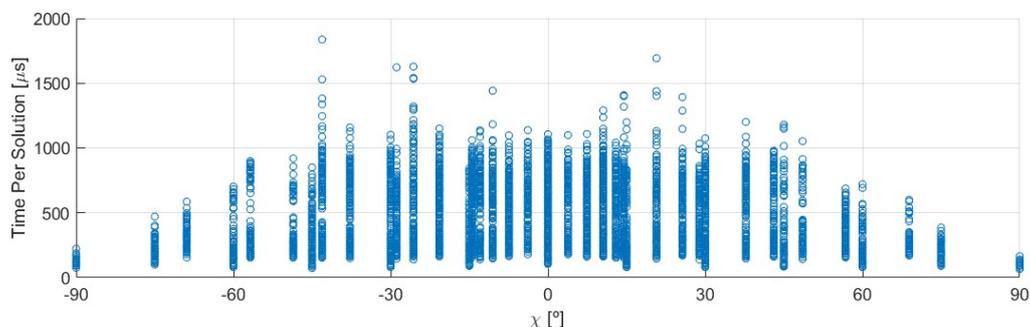


Fig. 6-9. Time per Solution of the Algorithm as a Function of the Angle between the End Effector's Vertical axis and the Horizontal plane.

6.2 Solution in the Velocity Domain

Due to the complexity of the analytical approach, the inverse kinematics is often solved in the velocity domain [4, 27]. This is often done by obtaining the Jacobian which relates configuration velocities to space coordinates velocities. However, the global manifold of solutions is not usually determined. In this section both the Jacobian and the manifold of solutions in the velocity domain will be obtained in order to enable a better assessment of the analytical solution of the point problem, which was addressed previously.

This section will present a similar outline as the former one, with problem statement, solution description and examples finishing with the algorithm's testing.

6.2.1 Problem Statement

The inverse kinematics in the velocity domain aims to find the manifold of configuration velocities that provide a given coordinates' time derivatives for the current configuration q . Therefore, For the case of the RACOON-Lab's Rotator this problem can be stated as follows:

Given the desired Euler angles' rates $\dot{\Xi}_d = (\dot{\varphi}_d, \dot{\theta}_d, \dot{\psi}_d)$, and the current robot configuration $q = (\alpha, \beta, \gamma, \delta, \varepsilon)^T$ associated to the Euler angles $\Xi = (\varphi, \theta, \psi)$, which provide the desired rotation matrix time derivative $\dot{M}_d(\Xi, \dot{\Xi}_d)$:

Find the manifold σ , consistent of all configurations $\dot{q} = (\dot{\alpha}, \dot{\beta}, \dot{\gamma}, \dot{\delta}, \dot{\varepsilon})^T$ that fullfill the following equality:

$$\dot{q} \in \sigma(q, \dot{\Xi}_d) \leftrightarrow \{\dot{M}_{IER}(q, \dot{q}) = \dot{M}_d(\Xi, \dot{\Xi}_d)\} \wedge \{|\dot{q}_i| \leq |\dot{q}_{i,max}| \forall i\} \quad \text{Eq. (6-23)}$$

With \dot{M}_{IER} being the time derivative of the rotation matrix M_{IER} from the Rotator's end effector to the inertial frame and $|\dot{q}_{i,max}|$ denotes the maximum allowable angular velocity of the i^{th} joint.

6.2.2 Strategy Overview

Although the problem statement in the velocity domain may lead to the conclusion that it is more demanding compared to the previous problem, this is not the case. The reason for this is the linear nature of the resulting velocity equations, contrary to their geometrical counterpart.

In order to obtain the solution to the stated problem, the angular rates will be calculated in the End effector's frame using the Rotation tensor. This will provide the Jacobian tensor of body to configuration rates. Such tensor can be easily transformed into Euler angles to configuration rates with the use of a rotation matrix. This will provide a linear system of equations, whose solution is the inverse kinematics manifold in the velocity domain.

6.2.3 Problem Solution

It is possible to obtain the values of the angular velocities in the inertial frame by computing the rotation tensor S which can be obtained as the product of global rotation matrix and its time derivative:

$$S = \begin{pmatrix} 0 & -r_E & q_E \\ r_E & 0 & -p_E \\ -q_E & p_E & 0 \end{pmatrix} = \dot{M}_{IER} M_{ERI} \quad \text{Eq. (6-24)}$$

Where p_E, q_E and r_E are the angular velocities in the End effector's body frame. This tensor was computed for the RACOON Lab's Rotator. From this result, each non-diagonal entry provides the expression of the angular velocities in the body frame as a function of the time derivative of the robot's configuration. Since this dependency is

linear for a given configuration, it is suitable to express the angular velocities with the following compact expression:

$$\begin{pmatrix} p_E \\ q_E \\ r_E \end{pmatrix} = J_{\dot{E}q}(q) \cdot \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \\ \dot{\delta} \\ \dot{\epsilon} \end{pmatrix} \quad \text{Eq. (6-25)}$$

Where $J_{\dot{E}q}$ is the 3x5 Jacobian tensor, which contains the sensitivity coefficients of the angular rates in the end effector's frame with respect to each of the configuration's parameters. Each of this matrix's entries depends on the current robot's configuration q . The reader can find the exact expression for each of these terms for the RACOON Lab's rotator in the Appendix B, Jacobian Tensor.

Finally, it is necessary to convert the angular velocities in the body frame to the Euler angles' rates. This can be done easily with the attitude conversion matrix, which leads to the following formulation:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{pmatrix} J_{\dot{E}q} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \\ \dot{\delta} \\ \dot{\epsilon} \end{pmatrix} = J_{\dot{\Xi}q} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \\ \dot{\delta} \\ \dot{\epsilon} \end{pmatrix} \quad \text{Eq. (6-26)}$$

At this stage, there are three equations which relate the time derivative of the robot's configuration to the global angular velocities of the robot. This provides once again an underdetermined system. Therefore, the solution manifold is an infinite set of combinations of the angular velocities of the actuators. Unlike the point problem, the analytical solution does not entail any complex derivation as no sinusoidal function applies to any output. Indeed, the procedure is a mere linear system of equations.

A feasible procedure for solving linear systems of this sort can be based on parametrization [28]. The procedure for obtaining the whole manifold of solutions departing from Eq. (6-25) can be summarized as follows:

1. Define two suitable pairs from the five angular velocities of the actuators
2. Use these velocities from their minimum to their maximum value in order to obtain the remaining three velocities combining the three equalities provided in Eq. (6-25).
3. Repeat the previous steps for another suitable pairs until all the 10 possibilities have been considered.

The third step is in order to ensure the exploration of all the solution space under fulfillment of the inequalities constraint ($|\dot{q}_i| \leq |\dot{q}_{i,max}| \forall i$). In this context, the expression, suitable refers to the condition in which the remaining Jacobian 3x3 after the rows have been subtracted presents a different than zero. This issue will be addressed with more detail in the following two sections.

6.2.4 Singularity Handling

The algorithm visits every possible combination of two configuration angles in order to determine points of the solution manifold. If we use $J_{\dot{E}q}^{\circ}$ to denote the 3x3 resulting Jacobian after removing two rows, the singularity risk can be assessed with its determinant i.e. $\det(J_{\dot{E}q}^{\circ})$. If choosing two particular configuration angles q_a and q_b as parameters leads to $\det(J_{\dot{E}q}^{\circ}) = 0$ this implies that no valid points can be obtained from parametrizing q_a and q_b . Therefore, if this determinant presents a value of zero, then another parameters combinations must be explored for parametrization. Nevertheless, from a practical standpoint it is advisable to refrain from using determinants whose value is close to zero, as it implies that the change in Euler rates must be achieved by very large angular velocities, which is highly undesirable. For the implementation of the present project a criteria of $|\det(J_{\dot{E}q}^{\circ})| > 10^{-3}$ has been used for considering the configuration parameters as valid.

6.2.5 Solution Example

In order to illustrate the previously described process, an example of the solution for the analytical inverse velocity kinematics will be exposed. Let consider the configuration in which all the actuators are set to zero with exception of the elevation angle which presents a value of -45° . The Jacobian can be computed using the formulas of the annex as:

$$J_{\dot{E}q} \begin{pmatrix} \alpha = 0^{\circ} \\ \beta = -45^{\circ} \\ \gamma = 0^{\circ} \\ \delta = 0^{\circ} \\ \varepsilon = 0^{\circ} \end{pmatrix} = \begin{pmatrix} -2^{-\frac{1}{2}} & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 2^{-\frac{1}{2}} & 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Eq. (6-27)}$$

With this Jacobian it is now possible to compute the solutions which enable the robot to achieve an angular velocity of $\Omega_E = (p_E \ q_E \ r_E)$. It is necessary to remove two columns in a way that the remaining matrix has a rank of 3. For example, if the first and the last columns are removed, the remaining system has rank of 2. Removing the fourth and the fifth column columns however leads to a matrix with a determinant different than zero. The columns that are removed are identified with the angular velocities that will be constrained and the remaining columns are associated to the angular velocities that will be calculated as a function of the constrained ones. In a general way, it is possible to express the resulting system using $J_{\dot{E}q}^{\Delta}$ to denote the 3x2 matrix removed from the original Jacobian and $J_{\dot{E}q}^{\circ}$ the remaining 3x3 system as:

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} - J_{\dot{E}q}^{\Delta} \begin{pmatrix} q_a \\ q_b \end{pmatrix} = J_{\dot{E}q}^{\circ} \begin{pmatrix} q_{1,f} \\ q_{2,f} \\ q_{3,f} \end{pmatrix} \rightarrow \begin{pmatrix} q_{1,f} \\ q_{2,f} \\ q_{3,f} \end{pmatrix} \rightarrow \begin{pmatrix} q_{1,f} \\ q_{2,f} \\ q_{3,f} \end{pmatrix} = (J_{\dot{E}q}^{\circ})^{-1} \left(\begin{pmatrix} p \\ q \\ r \end{pmatrix} - J_{\dot{E}q}^{\Delta} \begin{pmatrix} q_{1,c} \\ q_{2,c} \end{pmatrix} \right) \quad \text{Eq. (6-28)}$$

Where $q_{i,c}$ denotes the i^{th} parametrized angle and $q_{j,f}$ denotes the j^{th} non-parametrized angle. In the proposed example, this procedure leads to the following expression:

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\delta} \\ \dot{\varepsilon} \end{pmatrix} = \begin{pmatrix} -2^{-\frac{1}{2}} & 0 & 1 \\ 0 & 1 & 0 \\ 2^{-\frac{1}{2}} & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{pmatrix} \rightarrow \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{pmatrix} = \begin{pmatrix} \sqrt{2}(r - \varepsilon) \\ q - \delta \\ p - r + \varepsilon \end{pmatrix} \quad \text{Eq. (6-29)}$$

As it can be seen, the degrees of freedom which remained free can be easily obtained with a linear application dependent on the desired angular velocities and the two degrees of freedom which were initially constrained. Fig. 6-10 illustrates all the possible solutions with B-Nick and polar angles ranging from -1 to 1 deg/s for obtaining an angular velocity of $(p \ q \ r) = (1 \ 0 \ 0) \text{ deg/s}$

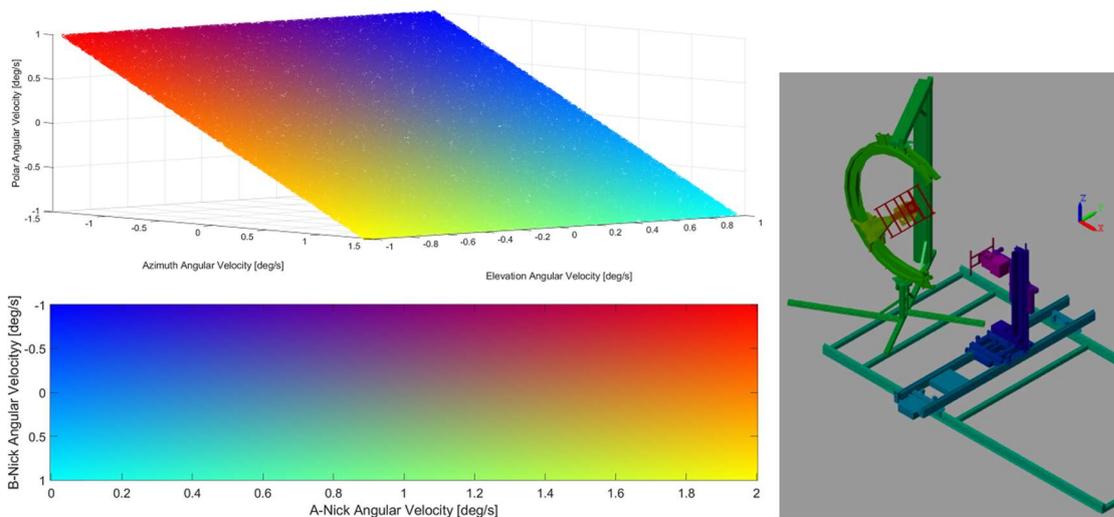


Fig. 6-10: Solution Example of the Inverse Kinematics in the Velocity Domain

Generally, one wants that the angular velocities are as small as possible. This condition can be very easily implemented for the constrained actuators as they are input manually. However, there is no control at all regarding the angular velocities which were set to free. Therefore, when the determinant of $J_{E\dot{q}}^0$ tends to zero the angular velocities of the remaining actuators will tend to infinite. This issue can cause significant problems even if the matrix is far from being singular. For example, angular velocities with an absolute value 10 or 20 times higher than the desired ones are already highly undesirable. Another significant shortcoming of this method is that it is not possible to identify and avoid locking situations. For example, if $\delta \approx 30^\circ = \delta_{max}$ then every $\dot{\delta} > 0$ is highly undesirable, but that may be impossible for some configurations. In spite of these inconveniences, this approach is very robust for the computation of instantaneous angular velocities.

6.2.6 Algorithm Testing

As in the analytical solution, the implementation of the algorithm in the velocity domain has been tested with both random and singularity tests. It is important to remark, that in this case the element which influence in the performance is the current configuration q instead of the desired values for the kinematics (in this case Euler angles' rates). The reason for this is that the key points of the presented approach is the determination of the Jacobian tensor, which merely depends on the current configuration. Therefore, in this case, the random inputs were for the configuration's angles.

The results of these tests are displayed in Fig. 6-11 and Fig. 6-12. In this case, the mean for the random test was $84.3 \mu\text{s}$ with a standard deviation of $55.5 \mu\text{s}$, whilst the singularity test exhibited a mean performance of $77 \mu\text{s}$ with a standard deviation of $27.4 \mu\text{s}$. Both results are much faster than the analytical solution, which is logical due to the lower complexity of the involved functions. Furthermore, the singularity test provides a slightly better performance than the random one, which is in principle unexpected. A possible explanation could be that the singular combinations are discarded fast causing virtually no lost in time, whilst the remaining combinations present sounder determinants since the linear dependencies are concentrated in the unfavourable combinations that are already discarded.

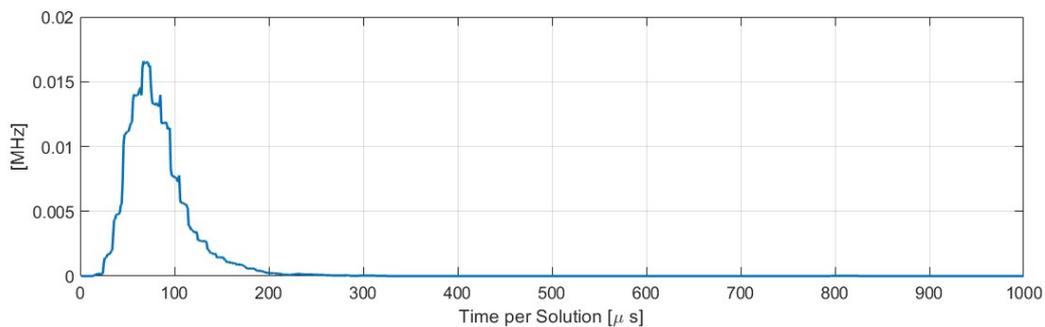


Fig. 6-11: Probability Density Function of the Time per Solution in the Random Test for the Solution of the Inverse Kinematics in the Velocity Domain

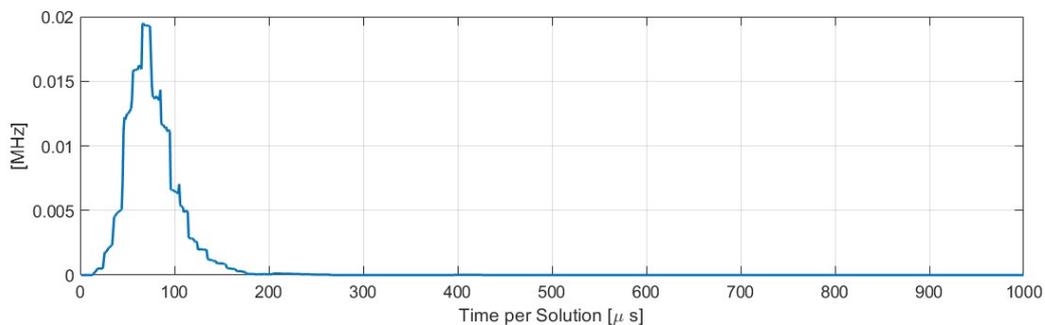


Fig. 6-12: Probability Density Function of the Time per Solution in the Random Test for the Solution of the Inverse Kinematics in the Velocity Domain

7 Trajectory Planning

Once the inverse kinematics for single points has been explained it is possible to address the trajectory problem. This means, that the main goal is the determination of the robot's optimal configuration in the facility over series of attitude way-points instead of single ones.

Since the chaser has only three unconstrained degrees of freedom there is a finite number of solutions for each configuration (Generally 2, with a shift of 180° in roll). Therefore, in this case the transition from one position to another is unequivocal. This is not the case for the Rotator. Unlike the Chaser, this element is over actuated and has infinite solutions for a given desired position. Therefore, it is necessary to implement policies which indicate which is the most suitable sequence of configurations for a given desired trajectory.

In this chapter, algorithms based on the analytical solution described in the section 6.1 will be presented. In order to enable a comparison with conventional methods, the trajectory problem will be initially addressed using some of the approaches presented in the state-of-the-art chapter.

7.1 Definitions

Before jumping into the description of trajectory algorithms, it is necessary to enunciate several basic concepts, that will be used as assumptions repeatedly. These notions will provide the necessary background for the definition and assessment of the trajectory algorithms.

7.1.1 Trajectory Problem Statement

The statement of the trajectory problem for this system presents several particularities with respect to the ordinary one. Usually, what is wanted is that the robot traverses a series of spatial way points as fast as possible. This is meaningful when following the trajectory itself is the main goal of the project. In this case, however, performing a given trajectory is a mean for simulating experimentally on-orbit scenarios. Therefore, in the ideal case, it would be necessary that the robot reaches every position at a given timestamp. This would merely imply to make spatial temporal way points instead of spatial way points. However, the angular velocities of the actuators are in the order of 1 deg/s, whilst the bodies that want to be represented can spin with angular velocities up to 100 deg/s. Therefore, when the body, whose trajectory wants to be recreated spins with angular rates higher than 1 deg/s it is not realistic to target an exact simulation of the motion. In such cases, what is desired is to exactly represent the motion with a given slowdown rate, which should be as small as possible. With all these considerations, it is possible to state the trajectory problem for the current application:

Given a finite set of n desired attitude way points:

$$M_d(t) = M_d(t_0 = 0), M_d(t_1), M_d(t_2), \dots, M_d(t_k), \dots, M_d(t_n) \quad \text{Eq. (7-1)}$$

Find a set of n configurations:

$$q(\tilde{t}) = q(\tilde{t}_0 = 0), q(\tilde{t}_1), q(\tilde{t}_2), \dots, q(\tilde{t}_k), \dots, q(\tilde{t}_n) \quad \text{Eq. (7-2)}$$

Such that:

$$\left\{ M(q(t_k)) = M_d(t_k) \wedge \frac{\tilde{t}_k - \tilde{t}_{k-1}}{t_k - t_{k-1}} = \eta_s \leq 1 \right\} \forall k \in [1, n] \quad \text{Eq. (7-3)}$$

With a trajectory being optimal if $\eta_s = 1$. The parameter η_s is the system's efficiency. It represents how much seconds are required to execute a given trajectory for each second of the actual trajectory. This parameter is dependent on both the facility's capabilities and the trajectory's suitability.

7.1.2 Trajectory Efficiency

In every engineering context the definition of efficiency coefficients must be able to represent the limitations in performance of the considered system from a practical standpoint. In this case, the maximum angular velocities of the facility are limited and increasing these speeds is out of the range of the present project. The current target is however to make the most with these constrains. At this point is where the trajectory's efficiency comes into play. There is an infinite set of possible trajectories which can fulfill the desired sequence of desired attitudes. However, some of the possible trajectories may take excessively long considering the limited angular velocity of the actuators. This can greatly impede the execution of experimental simulations. An optimal trajectory should avoid unnecessary motions in order to maximize the change in attitude per unit of control effort. In other words, the differential motions of the configuration should be as large as possible when compared to the achieved differential changes in coordinates.

These intuitive ideas shall be presented in a mathematical way in order to enable an objective assessment of a given trajectory's suitability. Let consider the process of an evolving trajectory where the Euler angles are constantly changing. The differential attitude change can be defined as:

$$d\Xi = (d\phi \quad d\theta \quad d\psi)^T \quad \text{Eq. (7-4)}$$

This change in the attitude is achieved by a differential change in the robot's configuration, which has as many dimensions as the number of degrees of freedom of the robot:

$$dq = (d\alpha \quad d\beta \quad d\gamma \quad d\delta \quad d\varepsilon)^T \quad \text{Eq. (7-5)}$$

Combining these two elements, it is possible to build a compact Jacobian tensor which represents the derivative of the Euler angles with respect to the configuration:

$$\frac{\partial \Xi}{\partial q} = \begin{pmatrix} \frac{\partial \phi}{\partial \alpha} & \frac{\partial \phi}{\partial \beta} & \frac{\partial \phi}{\partial \gamma} & \frac{\partial \phi}{\partial \delta} & \frac{\partial \phi}{\partial \varepsilon} \\ \frac{\partial \theta}{\partial \alpha} & \frac{\partial \theta}{\partial \beta} & \frac{\partial \theta}{\partial \gamma} & \frac{\partial \theta}{\partial \delta} & \frac{\partial \theta}{\partial \varepsilon} \\ \frac{\partial \psi}{\partial \alpha} & \frac{\partial \psi}{\partial \beta} & \frac{\partial \psi}{\partial \gamma} & \frac{\partial \psi}{\partial \delta} & \frac{\partial \psi}{\partial \varepsilon} \end{pmatrix} \quad \text{Eq. (7-6)}$$

This tensor provides a general overview of the relationship between the changes in the Euler angles with respect to the robot's configuration. However, it lacks a direct practical interpretation from the point of view of performance. In order to assess the differential change in attitude achieved, the following expression is more suitable:

$$d\mu = d\|\Xi\| = d\sqrt{\phi^2 + \theta^2 + \psi^2} \quad \text{Eq. (7-7)}$$

Assuming, that all the joints present the same maximum angular velocity, the element which limits the time of the trajectory's execution is the one which requires a higher change. With this consideration, it is possible to define the differential control effort as:

$$d\kappa = \max_i\{d|q_i|\} = \max\{d|\alpha|, d|\beta|, d|\gamma|, d|\delta|, d|\varepsilon|\} \quad \text{Eq. (7-8)}$$

And the efficiency for a given maneuver is defined as the ratio between these two terms, which can be obtained from the angular velocities since the differential time cancels out:

$$\eta_M(\dot{\Xi}, \dot{q}) = \frac{d\mu}{d\kappa} = \frac{\|\dot{\Xi}\|}{\max_i\{|\dot{q}_i|\}} = \frac{\sqrt{\dot{\phi}^2 + \dot{\theta}^2 + \dot{\psi}^2}}{\max\{|\dot{\alpha}|, |\dot{\beta}|, |\dot{\gamma}|, |\dot{\delta}|, |\dot{\varepsilon}|\}} \quad \text{Eq. (7-9)}$$

For n waypoints there are $n - 1$ maneuvers that must be performed. The maneuver which requires a higher control effort is the one which limits the continuous execution of the trajectory. Therefore, the efficiency of a given trajectory is defined as the minimum efficiency across its maneuvers in time:

$$\eta_T = \underset{t}{\operatorname{argmin}}\{\eta_M(t)\} \quad \text{Eq. (7-10)}$$

Under the assumption that the total limitations that a given facility experiences are given by the trajectory's efficiency and by its maximum spinning capabilities, it is possible to estimate the facility's efficiency as:

$$\eta_S = \eta_T \cdot \eta_F \quad \text{Eq. (7-11)}$$

Where η_F is the ratio between the maximum angular velocity achievable by the actuators and the spinning rate of the spacecraft whose shall be simulated. This term accounts for the limitations introduced by the facility. These definitions for efficiency are very convenient due to its straightforward interpretation. With these terms it is possible to identify the challenges that the system undergoes in order to accurately represent a motion in the minimum time. For example, considering the scenario in which one wants to recreate 1 second of the trajectory of a body which spins with a rate of 10 degrees per second with actuators capable of spinning at 1 degree per second. In this case, if a given trajectory is able of recreating this motion in 10 seconds, then $\eta_T = 1$ and $\eta_S = 0.1$, therefore $\eta_F = 0.1$. These parameters tell that the limiting factor in the execution time of the motion is the actuator's maximum spinning rate, which is completely meaningful from a physical perspective.

It is important to note, that for over actuated robots the trajectory's efficiency can be higher than a 100%. This can be illustrated very easily in the case of the RACOON Lab. For example, if the desired trajectory is a rotation around the Z axis, it is possible

to achieve this movement by simultaneously using the azimuth and polar angles. In this example, a change of 1 degree in azimuth can be achieved by a change of a half degree in both azimuth and polar. Whenever $\eta_S \geq 1$, it means that it is possible to represent the desired motion both spatially and temporally. In the opposite case, $\eta_S < 1$, it will be necessary to execute the considered trajectory with a certain slowdown rate. In particular, each actual second of the trajectory will require $1/\eta_S$ seconds of the facility's operation in order to be accurately represented.

7.1.3 Attitude Error

At some points of the present section it will be necessary to assess the error of the attitude for a given configuration. This requires an objective measure of the deviation between the actual and the desired orientation. The attitude error used in this chapter is defined as the second norm of the difference between the actual Euler angles and the desired ones:

$$\epsilon_{\Xi} = \sqrt{(\phi - \phi_d)^2 + (\theta - \theta_d)^2 + (\psi - \psi_d)^2} \quad \text{Eq. (7-12)}$$

7.1.4 Static and Dynamic Trajectories

The trajectory problem can be stated in two different contexts. In the static trajectory problem, the entire desired trajectory history is given as an input. In this situation, it is easier to optimize the trajectory and it is not necessary to account for aspects such as computational time or latency.

Unlike the offline case, in the dynamic problem the new points of the trajectory are computed during the execution of the simulation. This is required for dynamic simulations where the interaction with the chaser or other elements is relevant. In such situations, it is not possible to know the trajectory beforehand.

7.1.5 Rotation Example

In order to provide a practical illustration for the proposed approaches, the same basic rotation movement will be considered with each methodology. Such trajectory shall be challenging from a kinematical standpoint in order to provide relevant information with respect to the algorithm's robustness.

The main challenge of the RACOON-Lab's Rotator is achieving continuous rotations around an axis contained in the XY plane due to the constraints of the elevation angle. The rotations around the Z axis can be regarded as undemanding since the joints involved present no limitations. Therefore, a suitable trajectory for containing demanding motions is a continuous rotation around the X axis with the angle φ varying from 0 to 360 degrees as portrayed in Fig. 7-1.

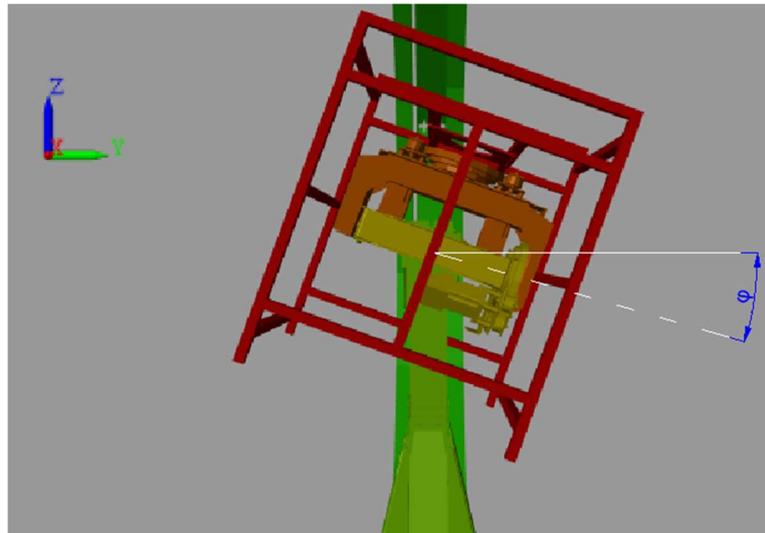


Fig. 7-1: Example Trajectory Used for the Algorithm Testing

This trajectory is a roll rotation and its rotation matrix can be written as a function of φ :

$$M_d = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{pmatrix} \quad \text{Eq. (7-13)}$$

This motion entails the sequence of the most challenging configurations for the rotator as it was introduced in Chapter 4. Due to this, if a given algorithm can traverse this trajectory successfully it is reasonable to assume that it will be able to perform well generally.

7.2 Conventional Algorithms

The example trajectory was tested applying some of the algorithms available for the control of constrained robots presented in the second chapter. In particular, Automatic Supervisory Control and Weighted Least Norm were used. The performance of these algorithms was evaluated using the efficiency definition presented in the previous sections. The obtained results will enable the assessment of the performance achieved by the algorithms based on the analytical solution, which are described in the following section.

7.2.1 Automatic Supervisory Control

The Automatic Supervisory Control algorithm (ASC) is one of the most widespread methodologies for the trajectory planning of constrained robots. This algorithm was tested and tuned for a continuous roll rotation example described in section 7.1.5. Here, the results of implementing this algorithm are presented and assessed.

7.2.1.1 Algorithm Description

The ASC algorithm was applied using the identity Matrix as the weighting Matrix and the performance criterion H as defined by *Zghal et al. 1990* [29]. Therefore, the optimization function for the RACOON-Lab's Rotator can be expressed as:

$$\Phi = \frac{1}{2}(\dot{\alpha}, \dot{\beta}, \dot{\gamma}, \dot{\delta}, \dot{\varepsilon})^T W \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \\ \dot{\delta} \\ \dot{\varepsilon} \end{pmatrix} + \alpha_k \frac{\partial H(q)}{\partial q} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \\ \dot{\delta} \\ \dot{\varepsilon} \end{pmatrix} \quad \text{Eq. (7-14)}$$

With:

$$H(q_i) = \frac{(q_{max} - q_{min})^2}{4(q_{max} - q_i)(q_i - q_{min})} \quad \text{Eq. (7-15)}$$

$$\frac{\partial H}{\partial q} = \frac{(q_{max} - q_{min})^2(2q_i - q_{max} - q_{min})}{4(q_{max} - q)^2(q_i - q_{min})^2} \quad \text{Eq. (7-16)}$$

The performance derivative with respect to the robot's configuration is therefore a row vector, whose length equals five.

The philosophy behind this cost function's formulation is the definition of two tasks that must be fulfilled simultaneously by the robot. The first one, involves the weighting matrix W and refers to minimizing the angular velocities. The second one, contains the performance function and it is related to the avoidance of the joint's limits. The constant α_k is used for assigning different priorities to each of these tasks. As the value of α_k increases, the second tasks' importance grows. For low values of α_k , the robot follows very efficient trajectories that easily reach joint's limits. Conversely, as α_k tends to infinity, the robot will try to remain as far as possible from the joints' limits even if it means to perform very abrupt movements. The optimal value of α_k is the lowest one which is still able to confine the joint's angles within its limits. This value depends on the robot's kinematics, the considered application and the cost function's definition. In the present project it was determined in an iterative way, departing from zero and gradually increasing its value until a physically feasible trajectory is achieved. The intermediate results of this tuning process are displayed in the Appendix C.

The cost function in Eq. (7-14) was optimized numerically for each value of α_k by performing a discretization over the solution's manifold. This was done for each point of the trajectory, which was itself discretized with a step of 0.1 degrees.

7.2.1.2 Results

The minimum value of α_k which proved capable of delivering a trajectory without exceeding the joint's limits was found to be 24 s/rad, which is remarkably high. Such value implies that avoiding the limits must be highly prioritized with respect to the primary task. As a result of this, the trajectory is valid at the expense of resorting eventually to high angular velocities. This sort of behavior can be visualized in Fig. 7-2, where high slopes for polar and azimuth can be observed.

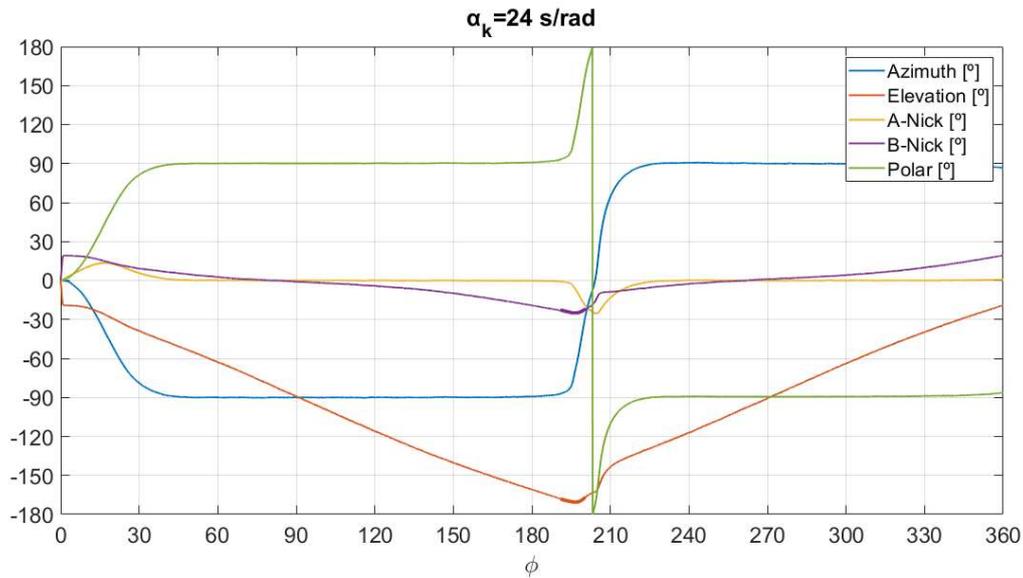


Fig. 7-2: Resulting Trajectory for Automatic Supervisory Control ($\alpha_k = 24 \text{ s/rad}$)

A detailed look into the trajectory reveals the presence of oscillations in the region where ϕ ranges from 190° to 200° . These oscillations are displayed with greater detail in Fig. 7-3 and Fig. 7-4.

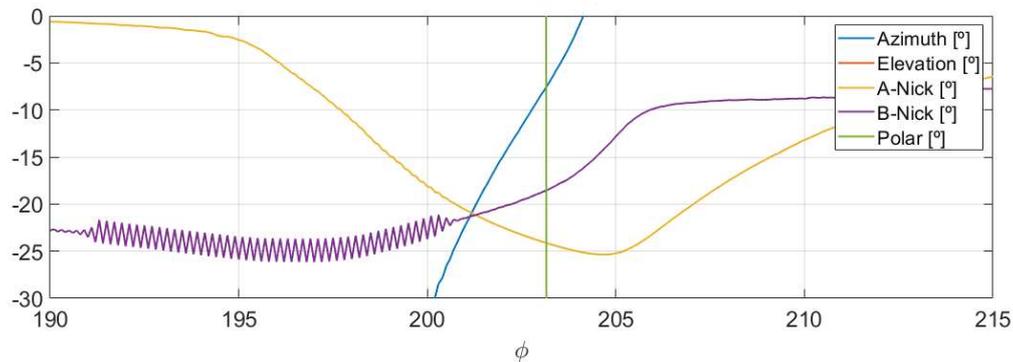


Fig. 7-3: Oscillations in the B-Nick Joint with Automatic Supervisory Control

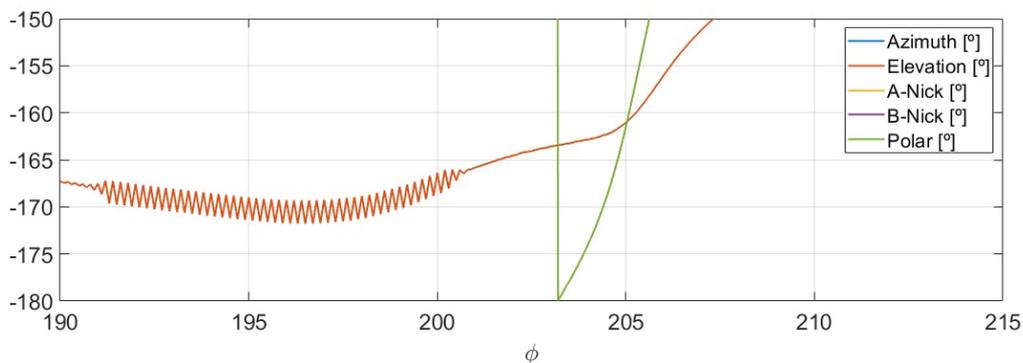


Fig. 7-4: Oscillations in the Elevation Joint with Automatic Supervisory Control

Oscillations are a known problem in ASC [9, 10]. They are prone to take place when local singularities appear, which reduce the robot's capabilities for performing the primary and secondary tasks simultaneously. This is often associated with a situation

in which actuators that are the best positioned for executing the primary task must refrain from moving due to proximity to its joint limit. In these cases, the trajectory has to be executed by joints which are instantaneously less suitable but present a sounder configuration.

In order to visualize how the robot is handling such contradictions internally, the Jacobian of the roll motion with respect to the configuration angles is displayed across the trajectory in Fig. 7-5.

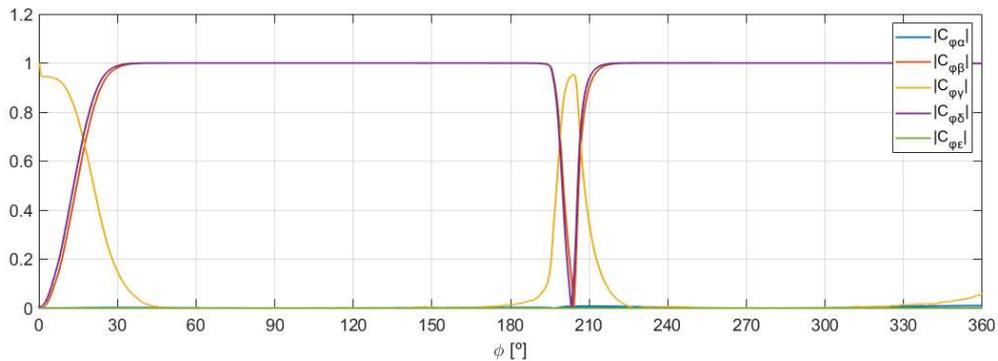


Fig. 7-5: Sensitivity Coefficients of the Roll Motion with Respect to the Configuration Parameters

In this figure, the notation $C_{xy} = |\partial x / \partial y|$ is used. Hence, the values plotted correspond to the absolute value of the first row in Eq. (7-6). As it can be seen, during most of the trajectory, the roll motion is achieved through β and δ . However, when the Elevation angle approaches its limit, the flip between azimuth and polar takes place. During this period, the roll motion is achieved by the A-Nick angle γ in detriment of the Elevation and B-Nick angles. This is done because the Elevation angle would require to dangerously approach to its limit in order to provide roll motion. These oscillations significantly hinder the global performance of the trajectory. Such effect can be observed in Fig. 7-6 where the associated efficiency graph is portrayed. As it can be seen, the areas with high slope ($\varphi \approx 0^\circ$ and $\varphi \approx 180^\circ$) present a significant decrease in efficiency, with values around a 20%.

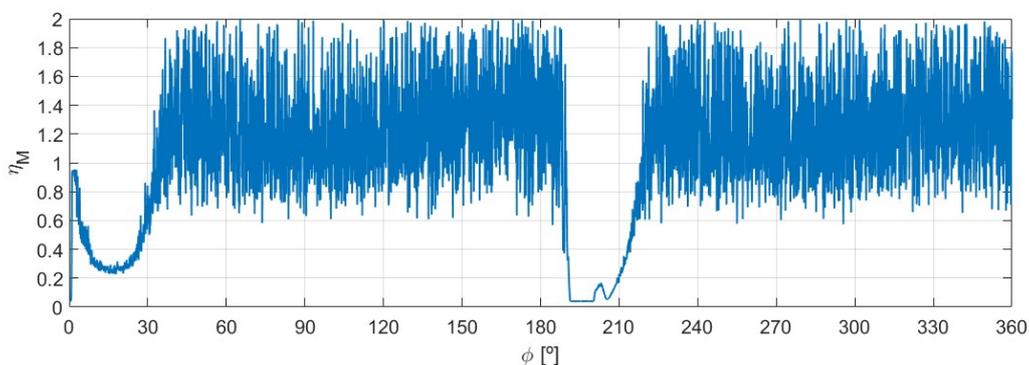


Fig. 7-6: Resulting Efficiency for Automatic Supervisory Control ($\alpha_k = 24$)

Furthermore, the oscillations generate local points in which the efficiency tends to zero. These points are what limit the global efficiency of the algorithm and make this approach unsuitable for practical applications.

7.2.1.3 Trajectory Filtering

There are several methodologies for reducing the oscillations and improving the algorithm's performance. A possible approach can be based on performing a Fourier analysis and removing the harmonics higher than a given Cut-Off Frequency (COF). However, if such approach is applied it is important to bear in mind that the resulting trajectory is not an exact solution to the inverse kinematics anymore. In addition, it is possible that the filtered signal surpasses the limits although the primordial signal does not. Fig. 7-7 illustrates an example of filtered signal with different COFs. As it can be seen, the high frequency oscillations are easily removed. It can also be observed, that the higher the Cut-off Frequency, the less discrepancy between the actual signal and the filtered one.

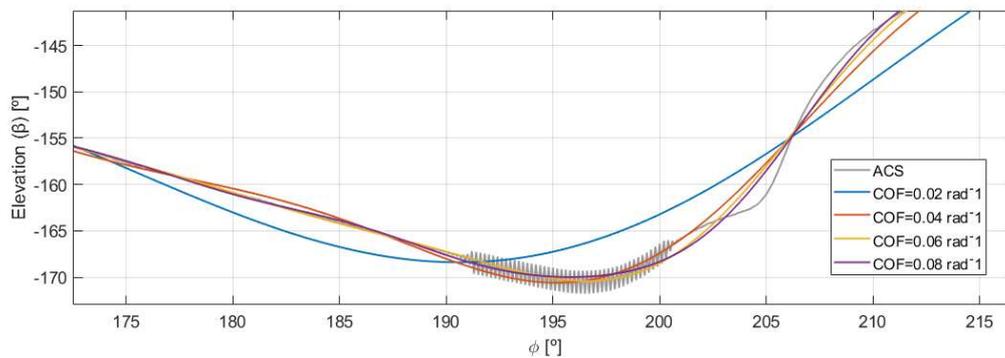


Fig. 7-7: Filtering of the Elevation Angle by Fourier Analysis

The error introduced due to the filtering is presented in Fig. 7-8. As it can be seen, if the Cut-off Frequency is set higher than 0.06 rad^{-1} the average error is generally very small, although it presents some peaks in the region right after the oscillations.

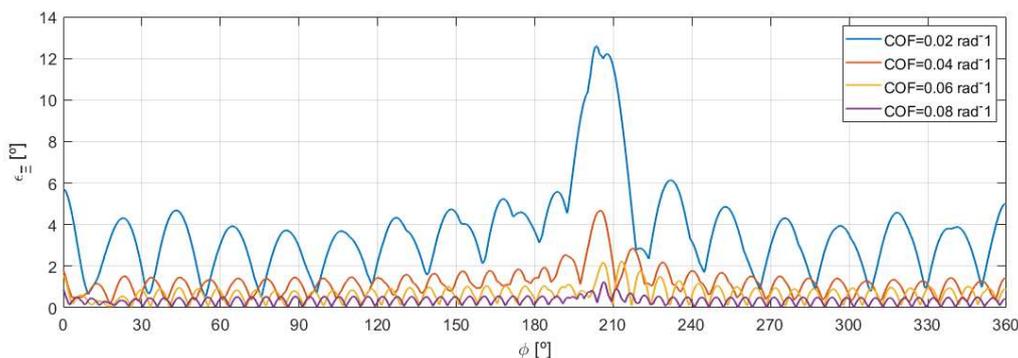


Fig. 7-8: Error Introduced by the filtering the ASC Trajectory for different Cut-off Frequencies

Finally, it is possible to re-calculate the efficiency graph for all the filtered signals. This is illustrated in Fig. 7-9. As it can be seen, the points with zero efficiency are removed thanks to the filtering.

It is possible to observe that the efficiency slightly decreases as the Cut-off Frequency increases. Such behavior is expected, since the high frequency harmonics introduce fast motions into the system. Nevertheless, this effect is rather small.

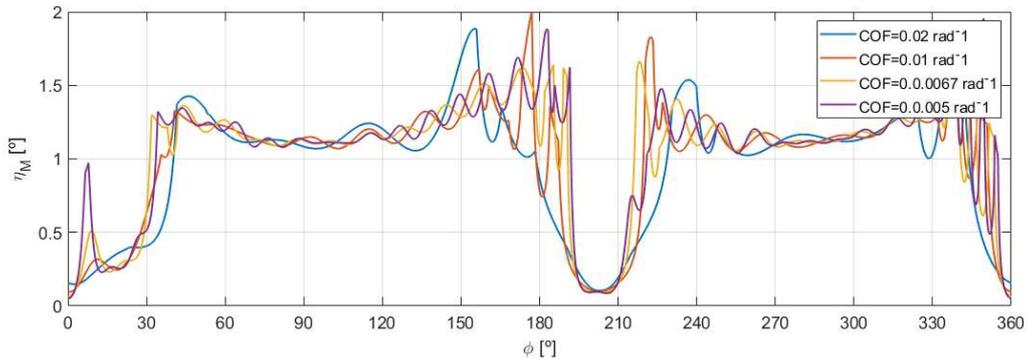


Fig. 7-9: Efficiency of the filtered ASC Trajectory for different Cut-off Frequencies

Finally, it is important to remark that this filtering strategy would only be applicable for static trajectories, since it requires that the whole trajectory is known beforehand. However, there are other possibilities for dynamic filtering, which exceed the scope of the present project.

7.2.2 Weighted Least Norm

The Weighted Least Norm method was proposed by *Chan et.al 1995* in order to address some of the problematics presented by the Automatic Supervisory Control. In particular, oscillations and problems with starting trajectories in positions very close to a joint's limit. This algorithm was used for testing a continuous roll rotation and the results were evaluated with the same structure as the ASC algorithm.

7.2.2.1 Algorithm Description

The weighted least norm method is based on inserting the performance function's derivative inside the weighting matrix. This enables the definition of following cost function:

$$\Phi = \frac{1}{2} (\dot{\alpha}, \dot{\beta}, \dot{\gamma}, \dot{\delta}, \dot{\varepsilon})^T \left(W + \alpha_k \left(I_5 \left(\frac{\partial H(q)}{\partial q} \right)^T \right) \begin{pmatrix} \text{sign}(\dot{\alpha}) \\ \text{sign}(\dot{\beta}) \\ \text{sign}(\dot{\gamma}) \\ \text{sign}(\dot{\delta}) \\ \text{sign}(\dot{\varepsilon}) \end{pmatrix} \right) \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \\ \dot{\delta} \\ \dot{\varepsilon} \end{pmatrix} \text{ Eq. (7-17)}$$

In this case, the primary and secondary tasks are merged into a single one. The sign functions are used in order to preserve the different effects between negative and positive angular velocities. This is necessary since now the performance term depends on the square of the the angular velocities, which cancels negative signs out. The dependency on the angular velocities' second power has also an important effect in the coefficient α_k since it changes its units. As a rule of thumb, the results in ASC should be divided by the average angular velocity of the trajectory (In the current example $\pi/180$ rad/s) in order to deliver similar results. The performance function was chosen to be the same as in Automatic Supervisory Control in order to ease the comparison between the obtained results.

The tuning process for WLN followed the same structure as in ASC and its intermediate results are displayed in the Appendix C.2. In this case the minimal value of α_k was found to be $25^2/\text{rad}^2$. The resulting trajectory is illustrated in Fig. 7-10.

As it can be seen, for this trajectory example the WLN Method does not prevent oscillations from taking place. Furthermore, the B-Nick and Elevation joints present additional lower frequency oscillations in form of signal curling.

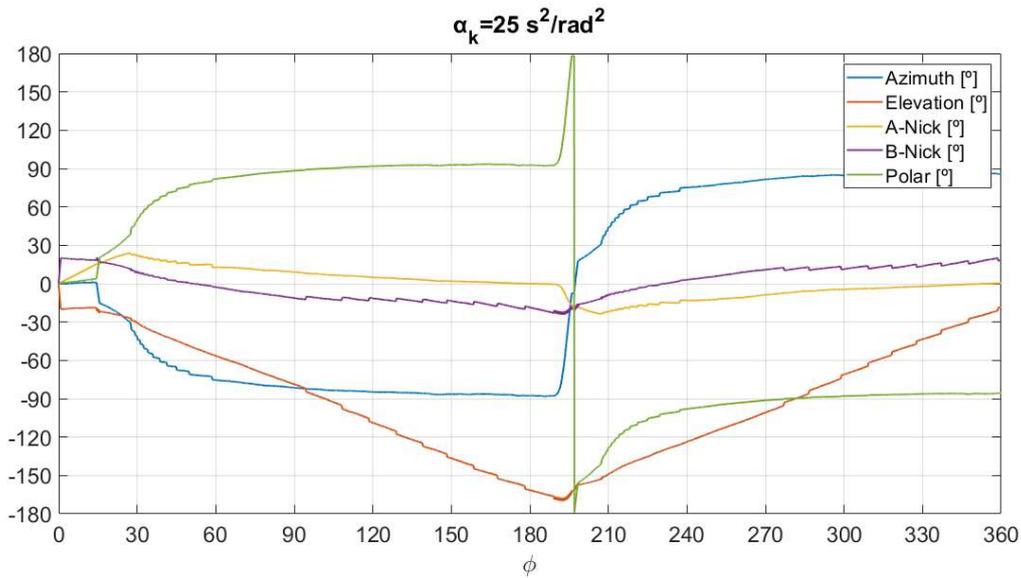


Fig. 7-10: Resulting Trajectory for Weighted Least Norm ($\alpha_k = 25 \text{ s}^2/\text{rad}^2$)

One interesting result of applying this algorithm is that the polar and azimuth angles change their flip way if the coefficient α_k is additionally increased. This can be visualized in Fig. 7-11.

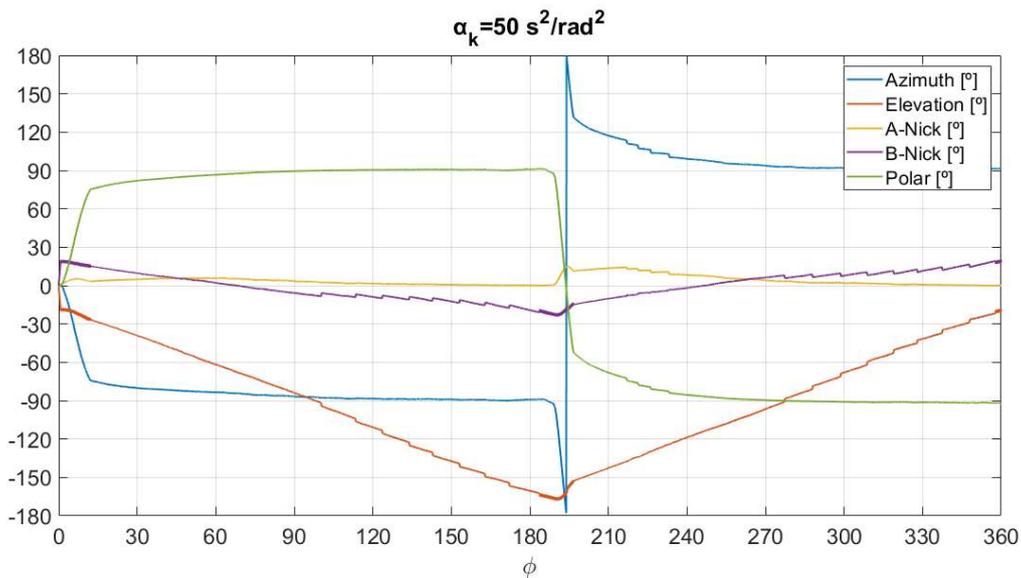


Fig. 7-11: Resulting Trajectory for Weighted Least Norm ($\alpha_k = 50 \text{ s}^2/\text{rad}^2$)

The efficiencies associated to these trajectories are displayed in Fig. 7-12 and Fig. 7-13. As it can be seen, this approach presents a similar performance as ASC. One

possible reason behind this lack of improvement is the high level of constraints of the joints. Since in WLN the optimization tasks are merged

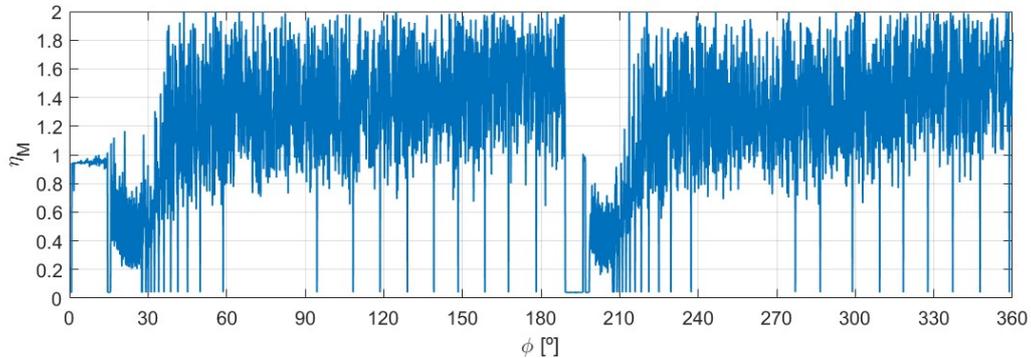


Fig. 7-12: Resulting Efficiency for Weighted Least Norm ($\alpha_k=25 \text{ s}^2/\text{rad}^2$)

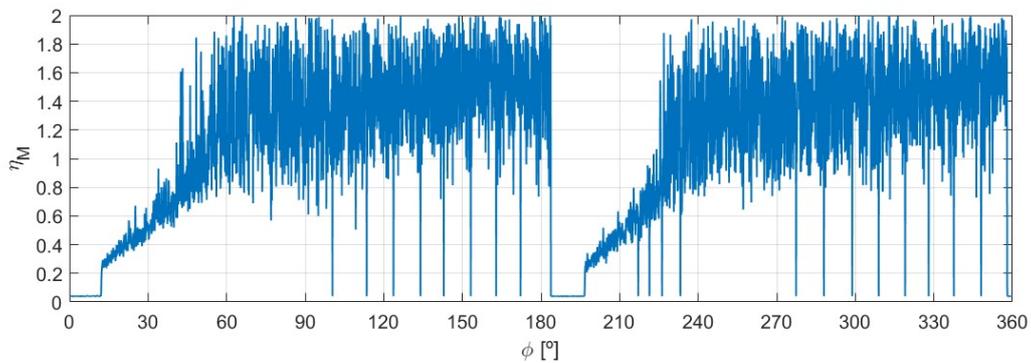


Fig. 7-13: Resulting Efficiency for Weighted Least Norm ($\alpha_k=50 \text{ s}^2/\text{rad}^2$)

7.2.2.2 Trajectory Filtering

As in the previous case, it is possible to perform a filtering of the signal in order to smooth it and improve its performance. In this case the filtering

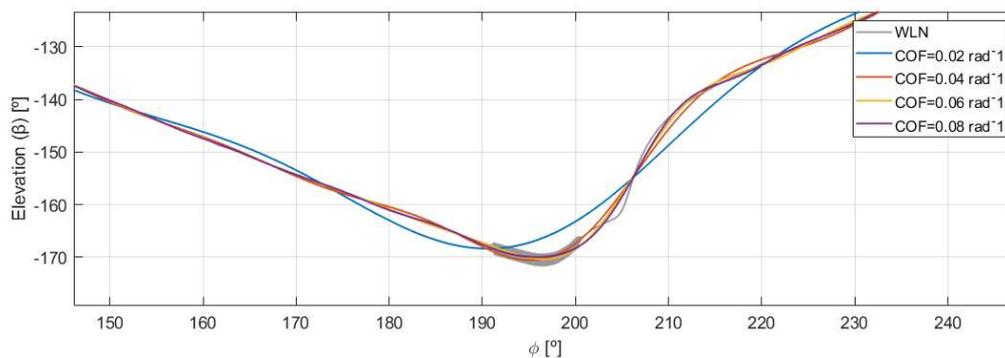


Fig. 7-14: Filtering of the Elevation Angle by Fourier Analysis

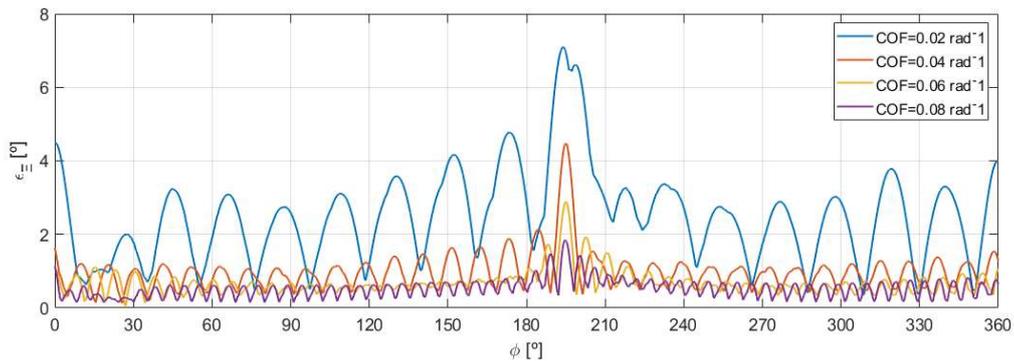


Fig. 7-15: Error Introduced by the Filtering the WLN Trajectory for Different Cut-off Frequencies

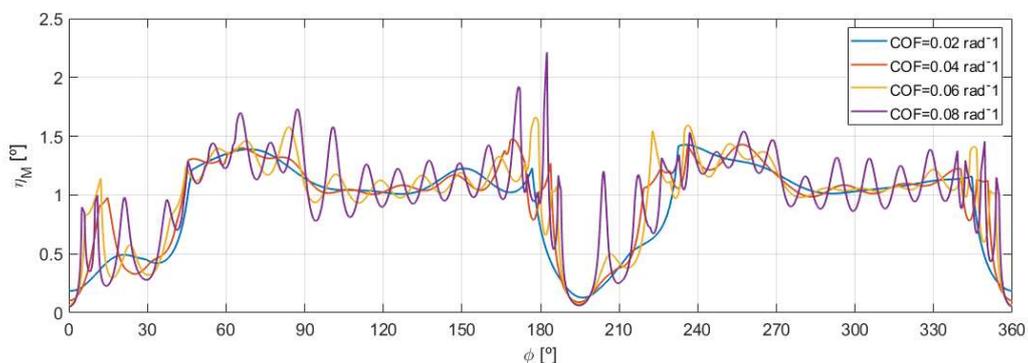


Fig. 7-16: Efficiency of the Filtered ASC Trajectory for different Cut-off Frequencies

7.3 Algorithms based on the Analytical Solution

The trajectory problem can be addressed from the analytical solution to the inverse kinematics too. In this case, the inverse kinematics manifold is obtained for each way point of the trajectory and the algorithm has to decide the most suitable solution within each of these manifolds. Although the physical process behind the problem remains identical, the analytical approach and the solution in the velocity domain present very different mathematical formulation. In the case of the analytical solution, what is known is the configuration at the i^{th} way point q_i and the inverse kinematics manifold at the $i + 1^{th}$ point σ_{i+1} . This enables the determination of all the possible angular velocities which lead to the following waypoint as:

$$\dot{\sigma}_i \approx \frac{\sigma_{i+1} - q_i}{\Delta t} \quad \text{Eq. (7-18)}$$

This can be assumed to be the manifold of \dot{q} , i.e. the solution of the inverse kinematics in the velocity domain. This can be obtained as well with the Jacobian as it was done in the previous section. However, the advantage of this approach is that $\dot{\sigma}_i$ contains all the velocities based on points within the actuator's joints. This formulation is highly beneficial as it enables more comprehensive understanding of the locking problematic and the robot's limitations as it will be seen in the following lines.

7.3.1 Minimum Instantaneous Control Effort

The minimum instantaneous control effort algorithm can be considered as the equivalent of Resolved Motion Rate Control for the analytical kinematics. This is the same as using the Automatic Supervisory Control algorithm with a null coefficient α_k . Therefore, this algorithm seeks the solution within the next way-point's manifold q_{i+1} which requires the minimum instantaneous angular velocities (maximum efficiency):

$$q_{i+1} = \underset{\dot{\sigma}_i}{\operatorname{argmax}} \{ \eta_M(\dot{\Sigma}_i, \dot{\sigma}_i) \} \quad \text{Eq. (7-19)}$$

The resulting trajectory for this algorithm is displayed in Fig. 7-17 and its associated efficiency graph is displayed in Fig. 7-18.

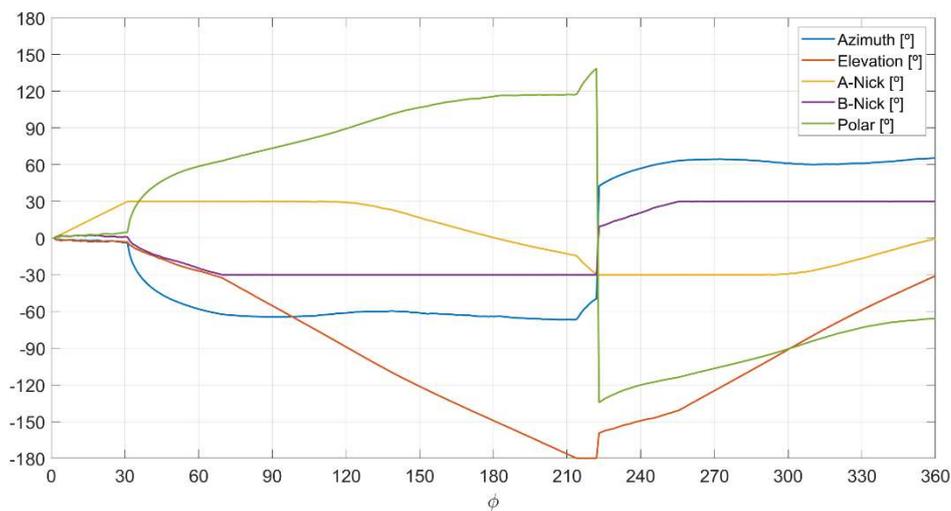


Fig. 7-17: Trajectory Obtained by the Minimum Instantaneous Control Effort Algorithm

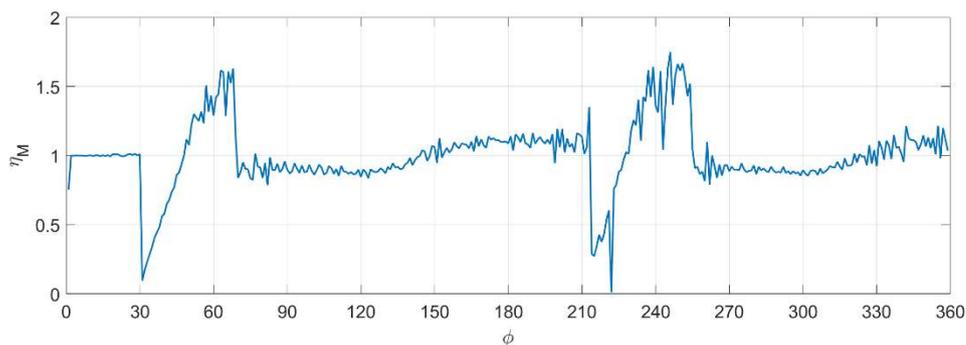


Fig. 7-18: Trajectory Efficiency Obtained by the Minimum Instantaneous Control Effort Algorithm

The Results are the same for RMRC until $\varphi = 30^\circ$. However, the MICE approach is able of maintaining a smooth trajectory afterwards, since the algorithm is aware of the full manifold of solutions. Nevertheless, if several joints are constrained, the locking problematic can occur too. This is what can be observed for $\varphi = 220^\circ$. In this particular case, it can be seen that the B-Nick angle is the first one which reaches a limit position. Subsequently, the elevation angle reaches its terminal position for $\varphi = 213^\circ$. With every actuator positioned in a terminal position a degree of freedom is lost. Therefore, when the A-Nick angle reaches a terminal position at $\varphi = 220^\circ$ the system has only

two actuators free and is not able to follow the trajectory with the same variations as it was doing it anymore. In this situation, there are no more solutions to the inverse kinematics in its current position and an abrupt change has to be executed. In this sort of transitions the efficiency tends to zero. This implies that the time or the angular velocity that the facility would require to execute the planned maneuver tends to infinite. This is by no means acceptable from a practical standpoint. In order to understand the mechanism which yields to a locking problematic, the whole solution domain for the different positions of the trajectory was studied. The reader can find these solutions in the Annex. Fig. 7-19 illustrates the solution to the inverse kinematics for several points around the locking point.

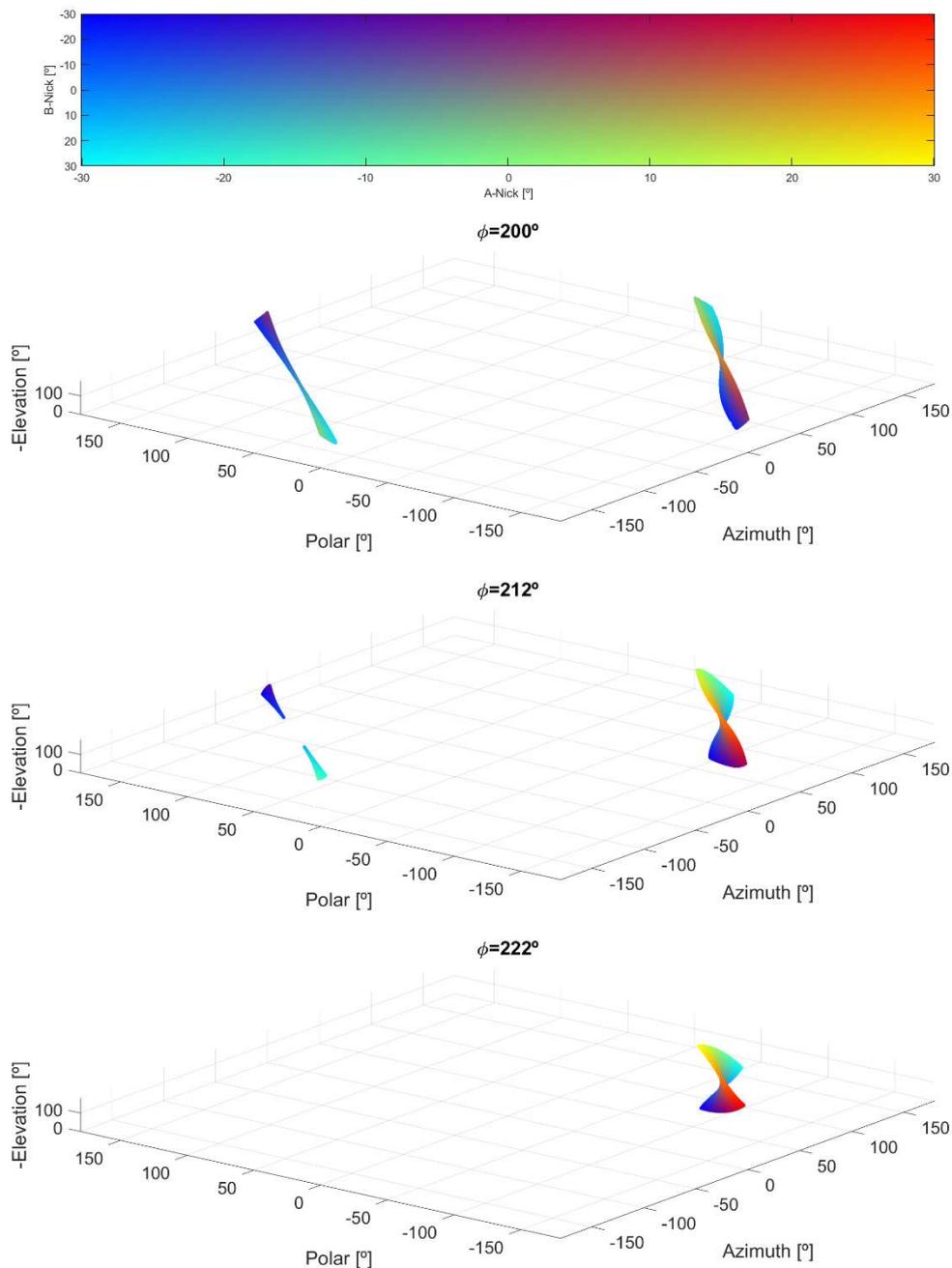


Fig. 7-19: Inverse Kinematics Manifold for Trajectory's Points around the Locking Instant

As it can be seen, for $\varphi = 200^\circ$ there are two different solution regions. The solution for the trajectory chosen by the MICE algorithm is located in the area with positive polar and negative azimuth. At this point of the trajectory, both solution regions are connected, but the link is lost by $\varphi = 210^\circ$. After this point, the region where the current solution is placed tends to become smaller as the trajectory progresses. This isolated region disappears by $\varphi = 222^\circ$. This is what causes the locking problematic, the lack of possibilities to progress with nearby points with the current configuration. This analysis illustrates that the MICE algorithm promotes that the solution stays in the initial region, which is inimical in the long term. The reason behind this behavior is that the configuration gradient within the two solution regions is higher, so looking for nearby points results in staying within the same region although it tends to disappear.

Therefore, it is necessary to develop algorithms which promote the transition to the other solution region once this one is become larger in detriment to the first one.

7.3.2 Isokinematic Centroid Tracking

As it was seen in the previous point, the initially proposed policy presents the problem that a configuration may tend to end in a cornered position of the solution domain σ , which can eventually become inaccessible. Therefore, a reasonable policy may be based on avoiding being in such positions. This would require that the trajectory heads towards the new solution region when it appears whilst the bridge between both regions is still standing. A possible way to implement such behavior is that the trajectory tracks the centroid of the solutions. In this case, when a secondary solution region appears the algorithm will promote the motion towards this region as its size grows. This arises the question of defining the centroid of a solution space based on angular variables. A reasonable way to do this is to perform a statistical analysis of the solution domain. The mean or most probable value can be assumed to be the equivalent of a centroid for the current context. In order to perform this analysis, the probability density function of the configuration for each actuator for each point of the trajectory was computed. The results of these calculations are displayed in the following figures. In these graphs, the color code indicates the probability density, the vertical axis indicates the configuration of the actuator, whilst the horizontal is related to the point of the trajectory.

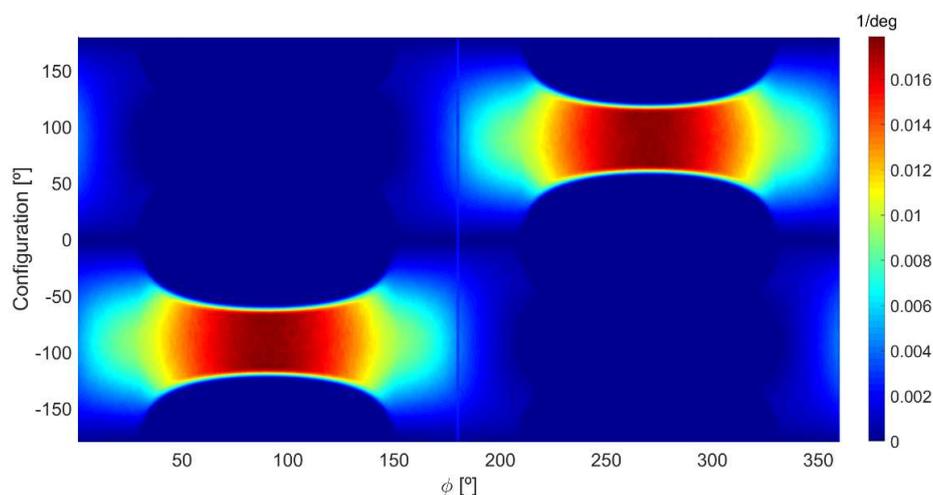


Fig. 7-20: Probability Density Function of the Azimuth Joint through the Example Trajectory

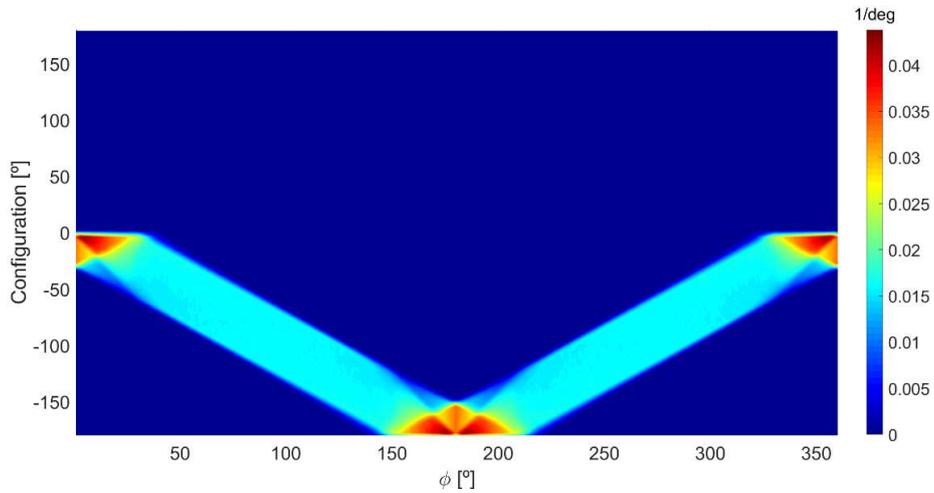


Fig. 7-21: Probability Density Function of the Elevation Joint through the Example Trajectory

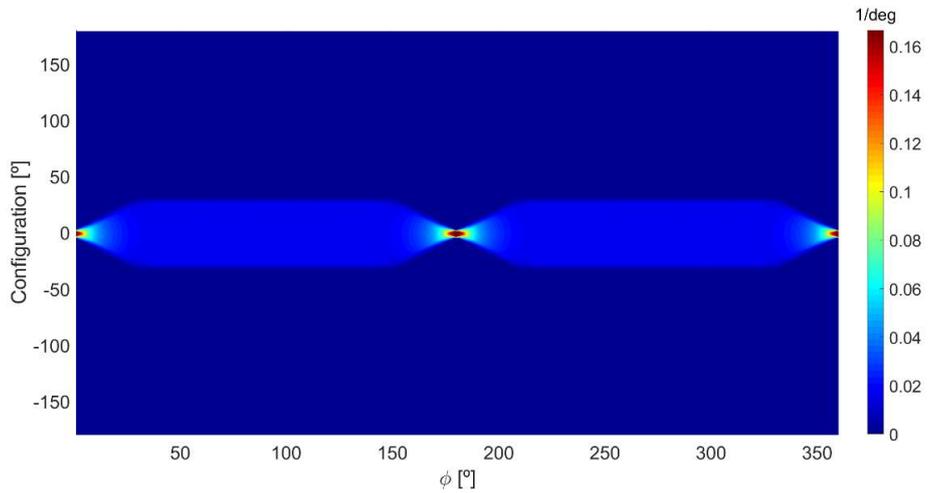


Fig. 7-22: Probability Density Function of the A-Nick Joint through the Example Trajectory

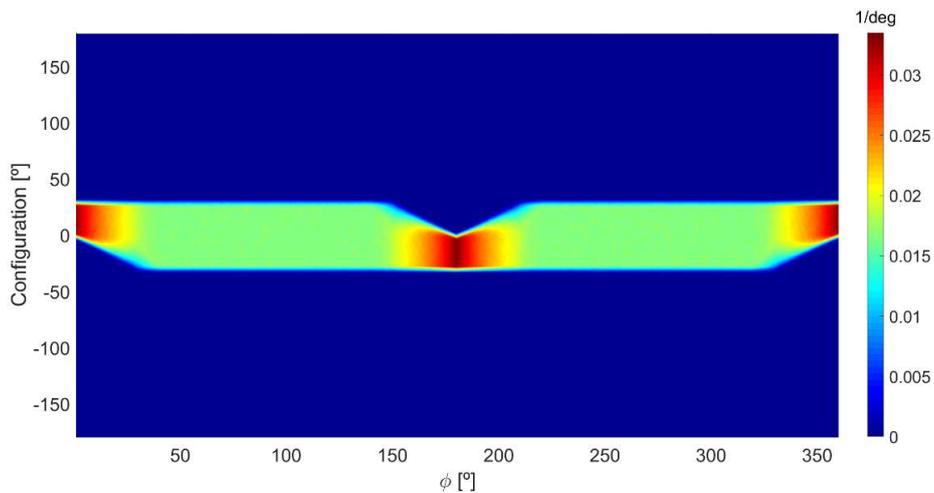


Fig. 7-23: Probability Density Function of the B-Nick Joint through the Example Trajectory

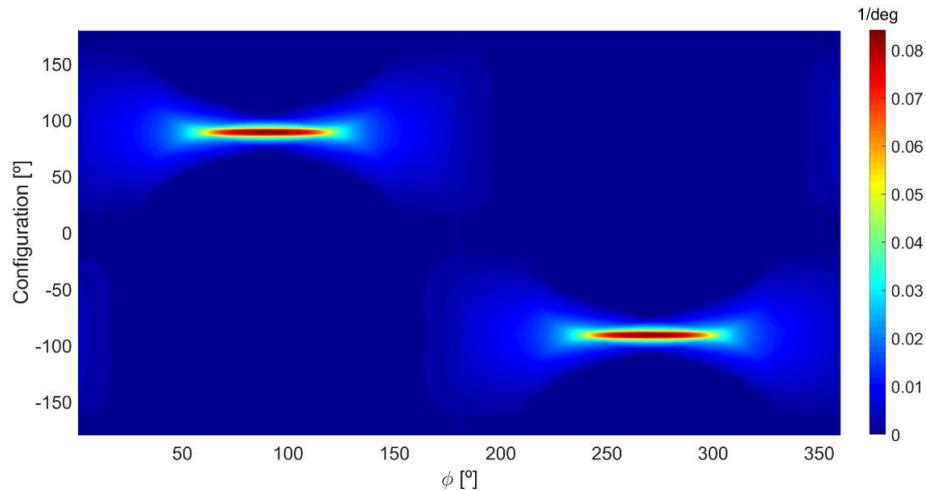


Fig. 7-24: Probability Density Function of the Polar Joint through the Example Trajectory

These figures illustrate the regions in which most solutions are concentrated for each point of the trajectory. Therefore, they provide relevant insights involving the locking problematic. For example, it can be seen, that the unconstrained actuators present lower peaks compared to the constrained ones. High peaks of values indicate that many solutions are concentrated around a certain value. This is a high indicator of locking danger. The actuator must reach such positions soon enough in order to prevent locking from taking place. The presence of simultaneous solution regions can be observed in the presented figures too. A closer look into the azimuth and polar angles indicates that for some parts of the trajectory there is a single region where most of the solutions are concentrated. As φ approaches 180° , the region where most of the solutions are concentrated changes. In order to determine objectively the ideal point, it is necessary to compute the most probable value for a given point of the trajectory. For the actuators which are limited, this calculation is analogous for the ordinary mean:

$$\omega_{i_L, \mu} = \int_{\omega_{min}}^{\omega_{max}} f_{\omega}(\omega) d\omega \quad \text{Eq. (7-20)}$$

Where f_{ω} is the probability density function of the actuator over its configuration. For a robot with n_L limited actuators, computing the different most probable values leads to the $n_L \times 1$ vector of most probable values for the limited actuators:

$$q_{\mu_L} = (\omega_{1, \mu} \quad \omega_{2, \mu} \quad \dots \quad \omega_{i_L, \mu} \quad \dots \quad \omega_{n_L, \mu})^T \quad \text{Eq. (7-21)}$$

Which in the case of the RACOON Lab results in:

$$q_{\mu_L} = (\beta_{\mu} \quad \gamma_{\mu} \quad \delta_{\mu})^T \quad \text{Eq. (7-22)}$$

Unlike these limited degrees of freedom, the actuators which can range from 0 to 360° are cyclic. In these cases, it is necessary to introduce certain modifications to the generalized notion of the most probable value. In order to visualize this necessity Fig. 7-25 illustrates the Probability Density Function of the Azimuth for several configurations.

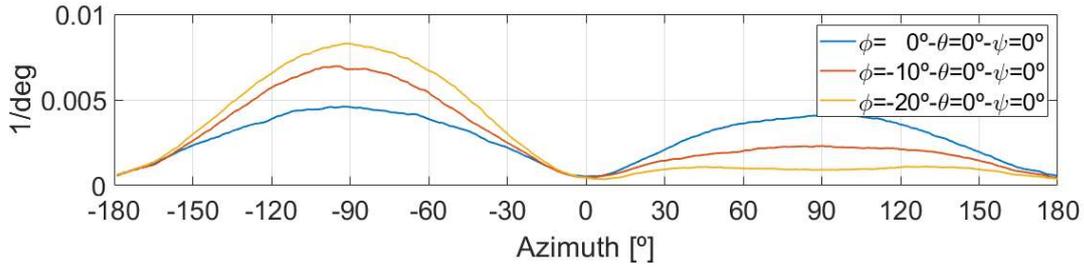


Fig. 7-25: Probability Density Function for the Azimuth for Different Coordinates

In this Figure, the two different solution regions can be observed very clearly. Furthermore, it can be seen how one of these regions decreases at expenses of the other one as the roll angle reduces. In order to promote the transition from one region to another, the target position should be between the two local maximums of the two solutions. Such behavior cannot be obtained with the conventional definition of the mean. If one uses this definition, the result will be dependent of the integration boundaries. For example, if the PDF is defined from 0 to 360° the result would be between 180° and 270°, but if the boundaries of Fig. 7-25 are used, the result would lay between -90° and +90°. Indeed, these two values are valid as a variable's centroid since they lay within an intermediate position of the two regions. In order to determine these two centroids for the general case, the following procedure is followed:

1. Find the configuration with the highest density Ω (-90° in Fig. 7-25)

$$\Omega = \underset{\omega}{\operatorname{argmax}}\{f(\omega)\} \quad \text{Eq. (7-23)}$$

2. Find the bounds r_0 and r_1 (-180° and 0° in Fig. 7-25) of the region that contains Ω :

$$r_0 = \underset{\omega}{\operatorname{argmin}}\{\Omega - \omega\} \quad \forall \omega < \Omega \mid f(\omega) = 0 \quad \text{Eq. (7-24)}$$

$$r_1 = \underset{\omega}{\operatorname{argmin}}\{\omega - \Omega\} \quad \forall \omega > \Omega \mid f(\omega) = 0 \quad \text{Eq. (7-25)}$$

3. Use with these inputs to determine the two most probable values as:

$$\begin{cases} \omega_{i_c, \mu_a} = \int_{r_0}^{r_1} f(\omega) d\omega + \pi \left(1 - \int_{r_0}^{r_1} d\omega \right) \\ \omega_{i_c, \mu_b} = 2\Omega - \omega_{i_c, \mu_a} \end{cases} \quad \text{Eq. (7-26)}$$

These two parameters lead to the matrix of the $n_c \times 2$ matrix of probable values for the cyclic actuators:

$$q_{\mu_c} = \begin{pmatrix} \omega_{1, \mu_a} & \omega_{2, \mu_a} & \dots & \omega_{i_c, \mu_a} & \dots & \omega_{1, \mu_b} \\ \omega_{1, \mu_b} & \omega_{2, \mu_b} & \dots & \omega_{i_c, \mu_b} & \dots & \omega_{1, \mu_a} \end{pmatrix}^T \quad \text{Eq. (7-27)}$$

Which in the case of the RACOON Lab's rotator results in:

$$q_{\mu_c} \begin{pmatrix} \alpha_{\mu_a} & \varepsilon_{\mu_a} \\ \alpha_{\mu_b} & \varepsilon_{\mu_b} \end{pmatrix}^T \quad \text{Eq. (7-28)}$$

The following figure illustrates the evolution of the different probable values for RACOON-Lab over the study trajectory.

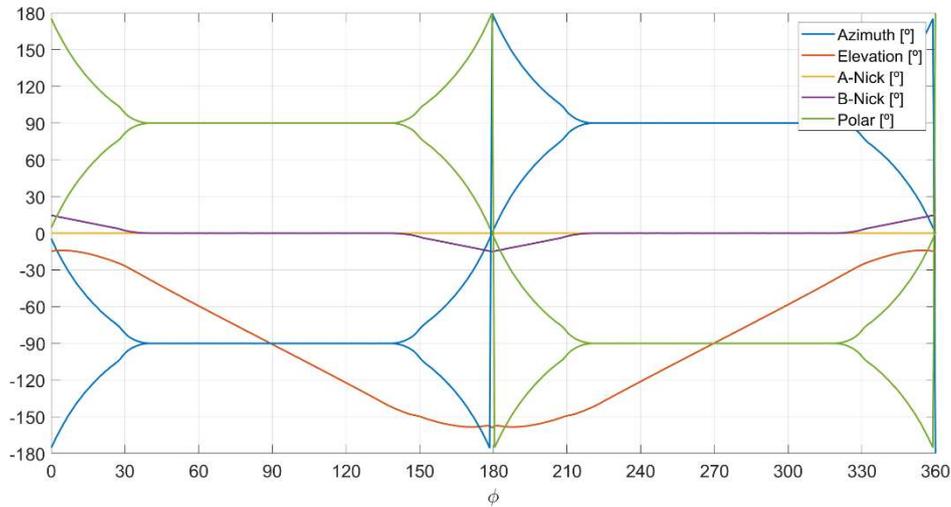


Fig. 7-26: Most Probable Values for the Different Joints over the Example Trajectory

Using these concepts, it is possible to define the minimum distance to a most probable value of a given configuration as:

$$\zeta(q, q_{\mu_L}, q_{\mu_C}) = \sum_{i_L=1}^{n_L} (\omega_{i_L} - \omega_{i_L, \mu})^2 + \sum_{i_C=1}^{n_C} \min \left\{ \begin{array}{l} (\omega_{i_C} - \omega_{i_C, \mu_a})^2 \\ (\omega_{i_C} - \omega_{i_C, \mu_b})^2 \end{array} \right\} \quad \text{Eq. (7-29)}$$

This definition can be used as a criterium for determining the evolution of the trajectory. In particular, the value the configuration which presents the minimizes the minimum distance to a most probable (isokinematic centroid) value over the solution space σ is the one chosen for the next point of the trajectory:

$$q_{i+1} = \underset{\sigma_{i+1}}{\operatorname{argmin}} \{ \zeta(\sigma_{i+1}, q_{\mu_L}(\sigma_{i+1}), q_{\mu_C}(\sigma_{i+1})) \} \quad \text{Eq. (7-30)}$$

The output of using this algorithm for the example trajectory is displayed in Fig. 7-27 and the associated efficiency can be visualized in Fig. 7-28.

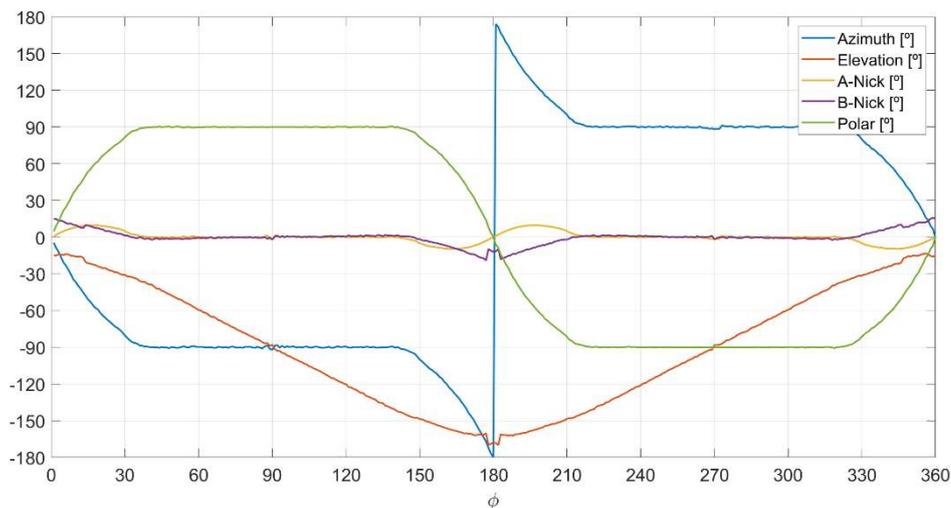


Fig. 7-27: Trajectory with the Isokinematic Centroid Tracking

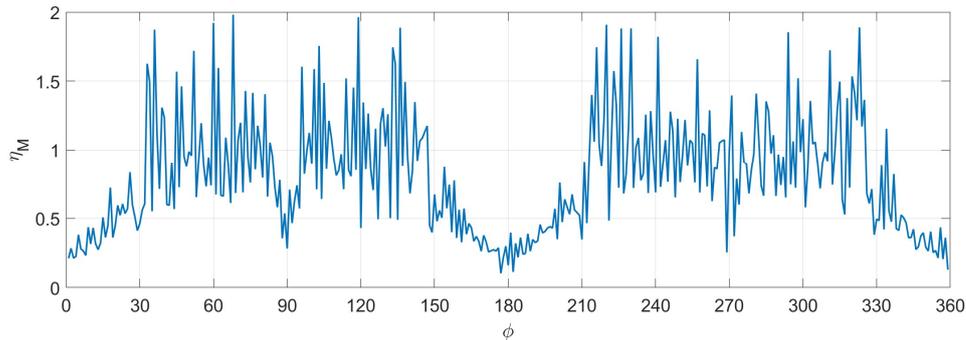


Fig. 7-28: Efficiency with the Isokinematic Centroid Tracking

This solution for the trajectory provides an efficiency of a 21%, which as it can be seen is capable of providing a seamless rotation around the X axis. Nevertheless, at some positions of the trajectory, high control efforts are required for both azimuth and polar angles. These sporadic high control efforts are what produce punctual low efficiencies, which diminish the global performance. Therefore, although the motivation behind the algorithm is sound some minor problems remain. This issues shall be addressed in the next point.

7.3.3 Relaxed Isokinematic Centroid Tracking

At this point a seamless trajectory with an acceptable efficiency has been achieved. However, the question is whether it is possible to improve it and obtain smoother paths which enable a more optimal utilization of the facility. The Relaxed Isokinematic Centroid Tracking (RIKCT) algorithm intends to avoid the sporadic abrupt changes in configuration in order to improve the overall efficiency of the algorithm. This is done by limiting the maximum instantaneous control effort for tracking the most probable values. Therefore, this variant of the algorithm limits the selection to points that imply transitions with a transition that has an efficiency higher than the minimum one which was specified:

$$q_{i+1} = \underset{\sigma_{i+1}}{\operatorname{argmin}} \{ \zeta(\sigma_{i+1}, q_{\mu_L}(\sigma_{i+1}), q_{\mu_C}(\sigma_{i+1})) \} \forall \eta_M \geq \eta_{min} \quad \text{Eq. (7-31)}$$

Such approach can avert the presence sporadic rather abrupt inputs within the trajectory such as the one observed in Fig. 7-27. However, implementing this approach implies that the isokinematic centroid is not followed as accurate as before. Therefore, if the maximum allowed control effort becomes too small, it is possible that the configuration is so far away from the most probable values that the locking problematic appears again. Therefore, it is required to tune this value i.e. finding the highest efficiency which still able to track the isokinematic centroid with an error that does not lead to locking configurations.

An iterative approach is used in order to maximize the efficiency of the trajectory whilst avoiding this risk. The iterations depart from the efficiency achieved by the IKCT Algorithm and this value is increased gradually until the point when locking takes place. This procedure was followed for the example trajectory and a maximum efficiency of a 50% was obtained. The reader can find the intermediate results of the tuning process

in the Appendix E The optimal trajectory and its efficiency are displayed in the following figures:

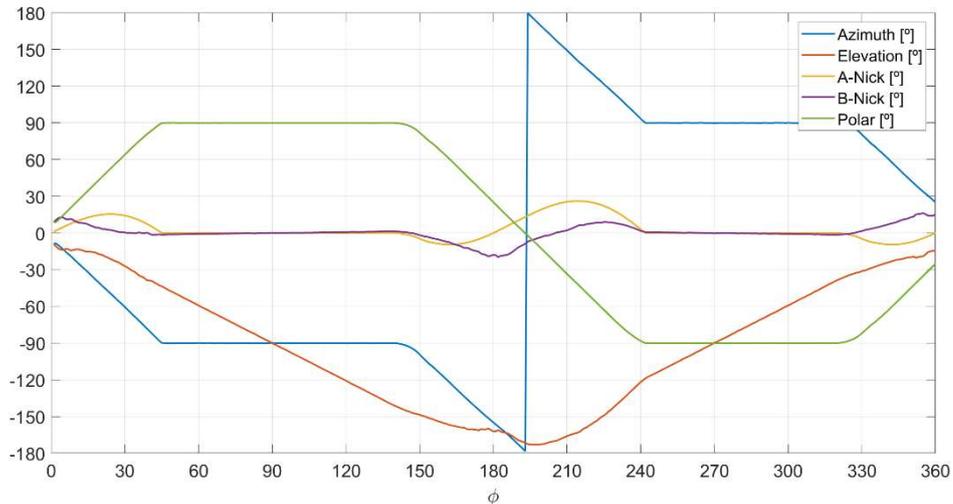


Fig. 7-29: Trajectory Obtained with the Relaxed Isokinematic Centroid Tracking with a Minimum Efficiency of a 50%

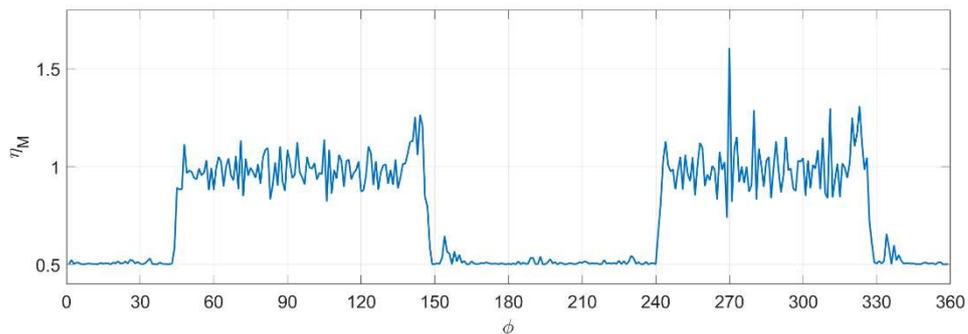


Fig. 7-30: Efficiency Obtained with the Relaxed Isokinematic Centroid Tracking with a Minimum Efficiency of a 50%

The imposed limitation of the control effort becomes evident with a detailed look at Fig. 7-30 where the efficiency of each discrete maneuver is displayed.

It is important to note that this algorithm is particularly well suited for static simulations, when the whole trajectory is known beforehand. For these cases, it is possible to tune the value of η_{min} accurately as it is possible to try every value until locking appears. In the case of dynamic experiments, the efficiency must be set beforehand and it is not possible to iterate. In such situations, a conservative approach is advised. Therefore, a reasonable value for the limitation of the maximum control effort would be an efficiency around a 30%.

7.3.4 Automatic Supervisory Control with Distance to an Isokinematic Centroid as Performance Criterion

Another possibility in order to improve the performance of both ASC and the IKCT can be to merge these two approaches. In this direction, the distance to the most probable value can be used as the performance criterium, thus:

$$H(q) = \underset{\sigma_{i+1}}{\operatorname{argmin}} \{ \zeta(\sigma_{i+1}, q_{\mu_L}(\sigma_{i+1}), q_{\mu_C}(\sigma_{i+1})) \} \quad \text{Eq. (7-32)}$$

As in the case of ASC it is necessary to tune the value of α_k . In this case, the optimal value was found to be 0.8 with the trajectory and associated efficiency displayed in Fig. 7-31 and Fig. 7-32 respectively.

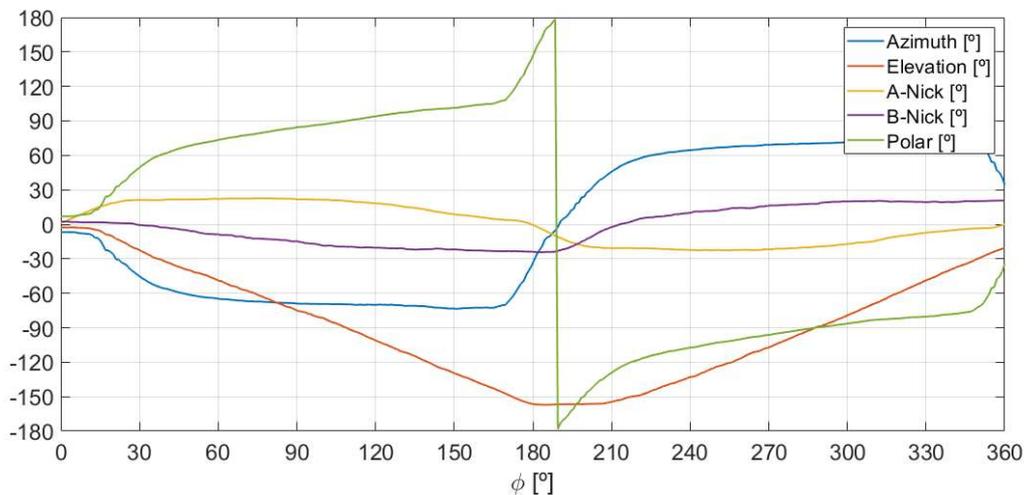


Fig. 7-31: ASC with Distance to the Minimum Distance to an Isokinematic Centroid as the Performance Criterion

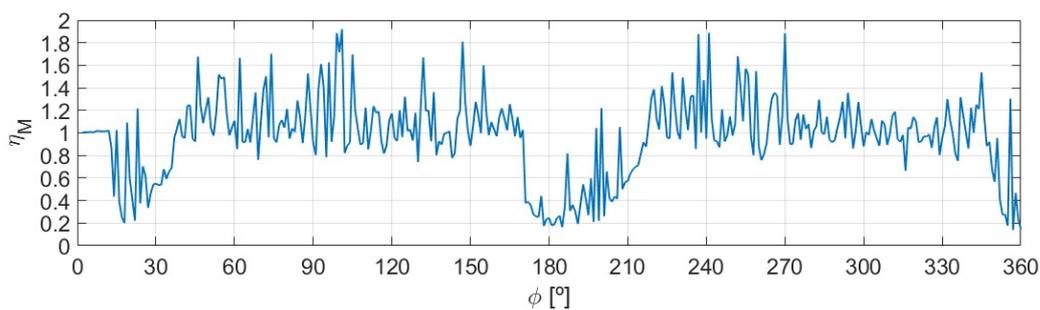


Fig. 7-32: Efficiency Obtained with ASC using the PDF as the Performance Criterion

As it can be seen, this approach provides a higher efficiency at the beginning of the trajectory. Nevertheless, when the azimuth and polar angles must perform the flip, high angular velocities are required, greatly diminishing the global performance of the algorithm. In particular, a minimum efficiency is a 14%, which provides no improvement at all compared to the IKCT approach although it is slightly better than the result obtained by ASC.

7.4 Summary

During the present chapter, several algorithms were introduced and tested for executing a roll rotation in the RACOON-Lab. In this final section, the obtained results are presented in a summarized way, in order to provide a general perspective of the performed work.

The conventional methodologies (ASC and WLN) exhibited oscillations, that could be removed by filtering techniques. However, this filtering introduced an attitude error, which decreased as the cut-off frequency increased. Even considering the efficiency enhancement due to filtering, the efficiency of these algorithms was in the order of a 10%.

Several algorithms based on the analytical solution to the inverse kinematics were developed. The first approach, MICE, focused on minimizing the instantaneous control effort, which eventually lead to locking configurations. Such configurations could be averted by tracking the centroid of the isokinematic manifold. The algorithms based on this philosophy were IKCT and RIKCT and proved able to execute the roll rotation with efficiencies over 30% in the absence of oscillations. The former algorithm tracks the isokinematic centroid with as much control effort as necessary, whilst the later enables a certain relaxation of the tracking task that improves the overall trajectory's efficiency. The consideration of the isokinematic centroid was introduced into conventional methodologies too, by using the distance to an as the performance criterium in ASC. The efficiency of this approach was higher than ASC but poor compared to the isokinematic tracking algorithms. All these results are summarized in Table 7-1.

Table 7-1: Summary of Performance for the Inverse Kinematic Algorithms tested for a Roll Rotation

Algorithm	Efficiency, η	Oscillations	Attitude Error, ϵ_E
ASC	1.6%	High frequency	0°
ASC+Filtering	9.3%	No	1°-10°
WLN	2.1%	Low and High frequency	0°
WLN+Filtering	10.2%	No	1°-10°
MICE	0%	No	0°
IKCT	19.4%	No	0°
RIKCT	50%	No	0°
ASC+IKCT	19.6%	No	0°

8 System Evaluation

The previous chapters have provided a solid background for understanding the RACOON-Lab's kinematics. In this section this asset is used in order to evaluate the design of the robot and its capabilities. This assessment will focus on studying the sensitivity of the RACOON-Lab's performance with respect its joints limitations and the singularities that can take place at certain configurations.

8.1 System Capabilities

After visualizing the different trajectories obtained in the previous section, it is possible to appreciate several common patterns. This exercise can be used in order to identify the system's weaknesses from a kinematic point of view.

With a closer look at every trajectory, it can be observed that the most problematic points are concentrated around $\varphi = 180^\circ$. It is in this area where the deadlocks take place. Furthermore, even if locking can be avoided, it is in this region where the points with the lowest efficiencies are concentrated. This indicates that the most demanding movements take place around this area. The reason behind this is the presence of two different isokinematic regions as it was identified in section 7.3.1. These regions are connected for a reduced number of positions, and the connection is only available for a short period. In order to avoid ending in a deadlock, the robot must transition from one region to another. This transition is done by tracking the centroid of the inverse kinematics manifold, which often moves steeply. Therefore, the gradient of the inverse kinematics' centroid with respect to the space coordinates is the limiting factor in the robot's performance.

A possible way to assess this steepness is to evaluate the slope of the most probable values for the joints that must perform the most abrupt transitions. These joints are the azimuth and polar angles. Fig. 8-1 illustrates this slope for the case of the Azimuth angle. The absolute value of the slope is the same for the polar angle, therefore only the former angle will be considered from this point on.

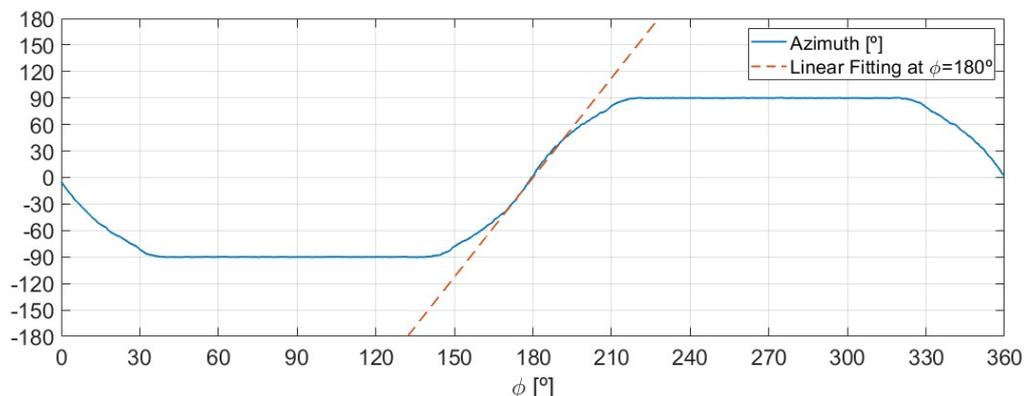


Fig. 8-1: Linearization of Inverse Kinematics Centroid Motion around $\varphi=180^\circ$

This slope is the main driver of the robot's capabilities for performing efficient trajectories. A suitable way to express it is as the sensitivity of the Azimuth angle's most probable value with respect to the roll angle and is a function of the joints' limitations N :

$$C_{\alpha_{\mu\varphi}}(N) = \left| \frac{\partial \alpha_{\mu}}{\partial \varphi}(N) \right| = \left| \frac{\partial \alpha_{\mu}}{\partial \varphi}(v_{\beta}, v_{\gamma}, v_{\delta}) \right| \quad \text{Eq. (8-1)}$$

N denotes the vector containing the joint's range of motion of the limited joints.

If the ranges become wider, so does the inverse kinematics manifold. Therefore, new isokinematic regions arise sooner and vanish later, which provides a larger margin for executing the transition, enabling greater efficiencies. In order to assess this effect, this gradient was evaluated changing the ranges of the constrained joints. In order to provide more intuitive results, the ranges of the Nick joints were merged into a single variable representing the \pm range. The results of this study are displayed in Fig. 8-2.

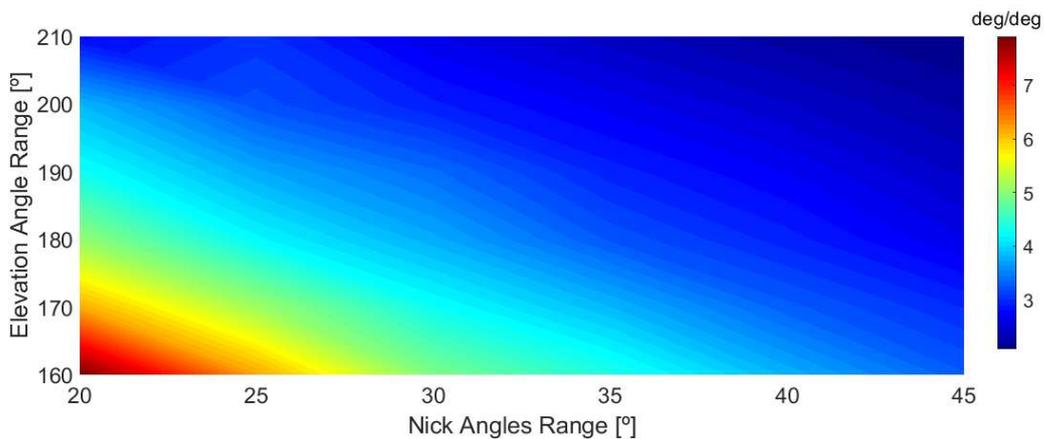


Fig. 8-2: Inverse Kinematics Centroid Gradient in a Roll Rotation for the RACOON-Lab

In this Figure, the Elevation angle range refers to the total range of motion of the elevation angle, which has a central value at $\beta = -90^{\circ}$. The Nick angles range is the \pm range. Therefore, a value of 30° corresponds to the default limits presented in Table 5-1. As it can be seen, the slope tends to increase as both ranges are reduced. The inverse of this slope can be considered as a reference value for the efficiency necessary for executing the roll rotation averting locking configurations. This value can be fitted very well as a plane with the following expression:

$$\frac{1}{c_{\alpha_{\mu\varphi}}} = c_0 + c_n v_n + c_{\beta} v_{\beta} \quad \text{Eq. (8-2)}$$

The regression of this formulation presents an R^2 value of 0.985 with the following numerical results:

$$\begin{cases} c_0 = -0.6033 \\ c_n = 6.8474 \cdot 10^{-3} \text{deg}^{-1} \\ c_{\beta} = 3.7065 \cdot 10^{-3} \text{deg}^{-1} \end{cases}$$

As it can be seen, elevation and nick angles have about the same relevance, since the parameter v_n refers to the increase in two angles and has about twice the value of the elevation's angle sensitivity. Since the maximum achievable efficiency is directly related to this parameter, these coefficients provide straightforward interpretation. For example, it can be estimated that the efficiency will be increased roughly by 10 % for each additional 25° of range in the Elevation angle.

8.2 Singularity Analysis

The RACOON-Lab's Rotator is equipped with five degrees of freedom which are used to provide seamless motion while avoiding the joint's limits. These two tasks are usually distributed between all the joints. Nevertheless, at certain positions some or several of these degrees of freedom can be lost. This may increase the workload of the remaining operative degrees of freedom, which can worsen their performance and limit their kinematic capabilities. Therefore, identifying these scenarios has a significant importance in order to understand recognize potential risky configurations. In this section, the possible singularities that can take place in the RACOON-Lab are identified and analyzed.

8.2.1 Singularity Definition

Singularities take place when the motion of a pair of joints can produce the same effect in the end effector. This means, that there is a linear dependency in the Jacobian tensor that maps their time derivatives to the end effector's. The practical effect of such scenario can be intuitively understood as having one or more joints repeated, with associated loss of dexterity.

In order to assess the existence of singularities, it is necessary to study the reduced 3x3 Jacobians. These quadratic tensors are obtained after removing two columns a, b , from the complete 3x5 Jacobian, so that a 3x3 matrix remains. Therefore, there is a total of 10 combinations which are summarized in Table 8-1.

Table 8-1: List of Reduced Jacobians for the RACOON-Lab

Removed Rows	Remaining Rows	Reduced Jacobian
α, β	$\gamma, \delta, \varepsilon$	$J_{\dot{E}q, \alpha\beta}^{\circ}$
α, γ	$\beta, \delta, \varepsilon$	$J_{\dot{E}q, \alpha\gamma}^{\circ}$
α, δ	$\beta, \gamma, \varepsilon$	$J_{\dot{E}q, \alpha\delta}^{\circ}$
α, ε	β, γ, δ	$J_{\dot{E}q, \alpha\varepsilon}^{\circ}$
β, γ	$\alpha, \delta, \varepsilon$	$J_{\dot{E}q, \beta\gamma}^{\circ}$
β, δ	$\alpha, \gamma, \varepsilon$	$J_{\dot{E}q, \beta\delta}^{\circ}$
β, ε	α, γ, δ	$J_{\dot{E}q, \beta\varepsilon}^{\circ}$
γ, δ	$\alpha, \beta, \varepsilon$	$J_{\dot{E}q, \gamma\delta}^{\circ}$
γ, ε	α, β, δ	$J_{\dot{E}q, \gamma\varepsilon}^{\circ}$
δ, ε	α, β, γ	$J_{\dot{E}q, \delta\varepsilon}^{\circ}$

In the notation used, $J_{\dot{E}q,ab}^{\circ}$ denotes the reduced Jacobian obtained after removing of q_a and q_b . With these definitions, it is now possible to state the condition for existence of a singularity:

$$\exists a, b \mid \det \left(J_{\dot{E}q,ab}^{\circ}(q) \right) = 0 \quad \text{Eq. (8-3)}$$

Therefore, the argument of the minimum distance to zero of the determinant of the reduced Jacobian can be used to assess the risk of a singularity:

$$\Gamma = \underset{a,b}{\operatorname{argmin}} \left\{ \left| \det \left(J_{\dot{E}q,ab}^{\circ}(q) \right) \right| \right\} \quad \text{Eq. (8-4)}$$

Hence, Γ approaching to zero is a strong indicator of singularity risk.

Due to the transformation order of the direct kinematics, only rotations around the X and Y axis as relevant for triggering singularities. Therefore, this parameter can be studied as a function of the elevation and Nick angles, thus $\Gamma = f(\beta, \gamma, \delta)$. Fig. 8-3 illustrates this function with some examples.

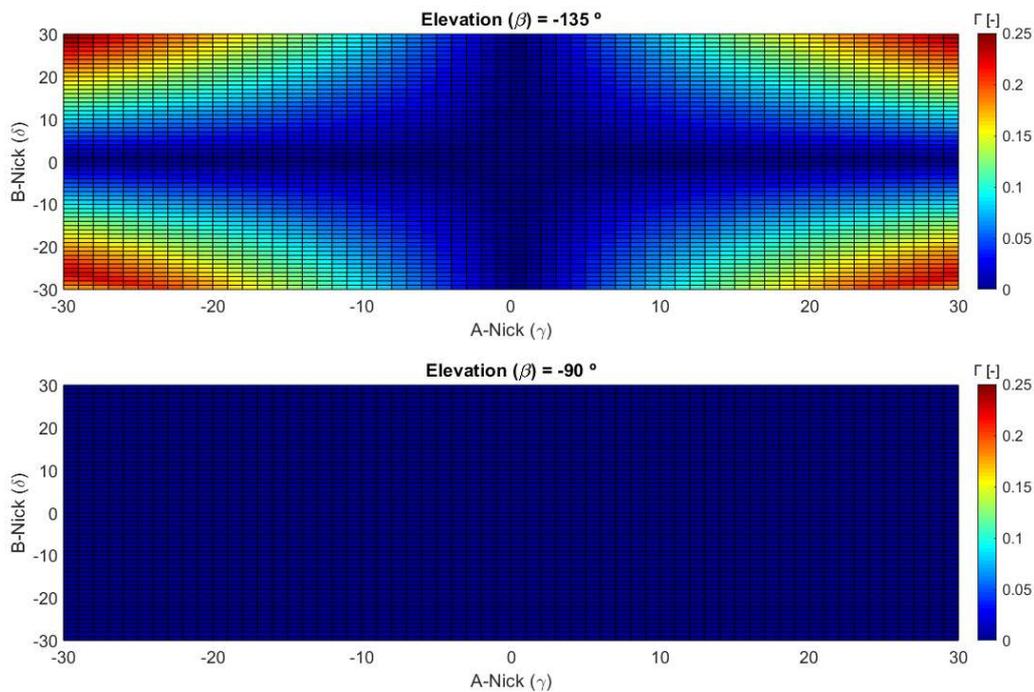


Fig. 8-3: Examples of the Argument of the Minimum Determinant for the Reduced Jacobian

The study of this function provides several relevant hints about the robot. The areas in which Γ presents a value close to zero have been checked manually in order to detect the singularities which can take place in the robot. These unfavorable scenarios are summarized in the next lines.

8.2.2 Singularity Identification

From the reduced Jacobians which exhibited a determinant close to zero, the involved degrees of freedom were checked in order to identify the linear dependencies. In Total, three possible singularity scenarios were found.

8.2.2.1 Azimuth-Polar Coupling

When the vertical axis of the end-effector and the inertial frame are aligned, the polar and azimuthal joints can provide the same effect as it is illustrated in Fig. 8-4.

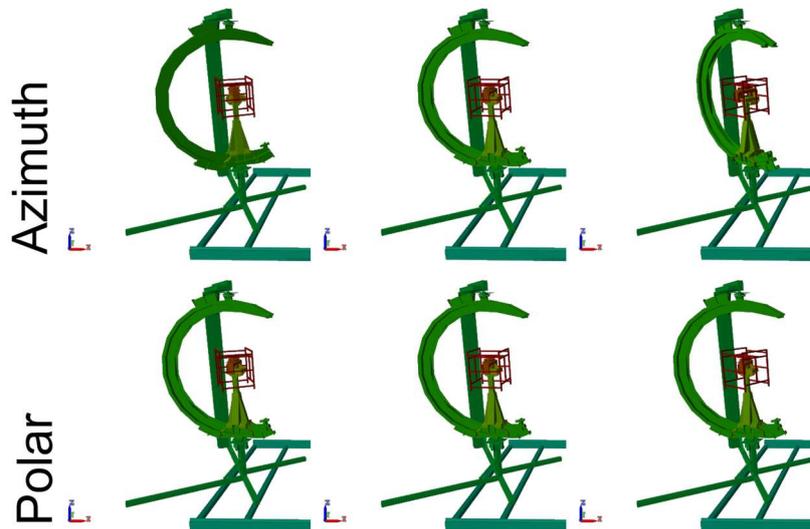


Fig. 8-4: Azimuth-Polar Coupling Singularity

This singularity can take place when the elevation angle is greater than -30° or smaller than -150° . In such cases, it is possible that certain combinations of the nick angles lead to coupling of α and ε . The presence of this singularity can be observed in the upper region of Fig. 8-5, where an elevation angle suitable to trigger this sort of singularities has been chosen.

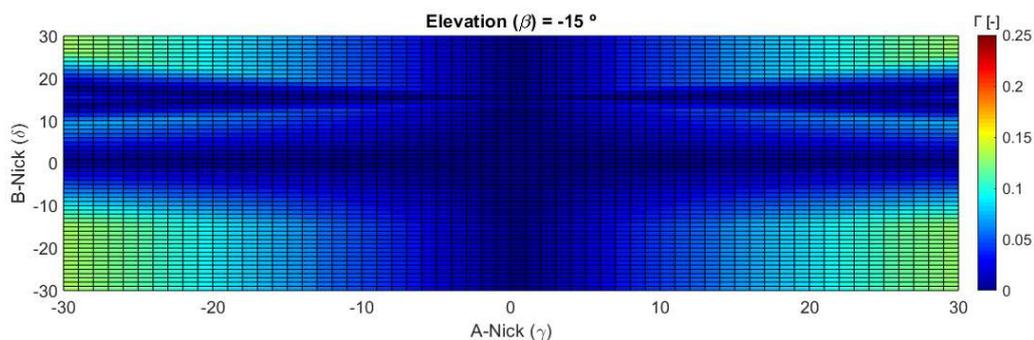


Fig. 8-5: Singularity Map with the Polar-Azimuth Coupling Visible in the Upper Area

8.2.2.2 Azimuth-A-Nick Coupling

The Azimuth A-Nick Coupling takes place in configurations where the elevation angle is a multiple of 90° . In such cases, the vertical axis of the end effector is contained in the horizontal plane and motions of the azimuthal joint have the same effect as the ones provided by the A-Nick as it can be visualized in Fig. 8-6.

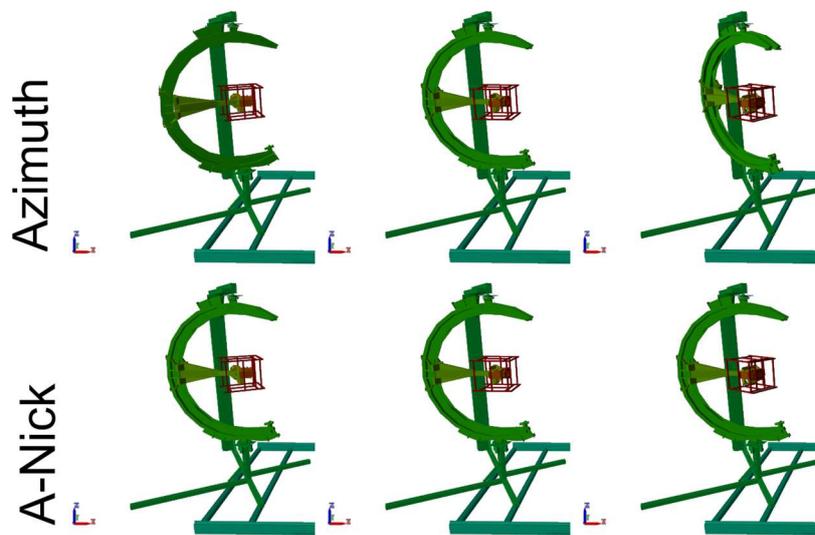


Fig. 8-6: Azimuth-A-Nick Coupling Singularity

8.2.2.3 Elevation-B-Nick Coupling

The Elevation B-Nick Coupling is the most common of all singularities. The reason for this is that it is associated to the cases in which one of the nick angles presents a value of zero. Since this configuration is often desired as it provides the optimal condition for joint limit avoidance, most of the trajectory algorithms tend to present such configurations. The effect of this coupling can be visualized in Fig. 8-7.

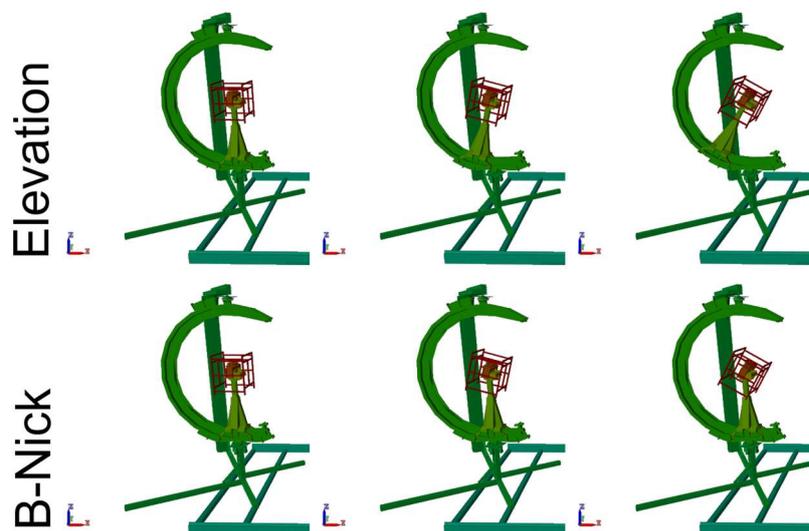


Fig. 8-7: Elevation-B-Nick Coupling Singularity

8.2.2.4 Multiple Singularity

Multiple singularities can take place whenever there is a combination of the three previously introduced singularities. In concrete, the first and the second singularities can appear in conjunction with the third one. For example, for the default configuration (all angles equal zero) there is simultaneously an Azimuth-Polar coupling and a

Elevation-B-Nick Coupling. Since the RACOON-Lab has only a degree of redundancy of two, these multiple singularities can pose a serious hindrance as they instantaneously remove the robot's redundancy. This implies, that instantaneously the robot can only fulfill one task (trajectory tracking) whilst there is no self-motion available for performing the joints' limits avoidance task.

8.2.3 Examples of Singularities Across a Trajectory

During the execution of a trajectory, the robot may undergo several singularities. In order to illustrate this, the determinants of the reduced Jacobians have been studied for roll rotation resulting with the ASC algorithm. This result is displayed in Fig. 8-8. In this graph, each curve represents the absolute value one of the ten reduced Jacobian's determinant. The labels indicate the remaining rows. Hence, $\gamma\delta\varepsilon$ for example corresponds to $\left| \det \left(J_{\dot{E}q, \alpha\beta}^o \right) \right|$.

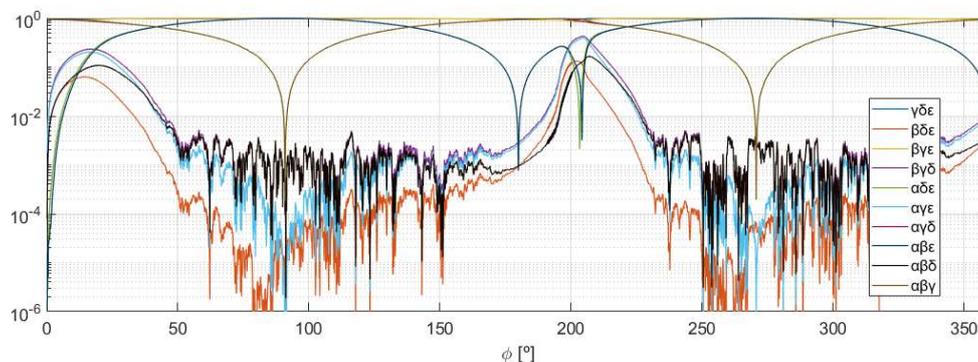


Fig. 8-8: Absolute Values of the Reduced Jacobians Across a Trajectory Performed with ASC

As it can be seen, during most of the trajectory the singularity of $\gamma - \beta$ coupling is present. This can be deduced from the fact that all the reduced Jacobians which contain these two terms present a determinant close to zero. The Polar-Azimuth singularity can be observed too by attending to the $\alpha\beta\varepsilon$ curve, which as it can be seen takes place only sporadically. The same behavior applies to the Azimuth-A Nick singularity, which can be tracked attending to the curve $\alpha\beta\delta$.

In the example of the roll rotation departing from the default configuration, the singularities do not pose any hindrance, besides the oscillations in the central part. Nevertheless, the loss of redundancy can be particularly critical for the use of conventional algorithms that solve the problem in the velocity domain. In these cases, there shall be at least one degree of freedom performing each task. However, if constrained degrees of freedom are involved, one of them may eventually become inoperative if a limit value is reached. In such cases, the remaining freedom of motion will be entirely used for following the trajectory without the possibility to avoid the joint's terminal positions.

In order to visualize this unfortunate chain of events a highly unfavorable scenario has been simulated. The case chosen is a rotation around the Y axis (pitch rotation) departing from the default configuration with the ASC algorithm. The result is displayed in Fig. 8-9.

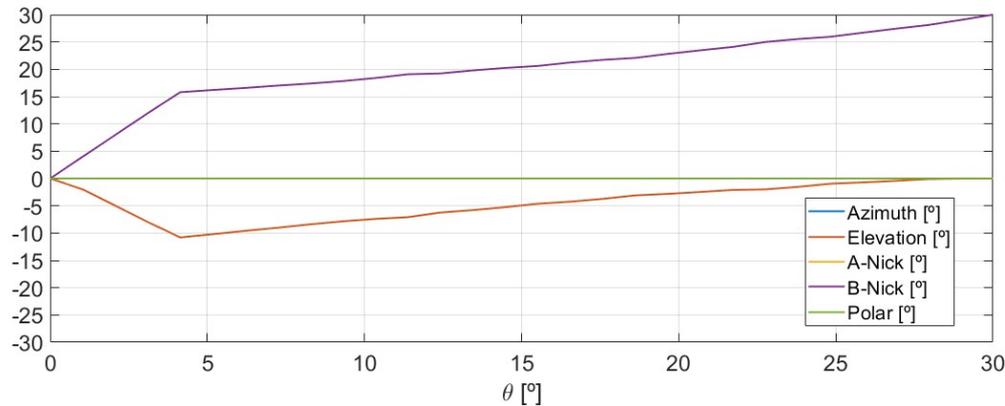


Fig. 8-9: Locking of ASC for Performing a Pitch Rotation

As it can be observed, the task of avoiding joint's limits is mainly performed by the elevation angle initially, but at certain point its terminal value is reached, and only the B-Nick angle can continue the motion, until it ends up in a deadlock. In this case, the convergence cannot be achieved regardless of the value of α_k , which only influences the point where the Elevation angle changes its slope. The same example was addressed with the IKCT algorithm. The resulting trajectory for the initial instants is displayed in Fig. 8-10. As it can be seen, in this case the deadlock can be avoided.

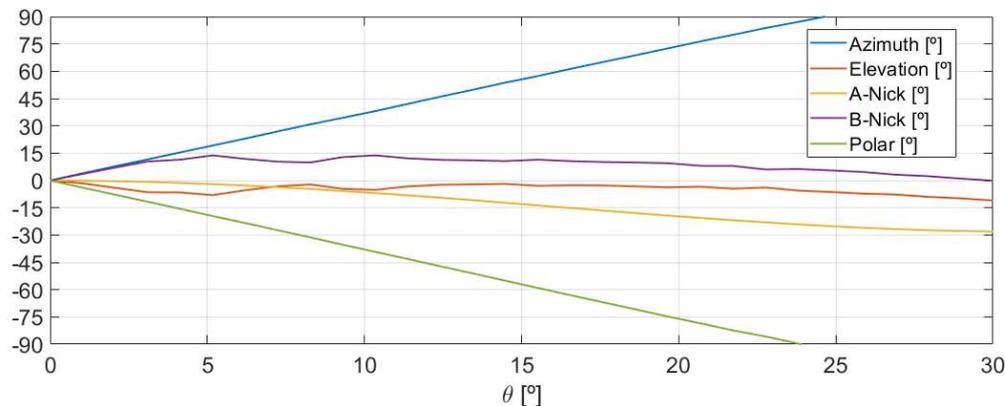


Fig. 8-10: RMPV Performing a Pitch Rotation from the Default Configuration

The reason for this is that the IKCT focuses on all degrees of freedom into tracking the manifold's centroid. This approach incorporates the joint limitation task in a more holistic fashion. In this case, this enables the avoidance of deadlocks as all the degrees of freedom are involved in both limit avoidance and trajectory tracking. Such strategy is more robust in scenarios when degrees of freedom are lost.

9 Practical Case

During the previous chapters, the solution of the inverse kinematics for both point and trajectory problems has been presented. However, the path examples which have been shown until this point were rather simple. Such trajectories are adequate for the study and validation of the inverse kinematics algorithms but hardly take place in the real world. In order to illustrate the applicability of the developed algorithms in a practical context, the motion of an actual orbiting satellite will be recreated using the elaborated schemes. The spacecraft used for this purpose will be MOVE-II, which was developed at the chair of Astronautics at the Technical University of Munich by the student society WARR (Wissenschaftliche Arbeitsgemeinschaft für Raketentechnik und Raumfahrt; Scientific Workgroup for Rocketry and Spaceflight).

During some stages of its early operation, this satellite tumbled in the absence of controlled actuation. The magnetic dipole moment of its solar cell wiring accelerated it to more than 100 deg/s. Its motion during that time strongly resembled that of uncooperative targets, which are the main examples that the rotator of the RACOON-Lab intends to represent in the future. In present chapter, the data collected by MOVE-II will be reconstructed and used for determining its attitude history. Subsequently, this attitude history will be used as an input trajectory for the trajectory algorithm and the configuration of the robot will be computed using the algorithms presented in Chapter 7. This will enable an assessment of the capabilities of the developed methodology under a realistic context.

9.1 MOVE-II Description

MOVE-II (Munich Orbital Verification Experiment II) is the second Satellite project developed at the Chair of Astronautics of the Technical University of Munich. It consists on two twin CubeSats: MOVE A and MOVE B which were launched at late 2018 and early 2019 respectively. A Computer Generated Graphic (CGI) of this spacecraft model can be visualized in Fig. 9-1.

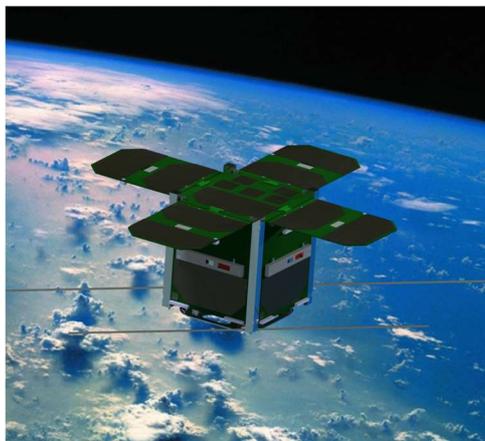


Fig. 9-1: Computer Generated Graphic (CGI) of MOVE-II with Deployed Flappanels [30]

MOVE-II is a CubeSat with a size of 10 cm x 10 cm x 10cm and a mass of 1.2 Kg [31]. This spacecraft's scientific payload requires an Attitude Determination and Control System (ADCS) in order to study the effect of the relative orientation with respect to

the Sun on the solar cell's performance. This motivated the requirement of a pointing accuracy of at least 10° , targeting 5° in MOVE-II's Critical Design Review (CDR) [32]. Such performance is achieved through the use of magnetic coils which interact with the Earth's magnetic field in order to generate a magnetic dipole [33]. One of the main tasks of this ADCS is to reduce the spinning rate of the spacecraft, as it would exhibit extremely high angular velocities otherwise. Nonetheless, during the first weeks of operation, due to the low budget in electronic power this actuation was not present at some points. This situation originated the afore stated high spinning rates. This kind of scenario resembles to the sort of behavior that the RACOON-Lab's rotator shall recreate in the future. Indeed, an unactuated free spinning satellite constitutes a very good example for the applications that will be researched within the RACOON-Lab. In this section, the motion of MOVE-II during a period of time in which it flew unactuated will be represented with the RACOON-Lab. In order to do this, some of the algorithms described in previous sections will be used. This exercise will serve as an initial validation of its applicability in a practical framework.

9.2 Attitude Data

The satellite is equipped with gyroscopes which record the angular rates in the body frame of the spacecraft. This is done by the small 9-axis sensor module BMX055 [34] that can be visualized in Fig. 9-2.



Fig. 9-2: BMX055 ADCS Sensor

Using the data collected by these instruments, it is possible to reconstruct the attitude history. The time frame in which the satellite's attitude will be study corresponds to May 13th from 10:04 to 10:14 Greenwich Mean Time (GMT).

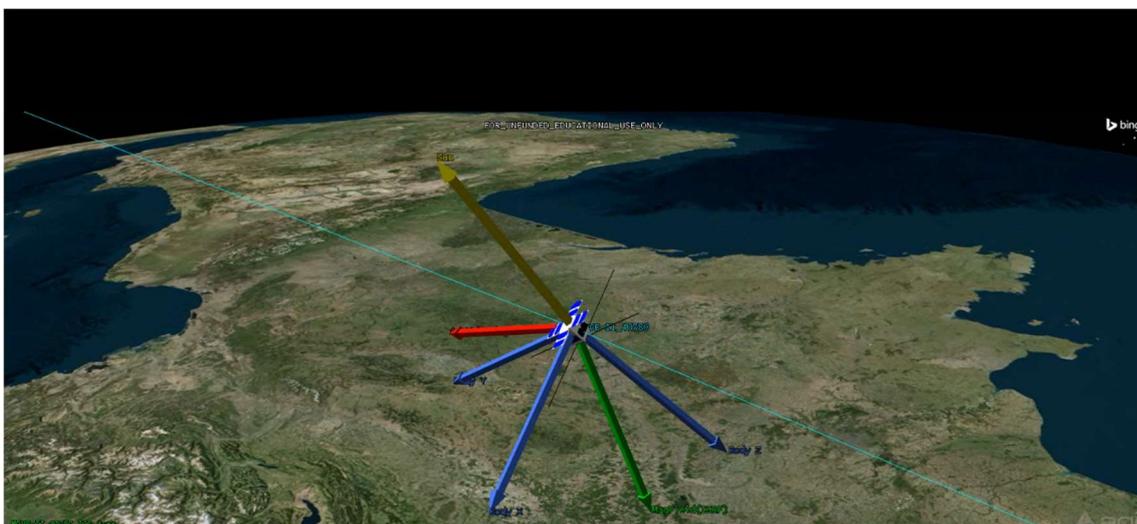


Fig. 9-3: Visualization of MOVE-II Orbiting on May 13th at 10:04 GMT

Unfortunately, the sensors' measurements are discontinuous and present several interruptions over time. Due to this, the continuous study of the trajectory over long periods of time requires the use of interpolation technics, which fill the missing data. The interpolation of the data has been performed considering the development in Fourier series with the available points. The resulting function has therefore the following formulation:

$$S(t) = \sum_{i=1}^n A_i \sin(2\pi \cdot i \cdot f_f \cdot t) + \sum_{i=1}^n B_i \cos(2\pi \cdot i \cdot f_f \cdot t) \quad \text{Eq. (9-1)}$$

The obtained fundamental frequency 34.75 μHz and the spectral relative energy can be visualized in Fig. 9-4.

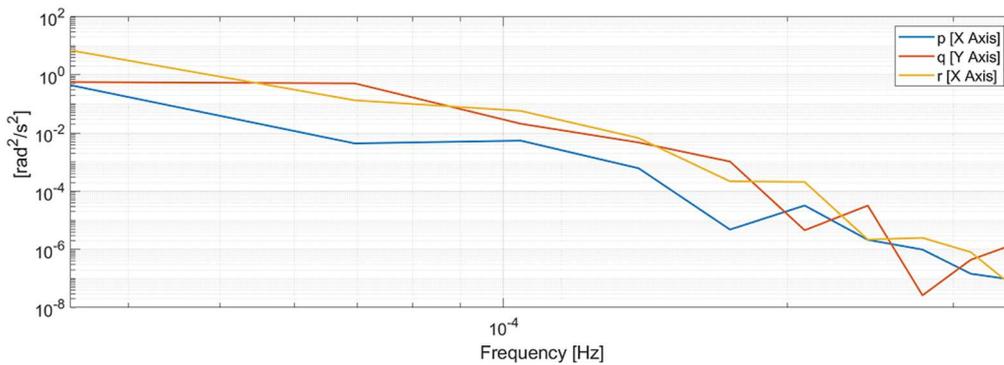


Fig. 9-4: Spectral Relative Energy of the Harmonics Obtained by the Regression

The resulting function of the angular rates and the measured data can be visualized in Fig. 9-5. As it can be seen, this sort of interpolation provides a more meaningful filling of the missing data compared to a linear method.

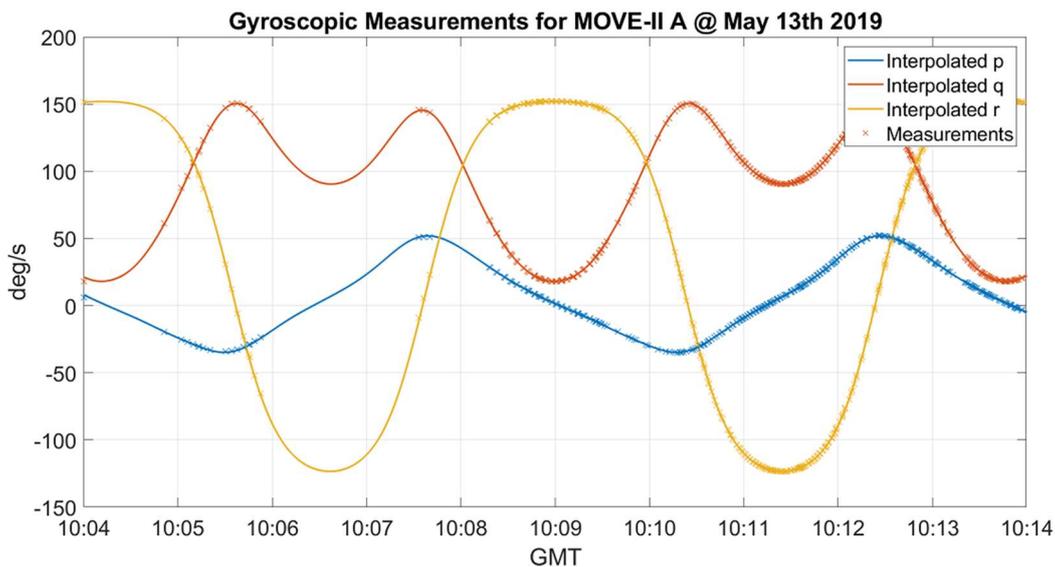


Fig. 9-5: Reconstruction of MOVE-II Gyroscopic Data

Once the values of p , q and r are available over time, it is possible to calculate the instantaneous change in attitude using the attitude transformation matrix:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad \text{Eq. (9-2)}$$

Assuming an initial attitude (e.g. $\phi_0 = \theta_0 = \psi_0 = 0^\circ$) it is possible to integrate with finite differences over time and thus obtain the attitude and attitude propagation history of the spacecraft. The attitude rates can be visualized in Fig. 9-6, which summarizes the numerical solution of the attitude equation. The reader is referred to the Appendix F.1 in order to find the complete figures which contain the information regarding the attitude history.

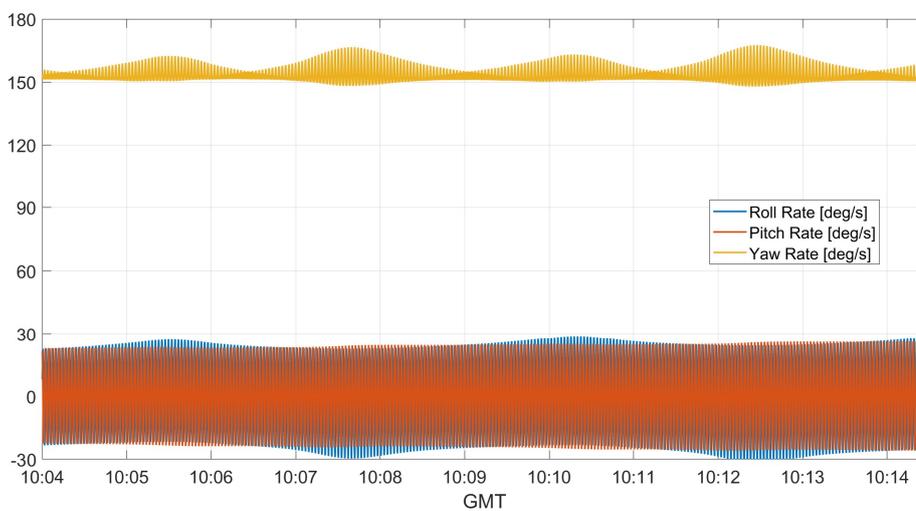


Fig. 9-6: Move II-A Euler Angles Rates over the Study Time

9.3 Attitude Simulation within the RACOON-Lab

With the computed attitude history as an input, it is possible to use the presented algorithms for obtaining the robot's configuration necessary to simulate the motion of MOVE-II. In this section both Automatic Supervisory Control and the Relaxed Isokinematic Centroid Tracking methods will be applied for this goal. The complete results of the algorithms used can be visualized in the Appendix F.2 of the present document. For the data display, the time of MOVE-II will be presented in the horizontal axis. However, it is important to remark that the time that the RACOON-Lab would need in order to achieve shall be higher due to its limitations in the maximum angular velocities.

9.3.1 Automatic Supervisory Control

Since the main motion experienced by the spacecraft is a yaw rotation, the roll rotation is unlikely to be the limiting factor during the initial moments. Therefore, it is possible to attempt lower values for α_k than the ones obtained in Chapter 7. Indeed, for simulating the first 10 seconds of MOVE-II a parameter of $\alpha_k=1$ is able to deliver a very good performance without exceeding the limits as it can be visualized in Fig. 9-7. In this case, the minimum efficiency is a 29.5% although it is around a 100% during most

of the trajectory. This can be seen by comparing Fig-F 9, which displays the angular velocities with Fig. 9-6.

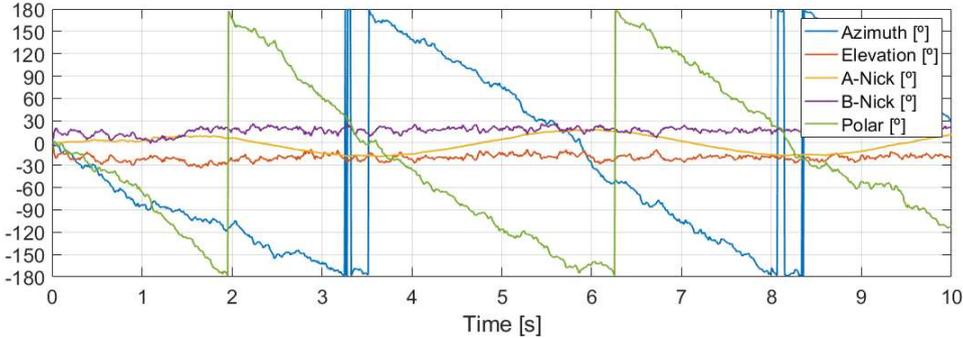


Fig. 9-7: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with ASC

In spite of this initial good performance, once the pitch angle becomes relevant the algorithm is not able to confine the motion within the joints' limits anymore as is displayed in Fig. 9-8. This behavior remains even with large values of the coefficient α_k .

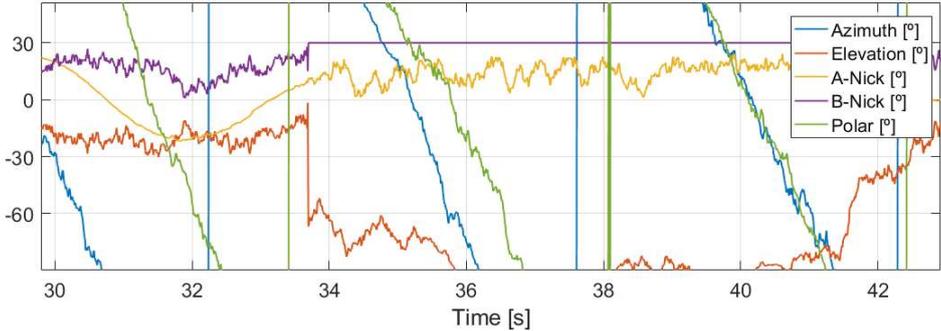


Fig. 9-8: Violation of Limits in Elevation and B-Nick with ASC and $\alpha_k=60$ s/rad

9.3.2 Relaxed Isokinematic Centroid Tracking

The relaxed isokinematic centroid tracking was tested starting with a high value for η_{min} (180%). This value was able to provide very efficient trajectories as it can be visualized in Fig. 9-9. As it can be seen, this highly efficient trajectory is possible due the fact that polar and azimuth angles work simultaneously.

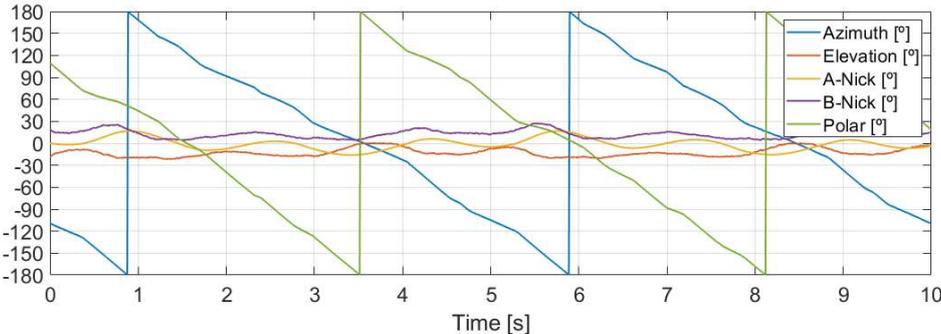


Fig. 9-9: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{min}=180\%$

However, as the spacecraft gains pitch angle, the azimuth and polar rotation axes are not aligned anymore. In such cases, highly efficient maneuvers are too slow to track the isokinematic centroid fast enough. This leads eventually to the robot cornered in a deadlock configuration.

Due to this, if one wants to represent the whole trajectory of MOVE-II it is necessary to use more conservative values η_{min} . These values produce a less efficient motion for the first moments as displayed in Fig. 9-10.

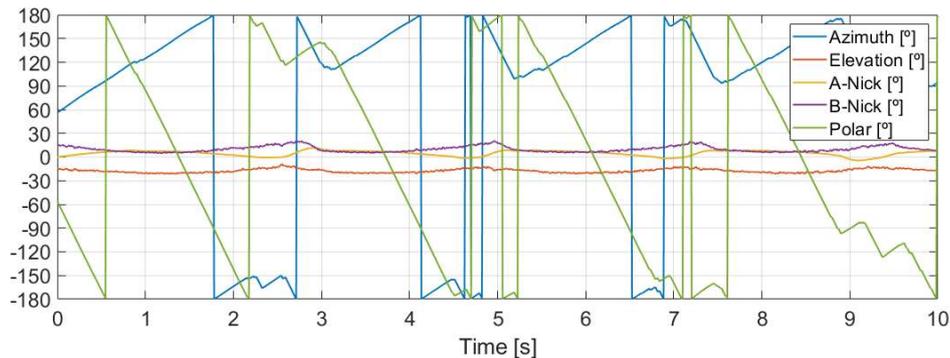


Fig. 9-10: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{min}=70\%$ (I)

The highest minimum efficiency η_{min} that was able to successfully complete the trajectory was 80%. This value is significantly higher than the one obtained for a roll rotation in Chapter 7. The reason for that is that the roll component of MOVE-II's rotation is very low compared to its yaw rate. Nevertheless, it can be observed that the resulting trajectory is much more inefficient. Indeed, in this case azimuth and roll do not work together but partially counteract. For example, at the first instant, a yaw rate of roughly 150 deg/s is achieved by a rate in the polar joint of approximately 220 deg/s complemented with -70 deg/s in the azimuthal joint.

This kind of inefficient movements is necessary in order to maintain the robot's configuration close enough to the isokinematic centroid. This is what allows the robot to keep up with the trajectory once the azimuth and polar angles are not aligned anymore as can be observed in Fig. 9-11.

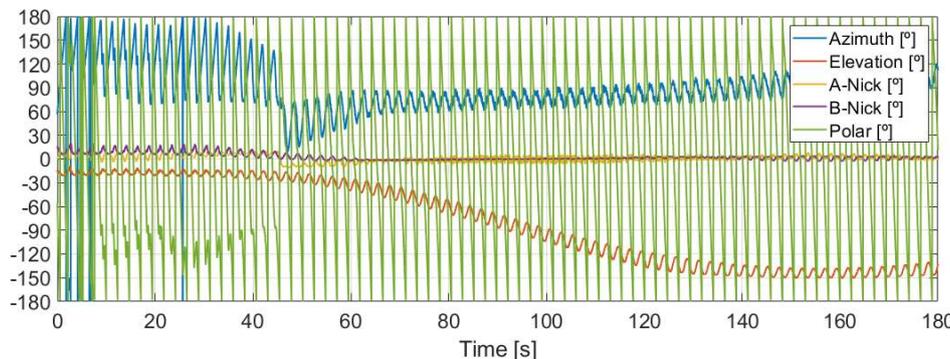


Fig. 9-11: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{min}=70\%$ (II)



This graph illustrates that after roughly two minutes, the elevation angle takes values below -90° , which indicates a high pitch angle. In such cases, polar and azimuth cannot work together anymore. Therefore, attempting for efficiencies over a 100% is not realistic anymore.

10 Discussion

The analytical approach of the inverse kinematics has proved able to provide a comprehensive overview of the robot's capabilities. Solving the problem from this perspective enhanced the understanding of the robot's limitations and their sensitivities to geometrical changes. It was possible to resort to this solution in order to study various aspects of the robots' kinematics such as the valid nick angles combinations or the theoretically maximum performance of the robot as a function of the joints.

The different algorithms presented in this work were tested against two different trajectories. The first one was a roll rotation, which can be considered as the most challenging motion due to the RACOON-Lab's Rotator's geometry. The other trajectory was the attitude of the satellite MOVE-II over a period of time in which tumbled in the absence of actuation.

The conventional algorithms, which solved the problem in the velocity domain were able to solve the roll rotation, but with a limited performance (10%) and the presence of oscillations. Automatic Supervisory Control (ASC) proved to be able to deliver a performance slightly better than the Weighted Least Norm method, but their general behavior was very similar. The oscillations could be removed with basic signal filtering methods. However, this filtering did not provide any significant improvement in efficiency terms and a position error was introduced. This error proved to decrease significantly as the cut-off frequency increased. In particular, the more heavily filtered trajectories presented a maximum attitude error close to 10° whilst the trajectories filtered with highest Cut-off frequencies had maximum errors lower than 1° . Whether such errors are acceptable or not depends on the considered application. Regarding the simulation of MOVE-II, ASC provided a much better performance, particularly for short periods of time when the spacecraft's motion resembled a pure yaw rotation.

The trajectories based on the analytical solution to the inverse kinematics proved able to outperform in the case of a roll rotation. Efficiencies in the range of a 30%-50% were achievable by these methodologies. For less challenging trajectories, where the motion was driven by the unconstrained joints, it was possible to deliver efficiencies over a 100%.

Regarding the complexity of both approaches, the conventional approach could deliver individual solutions with a mean time of $55.5 \mu\text{s}$ whilst the analytical solutions needed $272.5 \mu\text{s}$ in average. However, the solution in the velocity domain with other methods such as the Lagrange Multipliers should be much faster. This would further accentuate the differences in computational cost between the two families of algorithms.

11 Conclusions

The proposed algorithm has displayed a robust behavior in both point and trajectory problems. The best performing algorithm has proved to be the Relaxed Isokinematic Centroid Tracking (RIKCT), which was able to execute all the proposed trajectories with efficiencies ranging from 30% to 150%. The foundation of this algorithm is tracking the centroid of the isokinematic manifold, which is the point placed the furthest away from the robot's limit configurations. Therefore, this is a much more meaningful approach than the definition of performance functions used in conventional methods.

The analytical approach provides a more comprehensive overview of the robot's capabilities but it requires a significant initial effort for solving the mathematical problem. In the present case, the multibody presented five degrees of freedom, which is relatively low and enabled to handle of the problem's abstractness in a nearly intuitive way. The systematic approach presented to tackle the mathematical problem can be applied to any robot as far as it has at least as many revolute joints as degrees of freedom. Nevertheless, if the robot presents many more degrees of freedom it may become too abstract to be handled analytically. In such cases, identifying singularities such as the one found here for $\chi = \pm 90^\circ$ may become too challenging. Thanks to such identification of singularities the computational effort for performing the discretization over the isokinematic manifold can decrease significantly. If singularities remain undetected, a great deal of time can be lost in looking for isokinematic configurations where there is none. Conversely, a detected singularity eases the complexity of the problem as there is one less equation to solve. These findings prove the critical importance of being able to grasp the system's mathematical side, although the equations can be solved in a rather automatic fashion.

Another shortcoming of the analytical approaches is their computational effort. This sort of algorithms relies on determining the centroid of the isokinematic manifold. In order to determine this central configuration, it is necessary to perform a discretization, which must provide a representative sample of the hypergeometry. The necessary number of points for achieving an ergodic discretization may be high enough to hinder the algorithm's implementation on real time applications. A possible solution might be to perform a coarser discretization over the trajectory, using a lower number of waypoints. The interpolation between these points can be done using faster approaches such as Automatic Supervisory Control. This aspect may be of critical relevance when addressing the implementation of the analytical algorithms into embedded systems.

The analytical approach has proved to be a useful tool for assessing the capabilities of the robot's concept. In this direction, the isokinematic centroid gradient has proved to be the limiting factor in the robot's performance. Therefore, its study as a function of the geometrical limitations provides invaluable information for about the suitability of the robot's concept. This kind of studies could be used in the future in order to assess the goodness of a given multibody for executing certain task.

Finally, the study of the robot's performance over simpler trajectories provides a broader overview of the improvements brought by the analytical approach. In the case of the MOVE-II rotation, it was observed that the performance of the conventional methods is much better for the cases where the robot has enough redundancy. In such



cases, the analytical algorithms were able to deliver an improvement, but it was of a lower magnitude. Therefore, the algorithms generated by the analytical approach can provide remarkable improvements for conditions in which the spectrum of feasible trajectories is excessively narrow. Such is the case of a roll rotation in the RACOON-Lab. For these scenarios, the performance criteria used in conventional methods seemed to be unsuited for representing the optimality of a configuration. Conversely, if there is a higher availability of attainable trajectories, these functions provide an acceptable assessment.

12 Outlook

The future work derived from the present project's output can be divided in two main groups: The implementation at the actual facility and the application of the proposed algorithm in other robots.

Regarding the implementation on the RACOON-Lab, it will be necessary to program the presented algorithms in a suitable way, which enables the dynamic execution of trajectories. This task demands to account for factors such as latency and computational time. In order to address this issue successfully it might be necessary to assess the trade-offs between accuracy and computational effort. In the present project, the resolution used in the trajectories examples has been 1 deg in Euler angles for the sake of accuracy. However, coarser resolutions may imply negligible errors while being significantly more efficient from a computational standpoint. Furthermore, the connection between the points obtained from the analytical solution of the inverse kinematics could be done using conventional algorithms such as ASC or WLN. These methodologies shall deliver a good performance as far as they restrain to connect points which are within the joint's limits. Another key fact in the implementation at the actual facility is the addition of more convolute kinematic constraints. Within the scope of the present work, only the limitations in the joints' limits and maximum angular velocity have been considered. Nevertheless, other restrains such as maximum combined angular velocities must be taken into account. For example, practical experience has shown problems when high angular velocities of both the Elevation angle and B-Nick have been demanded simultaneously, although its individual values were under its theoretical limit. Identifying and quantifying these additional constrains constitutes itself a necessary work for the future of the RACOON-Lab.

The proposed methodology for determining the inverse kinematics' manifold can be applied to many other robots. In general, it is theoretically possible to adapt the presented procedure to any robot which has at least as many revolute joints as degrees of freedom in space. However, the computation of the solution may face several challenges, that could not be foreseen in the present robot due to its relative simplicity. In this direction, the visualization of the manifolds may become excessively abstract as the robot's degrees of freedom increase. Another challenging aspect might be the definition of the centroid of solutions to the inverse kinematics manifold if multiple unconnected isokinematic regions appear during the execution of a trajectory.

A References

- [1] C. P. Mark and S. Kamath, "Review of Active Space Debris Removal Methods," *Space Policy*, vol. 47, pp. 194-206, 2019.
- [2] F. Aschenbrenner, Mathematische Beschreibung der Kinematik des RACOON-Lab und Implementierung einer Inversen Kinematik als Softwaremodul, 2017.
- [3] D. DeMers and K. Kreuth-Delgado, *Neural Systems for Robotics*, 1997.
- [4] D. Rixen, Skript zur Vorlesung Roboterdynamik, © Lehrstuhl für Angewandte Mechanik, 2018.
- [5] D. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Transactions on Man Machine Systems*, vol. 10, no. 2, pp. 47-53, 1969.
- [6] D. N. Nenchev, A. Konno and T. Tsujita, *Humanoid Robots*, Elsevier Inc, 2019.
- [7] Y. Umetani and K. Yoshida, "Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 303-314, 1989.
- [8] A. Liégois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 868-871, 1977.
- [9] J. Euler, R. Dubey, S. Babcock and W. Hamel, "A comparison of two real-time control schemes for redundant manipulators with bounded joint velocities," in *International Conference on Robotics and Automation*, Scottsdale, 1989.
- [10] T. F. Chan and R. V. Dubey, "A Weighted Least-Norm Solution Based Scheme for Avoiding Joint Limits for Redundant Joint Manipulators," *Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 286-292, 1995.
- [11] H.-Y. Lee, B.-J. Yi and Y. Choi, "A Realistic Joint Limit Algorithm for Kinematically Redundant Manipulators," in *International Conference on Control, Automation and Systems*, Seoul, 2007.
- [12] M. Wang, *Constrained Model Predictive Control for Real-Time Tele-Operation Motion Planning*, 2015.
- [13] D. A. Drexler and I. Harmati, "Joint Constrained Differential Inverse Kinematics Algorithm for Serial Manipulators," *Electrical Engineering and Computer Science*, vol. 56, no. 4, pp. 95-104, 2012.
- [14] J.-D. Sun, G.-Z. Cao, W.-B. Li, Y.-X. Liang and S.-D. Huang, "Analytical inverse kinematic solution using the D-H method for a 6-DOF robot," in *International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Jeju, South Korea, 2017.

- [15] M. Brandstötter, A. Angerer and M. Hofbauer, "An Analytical Solution of the Inverse Kinematics Problem of Industrial Serial Manipulators with an Ortho-parallel Basis and a Spherical Wrist," in *Proceedings of the Austrian Robotics Workshop*, Linz, Austria, 2014.
- [16] C. L. Lück and S. Lee, "Self-Motion Topology for Redundant Manipulators with Joint Limits," in *International Conference on Robotics*, Atlanta, 1999.
- [17] C. Lück, "Self-Motion Representation and Global Path Planning Optimization for Redundant Manipulators through Topology-Based Discretization," *Journal of Intelligent and Robotic Systems*, vol. 19, pp. 23-38, 1997.
- [18] J. Harder, M. Dziura and S. Haberl, "Future Technologies for Operating Robots in Space," in *International Astronautical Congress*, Adelaide, 2017.
- [19] European Space Agency, "European Space Agency," [Online]. Available: https://www.esa.int/ESA_Multimedia/Images/2007/03/Artist_s_impression_showing_ATV_docking_with_ISS. [Accessed 17 05 2020].
- [20] DLR. [Online]. Available: https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-3827/5969_read-8759/. [Accessed 28 4 2020].
- [21] S. Sharma, C. Beierle and S. Dámico, "Pose Estimation for Non-Cooperative Spacecraft Rendezvous Using Convolutional Neural Networks," in *2018 IEEE , Big Sky, Montana*, 2018.
- [22] J. Ventura, *Autonomous Proximity Operations for Noncooperative Space Target*, 2016.
- [23] F. Pfeiffer and E. Reithmayer, "Roboterdynamik: Eine Einführung in die Grundlagen und technischen Anwendungen," 1987.
- [24] "CH Robotics," [Online]. Available: <http://www.chrobotics.com/library/understanding-euler-angles>. [Accessed 2 5 2020].
- [25] D. N. Nenchev, "Reaction Null Space of a Multibody System with Applications in Robotics," *Mechanical Science*, vol. 4, pp. 97-112, 2013.
- [26] J. M. Reza, *Theory of Applied Robotics*, 2010.
- [27] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, *Robotics, Modelling, Planning and Control*, Springer, 2009.
- [28] W. Madych, "Solutions of underdetermined systems of linear equations," 1991.
- [29] H. Zghal, V. Dubey and J. Euler, "Efficient Gradient Projection Optimization for Manipulators with Multiple Degrees of Redundancy," in *IEEE International Conference of Robotics and Automation*, Cincinnati, 1990.
- [30] N. Appel, S. Ruckerl and M. Langer, "A Robust and Efficient Protocol for Small Satellite Radio Links," Malta, 2016.

- [31] M. Langer, N. Appel, M. Dziura, C. Fuchs, P. Günzel, J. Gutmiedl, M. Losekamm, P. T. Meßmann and C. Trinitis, "MOVE-II-der Zweite Kleinsatellit der Technischen Universität München," in *Deutscher Luft- und Raumfahrtkongress (DLRK)*, Rostock, 2015.
- [32] D. Meßmann, J. Brügge, M. Cordovi, G. T. Dötterl, F. Mauracher, A. Meraner, T. Ohlenforst, S. Plamauer, L. Schmidt, P. Schnierle, M. Seifert and N. Weeter, "MOVE-II Critical Design Review (CDR) ADCS," 2016.
- [33] D. Meßmann, F. Coelho, P. Niermeyer, M. Langer, H. Huang and U. Walter, "7th European Conference for Aeronautics and Aerospace Sciences (EUCASS)," Milan, 2017.
- [34] Bosch Sensortec, "BMX055 Small, versatile 9-axis sensor Module," 2014.
- [35] "NASA," [Online]. Available: <https://www.nasa.gov/sites/default/files/thumbnails/image/iss031e070790.jpg>. [Accessed 28 4 2020].

B Jacobian Tensor

The Jacobian Tensor $J_{\dot{E}q}$ depends only on the current robot's configuration q contains the sensitivity coefficients that relate the configuration's velocities \dot{q} to the angular rates in the end effector's frame. The relationship between these elements can be expressed as follows:

$$\begin{pmatrix} p_E \\ q_E \\ r_E \end{pmatrix} = J_{\dot{E}q}(q)\dot{q} = \begin{pmatrix} J_{\dot{E}q}(1,1) & J_{\dot{E}q}(1,2) & J_{\dot{E}q}(1,2) & J_{\dot{E}q}(1,2) & J_{\dot{E}q}(1,2) \\ J_{\dot{E}q}(2,1) & J_{\dot{E}q}(2,2) & J_{\dot{E}q}(2,3) & J_{\dot{E}q}(2,4) & J_{\dot{E}q}(2,5) \\ J_{\dot{E}q}(3,1) & J_{\dot{E}q}(3,2) & J_{\dot{E}q}(3,3) & J_{\dot{E}q}(3,4) & J_{\dot{E}q}(3,5) \end{pmatrix} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \\ \dot{\delta} \\ \dot{\epsilon} \end{pmatrix} \quad \text{Eq. (B-1)}$$

With:

$$\begin{aligned} J_{\dot{E}q}(1,1) = & (\cos(\epsilon)(\cos(\gamma)\sin(\alpha) + \cos(\alpha)\sin(\beta)\sin(\gamma)) \\ & + \sin(\epsilon)(\sin(\delta)(\sin(\alpha)\sin(\gamma) - \cos(\alpha)\cos(\gamma)\sin(\beta)) \\ & + \cos(\alpha)\cos(\beta)\cos(\delta))(\cos(\delta)(\cos(\alpha)\sin(\gamma) + \cos(\gamma)\sin(\alpha)\sin(\beta)) \\ & + \cos(\beta)\sin(\alpha)\sin(\delta)) - (\cos(\epsilon)(\cos(\alpha)\cos(\gamma) \\ & - \sin(\alpha)\sin(\beta)\sin(\gamma)) + \sin(\epsilon)(\sin(\delta)(\cos(\alpha)\sin(\gamma) \\ & + \cos(\gamma)\sin(\alpha)\sin(\beta)) - \cos(\beta)\cos(\delta)\sin(\alpha))(\cos(\delta)(\sin(\alpha)\sin(\gamma) \\ & - \cos(\alpha)\cos(\gamma)\sin(\beta)) - \cos(\alpha)\cos(\beta)\sin(\delta)) \end{aligned}$$

$$\begin{aligned} J_{\dot{E}q}(2,1) = & (\sin(\epsilon)(\cos(\gamma)\sin(\alpha) + \cos(\alpha)\sin(\beta)\sin(\gamma)) \\ & - \cos(\epsilon)(\sin(\delta)(\sin(\alpha)\sin(\gamma) - \cos(\alpha)\cos(\gamma)\sin(\beta)) \\ & + \cos(\alpha)\cos(\beta)\cos(\delta))(\cos(\delta)(\cos(\alpha)\sin(\gamma) + \cos(\gamma)\sin(\alpha)\sin(\beta)) \\ & + \cos(\beta)\sin(\alpha)\sin(\delta)) - (\sin(\epsilon)(\cos(\alpha)\cos(\gamma) \\ & - \sin(\alpha)\sin(\beta)\sin(\gamma)) - \cos(\epsilon)(\sin(\delta)(\cos(\alpha)\sin(\gamma) \\ & + \cos(\gamma)\sin(\alpha)\sin(\beta)) - \cos(\beta)\cos(\delta)\sin(\alpha))(\cos(\delta)(\sin(\alpha)\sin(\gamma) \\ & - \cos(\alpha)\cos(\gamma)\sin(\beta)) - \cos(\alpha)\cos(\beta)\sin(\delta)) \end{aligned}$$

$$\begin{aligned} J_{\dot{E}q}(3,1) = & (\cos(\epsilon)(\cos(\gamma)\sin(\alpha) + \cos(\alpha)\sin(\beta)\sin(\gamma)) \\ & + \sin(\epsilon)(\sin(\delta)(\sin(\alpha)\sin(\gamma) - \cos(\alpha)\cos(\gamma)\sin(\beta)) \\ & + \cos(\alpha)\cos(\beta)\cos(\delta))(\sin(\epsilon)(\cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\beta)\sin(\gamma)) \\ & - \cos(\epsilon)(\sin(\delta)(\cos(\alpha)\sin(\gamma) + \cos(\gamma)\sin(\alpha)\sin(\beta)) \\ & - \cos(\beta)\cos(\delta)\sin(\alpha)) - (\sin(\epsilon)(\cos(\gamma)\sin(\alpha) \\ & + \cos(\alpha)\sin(\beta)\sin(\gamma)) - \cos(\epsilon)(\sin(\delta)(\sin(\alpha)\sin(\gamma) \\ & - \cos(\alpha)\cos(\gamma)\sin(\beta)) + \cos(\alpha)\cos(\beta)\cos(\delta))(\cos(\epsilon)(\cos(\alpha)\cos(\gamma) \\ & - \sin(\alpha)\sin(\beta)\sin(\gamma)) + \sin(\epsilon)(\sin(\delta)(\cos(\alpha)\sin(\gamma) \\ & + \cos(\gamma)\sin(\alpha)\sin(\beta)) - \cos(\beta)\cos(\delta)\sin(\alpha)) \end{aligned}$$

$$\begin{aligned} J_{\dot{E}q}(1,2) = & (\cos(\epsilon)(\cos(\gamma)\sin(\alpha) + \cos(\alpha)\sin(\beta)\sin(\gamma)) \\ & + \sin(\epsilon)(\sin(\delta)(\sin(\alpha)\sin(\gamma) - \cos(\alpha)\cos(\gamma)\sin(\beta)) \\ & + \cos(\alpha)\cos(\beta)\cos(\delta))(\cos(\alpha)\sin(\beta)\sin(\delta) \\ & - \cos(\alpha)\cos(\beta)\cos(\gamma)\cos(\delta)) - (\cos(\epsilon)(\cos(\alpha)\cos(\gamma) \\ & - \sin(\alpha)\sin(\beta)\sin(\gamma)) + \sin(\epsilon)(\sin(\delta)(\cos(\alpha)\sin(\gamma) \\ & + \cos(\gamma)\sin(\alpha)\sin(\beta)) - \cos(\beta)\cos(\delta)\sin(\alpha))(\sin(\alpha)\sin(\beta)\sin(\delta) \\ & - \cos(\beta)\cos(\gamma)\cos(\delta)\sin(\alpha)) - (\cos(\beta)\sin(\delta) \\ & + \cos(\gamma)\cos(\delta)\sin(\beta))(\sin(\epsilon)(\cos(\delta)\sin(\beta) + \cos(\beta)\cos(\gamma)\sin(\delta)) \\ & - \cos(\beta)\cos(\epsilon)\sin(\gamma)) \end{aligned}$$

$$\begin{aligned}
J_{\dot{E}q}(2,2) = & (\cos(\delta) (\cos(\alpha) \sin(\gamma) + \cos(\gamma) \sin(\alpha) \sin(\beta)) \\
& + \cos(\beta) \sin(\alpha) \sin(\delta)) (\cos(\varepsilon) (\cos(\delta) \sin(\alpha) \sin(\beta) \\
& + \cos(\beta) \cos(\gamma) \sin(\alpha) \sin(\delta)) + \cos(\beta) \sin(\alpha) \sin(\gamma) \sin(\varepsilon)) \\
& - (\cos(\delta) (\sin(\alpha) \sin(\gamma) - \cos(\alpha) \cos(\gamma) \sin(\beta)) \\
& - \cos(\alpha) \cos(\beta) \sin(\delta)) (\cos(\varepsilon) (\cos(\alpha) \cos(\delta) \sin(\beta) \\
& + \cos(\alpha) \cos(\beta) \cos(\gamma) \sin(\delta)) + \cos(\alpha) \cos(\beta) \sin(\gamma) \sin(\varepsilon)) \\
& - (\sin(\beta) \sin(\delta) \\
& - \cos(\beta) \cos(\gamma) \cos(\delta)) (\cos(\varepsilon) (\cos(\beta) \cos(\delta) - \cos(\gamma) \sin(\beta) \sin(\delta)) \\
& - \sin(\beta) \sin(\gamma) \sin(\varepsilon))
\end{aligned}$$

$$\begin{aligned}
J_{\dot{E}q}(3,2) = & (\cos(\varepsilon) (\cos(\delta) \sin(\beta) + \cos(\beta) \cos(\gamma) \sin(\delta)) \\
& + \cos(\beta) \sin(\gamma) \sin(\varepsilon)) (\sin(\varepsilon) (\cos(\beta) \cos(\delta) - \cos(\gamma) \sin(\beta) \sin(\delta)) \\
& + \cos(\varepsilon) \sin(\beta) \sin(\gamma)) + (\sin(\varepsilon) (\cos(\gamma) \sin(\alpha) + \cos(\alpha) \sin(\beta) \sin(\gamma)) \\
& - \cos(\varepsilon) (\sin(\delta) (\sin(\alpha) \sin(\gamma) - \cos(\alpha) \cos(\gamma) \sin(\beta)) \\
& + \cos(\alpha) \cos(\beta) \cos(\delta))) (\sin(\varepsilon) (\cos(\alpha) \cos(\delta) \sin(\beta) \\
& + \cos(\alpha) \cos(\beta) \cos(\gamma) \sin(\delta)) - \cos(\alpha) \cos(\beta) \cos(\varepsilon) \sin(\gamma)) \\
& - (\sin(\varepsilon) (\cos(\alpha) \cos(\gamma) - \sin(\alpha) \sin(\beta) \sin(\gamma)) \\
& - \cos(\varepsilon) (\sin(\delta) (\cos(\alpha) \sin(\gamma) + \cos(\gamma) \sin(\alpha) \sin(\beta)) \\
& - \cos(\beta) \cos(\delta) \sin(\alpha))) (\sin(\varepsilon) (\cos(\delta) \sin(\alpha) \sin(\beta) \\
& + \cos(\beta) \cos(\gamma) \sin(\alpha) \sin(\delta)) - \cos(\beta) \cos(\varepsilon) \sin(\alpha) \sin(\gamma))
\end{aligned}$$

$$\begin{aligned}
J_{\dot{E}q}(1,3) = & \cos(\delta) (\cos(\gamma) \sin(\alpha) \\
& + \cos(\alpha) \sin(\beta) \sin(\gamma)) (\cos(\varepsilon) (\cos(\gamma) \sin(\alpha) + \cos(\alpha) \sin(\beta) \sin(\gamma)) \\
& + \sin(\varepsilon) (\sin(\delta) (\sin(\alpha) \sin(\gamma) - \cos(\alpha) \cos(\gamma) \sin(\beta)) \\
& + \cos(\alpha) \cos(\beta) \cos(\delta))) \\
& + \cos(\delta) (\cos(\alpha) \cos(\gamma) \\
& - \sin(\alpha) \sin(\beta) \sin(\gamma)) (\cos(\varepsilon) (\cos(\alpha) \cos(\gamma) - \sin(\alpha) \sin(\beta) \sin(\gamma)) \\
& + \sin(\varepsilon) (\sin(\delta) (\cos(\alpha) \sin(\gamma) + \cos(\gamma) \sin(\alpha) \sin(\beta)) \\
& - \cos(\beta) \cos(\delta) \sin(\alpha))) \\
& - \cos(\beta) \cos(\delta) \sin(\gamma) (\sin(\varepsilon) (\cos(\delta) \sin(\beta) + \cos(\beta) \cos(\gamma) \sin(\delta)) \\
& - \cos(\beta) \cos(\varepsilon) \sin(\gamma))
\end{aligned}$$

$$\begin{aligned}
J_{\dot{E}q}(2,3) = & (\cos(\delta) (\sin(\alpha) \sin(\gamma) - \cos(\alpha) \cos(\gamma) \sin(\beta)) \\
& - \cos(\alpha) \cos(\beta) \sin(\delta)) (\sin(\varepsilon) (\sin(\alpha) \sin(\gamma) - \cos(\alpha) \cos(\gamma) \sin(\beta)) \\
& + \cos(\varepsilon) \sin(\delta) (\cos(\gamma) \sin(\alpha) + \cos(\alpha) \sin(\beta) \sin(\gamma))) \\
& - (\sin(\beta) \sin(\delta) - \cos(\beta) \cos(\gamma) \cos(\delta)) (\cos(\beta) \cos(\gamma) \sin(\varepsilon) \\
& - \cos(\beta) \cos(\varepsilon) \sin(\gamma) \sin(\delta)) + (\cos(\delta) (\cos(\alpha) \sin(\gamma) \\
& + \cos(\gamma) \sin(\alpha) \sin(\beta)) + \cos(\beta) \sin(\alpha) \sin(\delta)) (\sin(\varepsilon) (\cos(\alpha) \sin(\gamma) \\
& + \cos(\gamma) \sin(\alpha) \sin(\beta)) + \cos(\varepsilon) \sin(\delta) (\cos(\alpha) \cos(\gamma) \\
& - \sin(\alpha) \sin(\beta) \sin(\gamma)))
\end{aligned}$$

$$\begin{aligned}
J_{\dot{E}q}(3,3) = & (\sin(\varepsilon)(\cos(\gamma)\sin(\alpha) + \cos(\alpha)\sin(\beta)\sin(\gamma)) \\
& - \cos(\varepsilon)(\sin(\delta)(\sin(\alpha)\sin(\gamma) - \cos(\alpha)\cos(\gamma)\sin(\beta)) \\
& + \cos(\alpha)\cos(\beta)\cos(\delta))(\cos(\varepsilon)(\sin(\alpha)\sin(\gamma) - \cos(\alpha)\cos(\gamma)\sin(\beta)) \\
& - \sin(\delta)\sin(\varepsilon)(\cos(\gamma)\sin(\alpha) + \cos(\alpha)\sin(\beta)\sin(\gamma))) \\
& - (\cos(\varepsilon)(\cos(\delta)\sin(\beta) + \cos(\beta)\cos(\gamma)\sin(\delta)) \\
& + \cos(\beta)\sin(\gamma)\sin(\varepsilon))(\cos(\beta)\cos(\gamma)\cos(\varepsilon) \\
& + \cos(\beta)\sin(\gamma)\sin(\delta)\sin(\varepsilon)) + (\sin(\varepsilon)(\cos(\alpha)\cos(\gamma) \\
& - \sin(\alpha)\sin(\beta)\sin(\gamma)) - \cos(\varepsilon)(\sin(\delta)(\cos(\alpha)\sin(\gamma) \\
& + \cos(\gamma)\sin(\alpha)\sin(\beta)) - \cos(\beta)\cos(\delta)\sin(\alpha)))(\cos(\varepsilon)(\cos(\alpha)\sin(\gamma) \\
& + \cos(\gamma)\sin(\alpha)\sin(\beta)) - \sin(\delta)\sin(\varepsilon)(\cos(\alpha)\cos(\gamma) \\
& - \sin(\alpha)\sin(\beta)\sin(\gamma)))
\end{aligned}$$

$$\begin{aligned}
J_{\dot{E}q}(1,4) = & -(\cos(\varepsilon)(\cos(\gamma)\sin(\alpha) + \cos(\alpha)\sin(\beta)\sin(\gamma)) \\
& + \sin(\varepsilon)(\sin(\delta)(\sin(\alpha)\sin(\gamma) - \cos(\alpha)\cos(\gamma)\sin(\beta)) \\
& + \cos(\alpha)\cos(\beta)\cos(\delta))(\sin(\delta)(\sin(\alpha)\sin(\gamma) - \cos(\alpha)\cos(\gamma)\sin(\beta)) \\
& + \cos(\alpha)\cos(\beta)\cos(\delta)) - (\cos(\varepsilon)(\cos(\alpha)\cos(\gamma) \\
& - \sin(\alpha)\sin(\beta)\sin(\gamma)) + \sin(\varepsilon)(\sin(\delta)(\cos(\alpha)\sin(\gamma) \\
& + \cos(\gamma)\sin(\alpha)\sin(\beta)) - \cos(\beta)\cos(\delta)\sin(\alpha)))(\sin(\delta)(\cos(\alpha)\sin(\gamma) \\
& + \cos(\gamma)\sin(\alpha)\sin(\beta)) - \cos(\beta)\cos(\delta)\sin(\alpha)) - (\cos(\delta)\sin(\beta) \\
& + \cos(\beta)\cos(\gamma)\sin(\delta))(\sin(\varepsilon)(\cos(\delta)\sin(\beta) + \cos(\beta)\cos(\gamma)\sin(\delta)) \\
& - \cos(\beta)\cos(\varepsilon)\sin(\gamma))
\end{aligned}$$

$$\begin{aligned}
J_{\dot{E}q}(2,4) = & \cos(\varepsilon) \left(\cos(\delta)(\cos(\alpha)\sin(\gamma) + \cos(\gamma)\sin(\alpha)\sin(\beta)) \right. \\
& \left. + \cos(\beta)\sin(\alpha)\sin(\delta) \right)^2 \\
& + \cos(\varepsilon) \left(\sin(\beta)\sin(\delta) - \cos(\beta)\cos(\gamma)\cos(\delta) \right)^2 \\
& + \cos(\varepsilon) \left(\cos(\delta)(\sin(\alpha)\sin(\gamma) - \cos(\alpha)\cos(\gamma)\sin(\beta)) \right. \\
& \left. - \cos(\alpha)\cos(\beta)\sin(\delta) \right)^2
\end{aligned}$$

$$J_{\dot{E}q}(3,4) = 0$$

$$J_{\dot{E}q}(1,5) = 0$$

$$J_{\dot{E}q}(2,5) = 0$$

$$J_{\dot{E}q}(3,5) = 1$$

C ASC and WLN Tuning

C.1 ASC

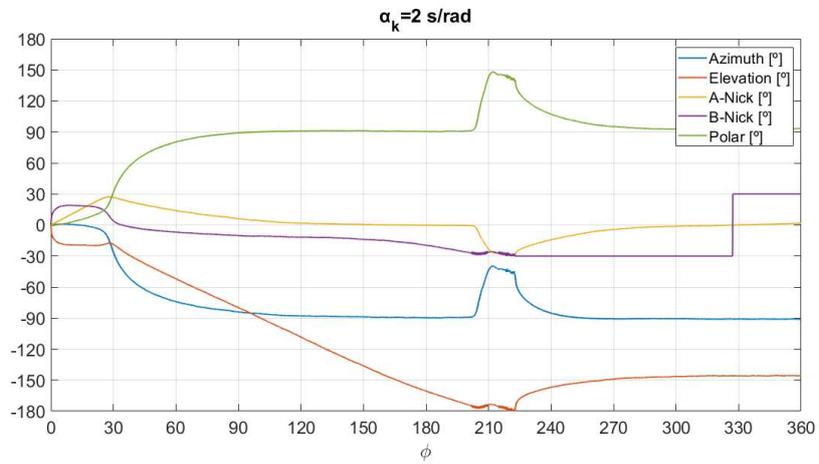


Fig-C 1: ASC for a Roll Rotation with $\alpha_k = 2 \text{ s/rad}$

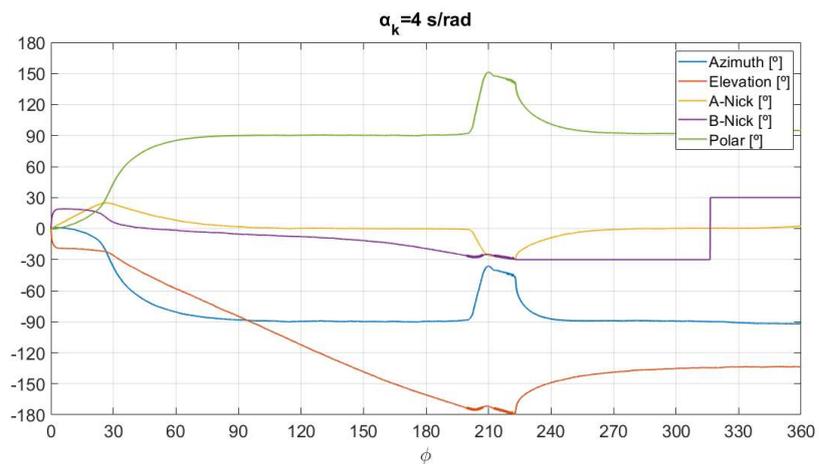


Fig-C 2: ASC for a Roll Rotation with $\alpha_k = 4 \text{ s/rad}$

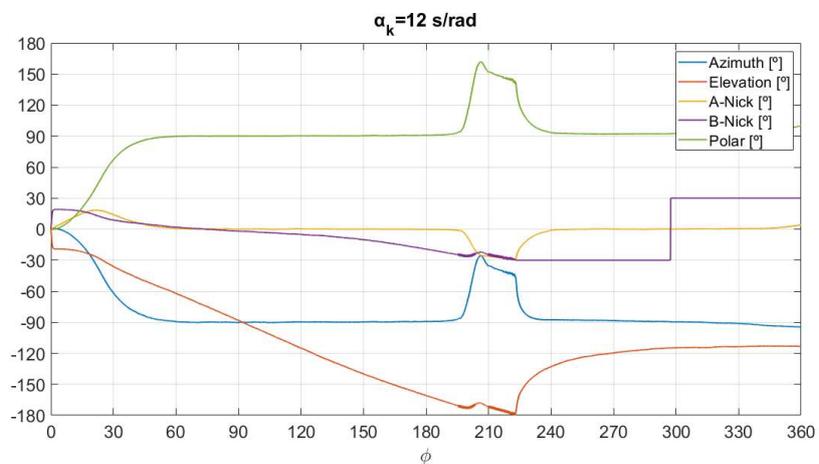


Fig-C 3: ASC for a Roll Rotation with $\alpha_k = 12 \text{ s/rad}$

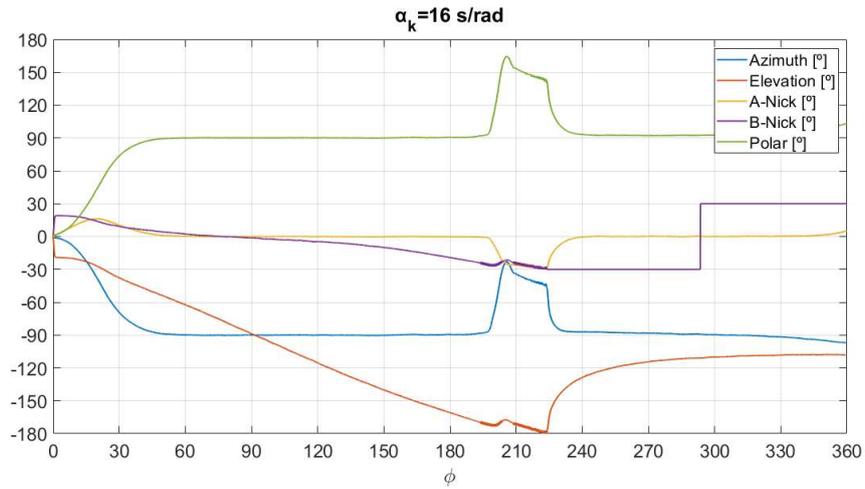


Fig-C 4: ASC for a Roll Rotation with $\alpha_k = 16 \text{ s/rad}$

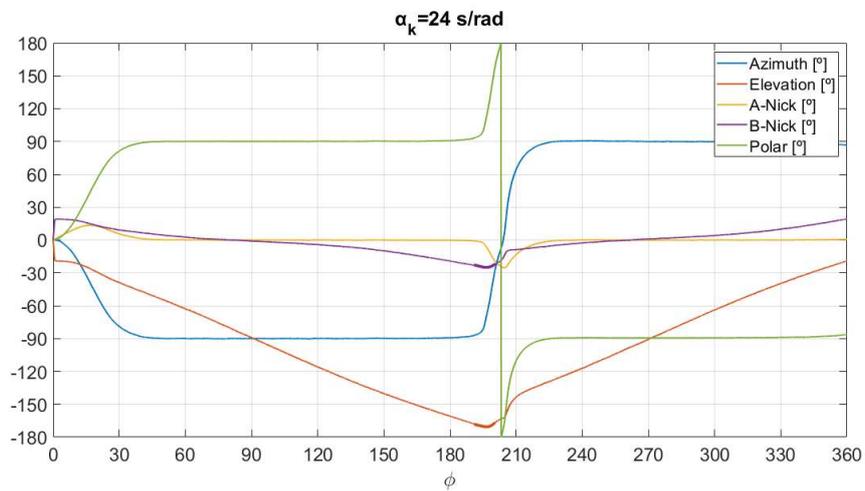


Fig-C 5: ASC for a Roll Rotation with $\alpha_k = 24 \text{ s/rad}$

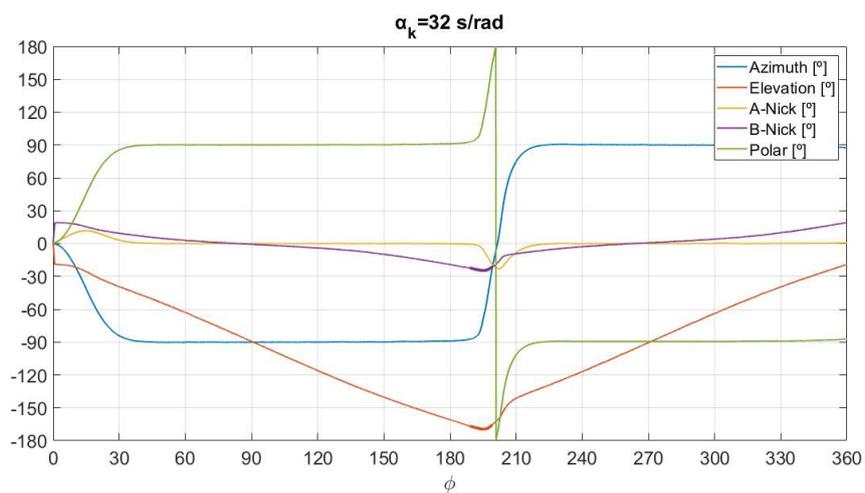


Fig-C 6: ASC for a Roll Rotation with $\alpha_k = 32 \text{ s/rad}$

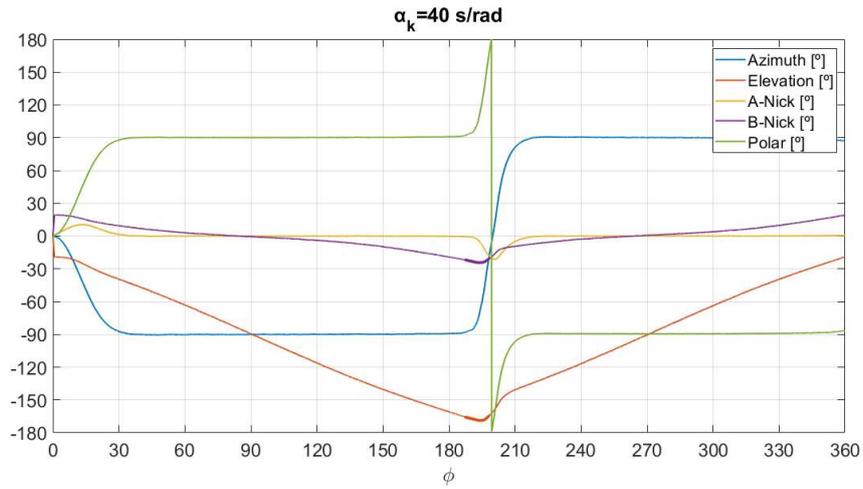


Fig-C 7: ASC for a Roll Rotation with $\alpha_k= 40$ s/rad

C.2 WLN

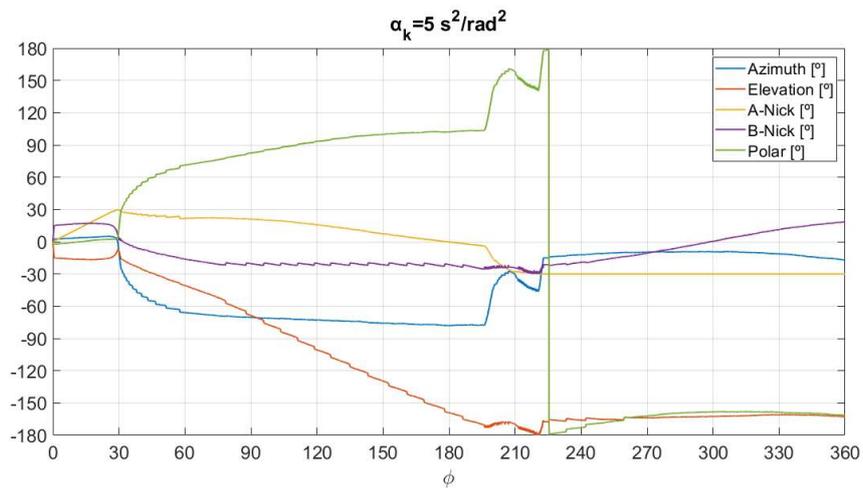


Fig-C 8: WLN for a Roll Rotation with $\alpha_k= 5$ s²/rad²

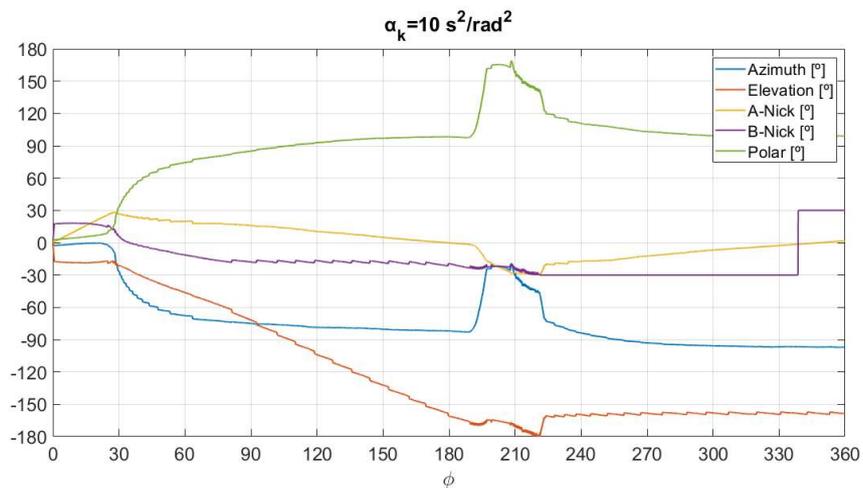


Fig-C 9: WLN for a Roll Rotation with $\alpha_k= 10$ s²/rad²

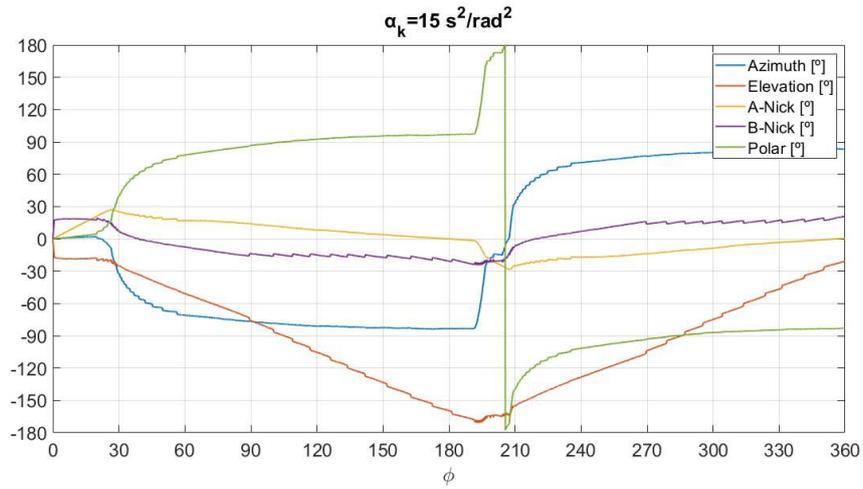


Fig-C 10: WLN for a Roll Rotation with $\alpha_k = 15 \text{ s}^2/\text{rad}^2$

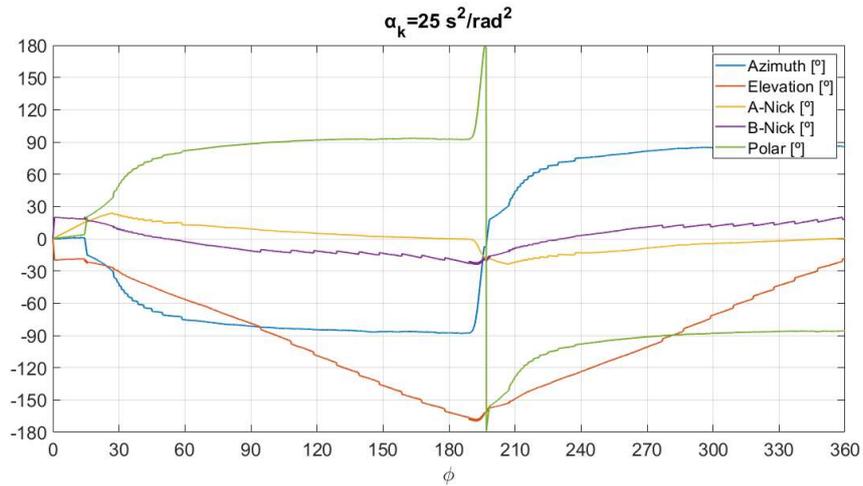


Fig-C 11: WLN for a Roll Rotation with $\alpha_k = 25 \text{ s}^2/\text{rad}^2$

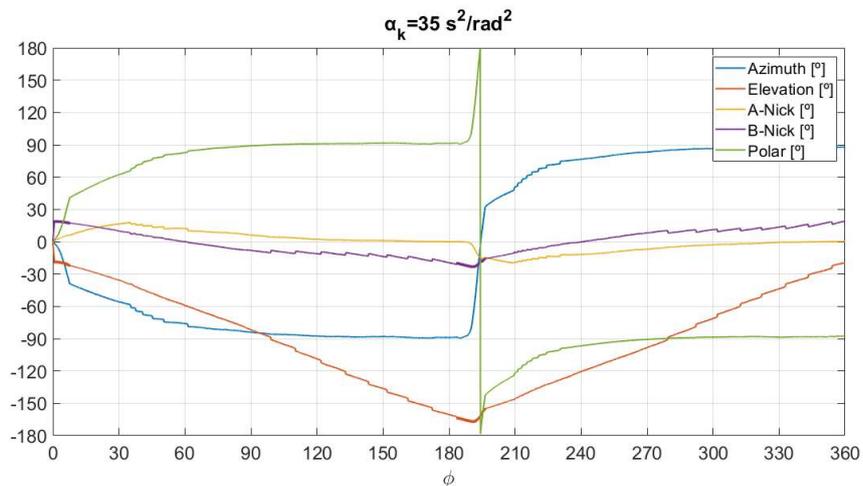


Fig-C 12: WLN for a Roll Rotation with $\alpha_k = 35 \text{ s}^2/\text{rad}^2$

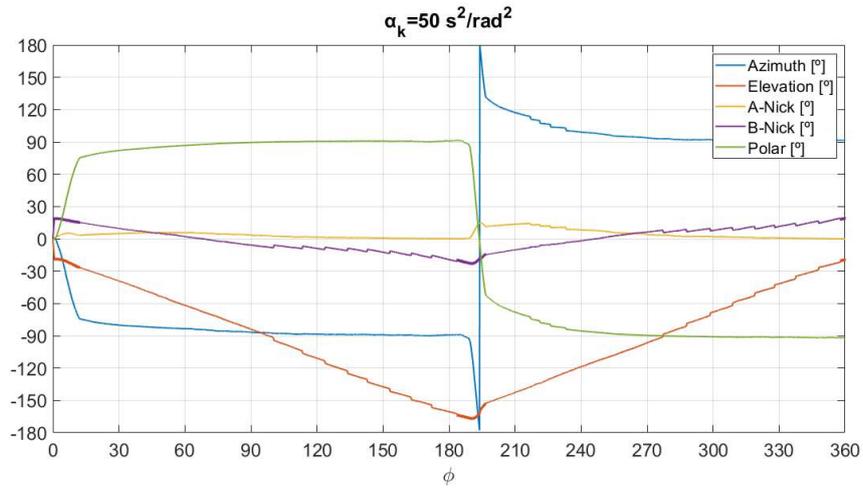


Fig-C 13: WLN for a Roll Rotation with $\alpha_k = 50 \text{ s}^2/\text{rad}^2$

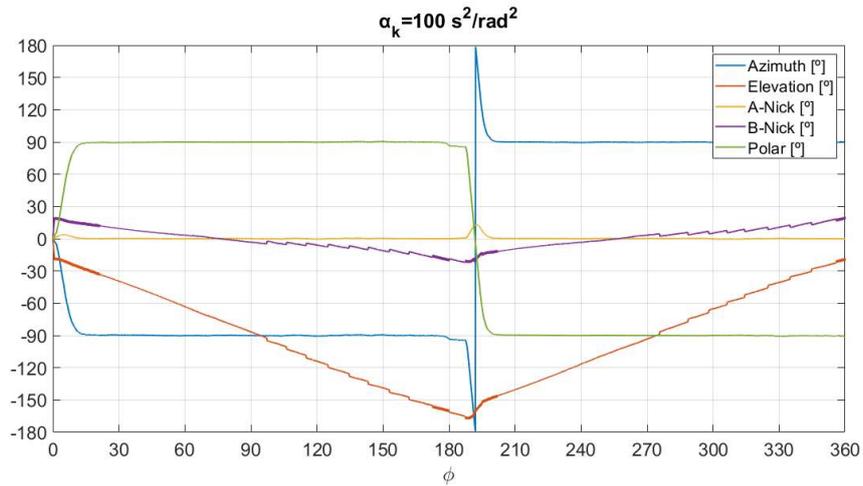


Fig-C 14: WLN for a Roll Rotation with $\alpha_k = 100 \text{ s}^2/\text{rad}^2$

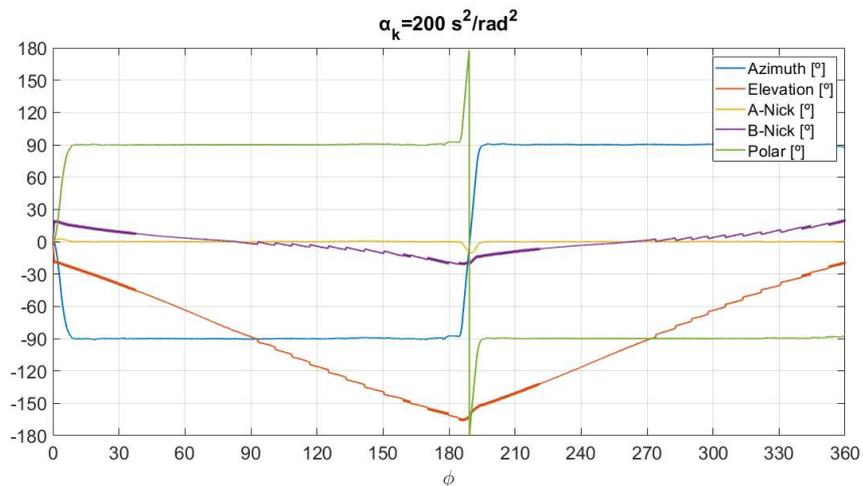


Fig-C 15: WLN for a Roll Rotation with $\alpha_k = 200 \text{ s}^2/\text{rad}^2$

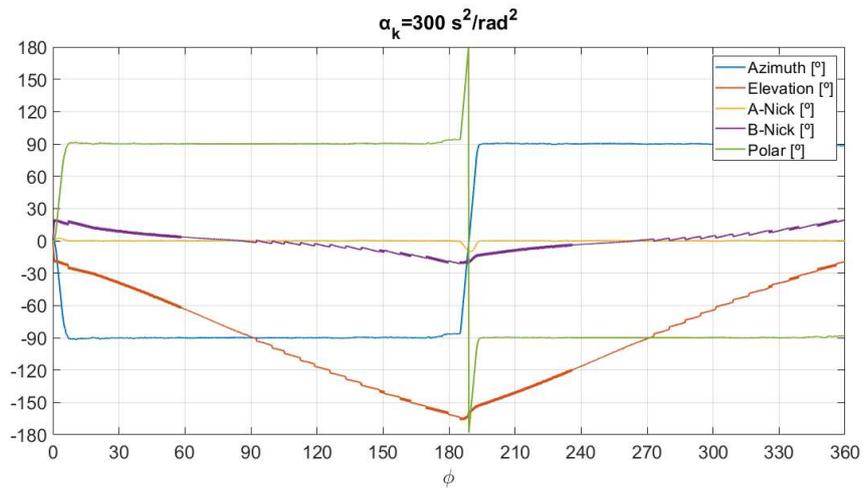


Fig-C 16: WLN for a Roll Rotation with $\alpha_k = 300 \text{ s}^2/\text{rad}^2$

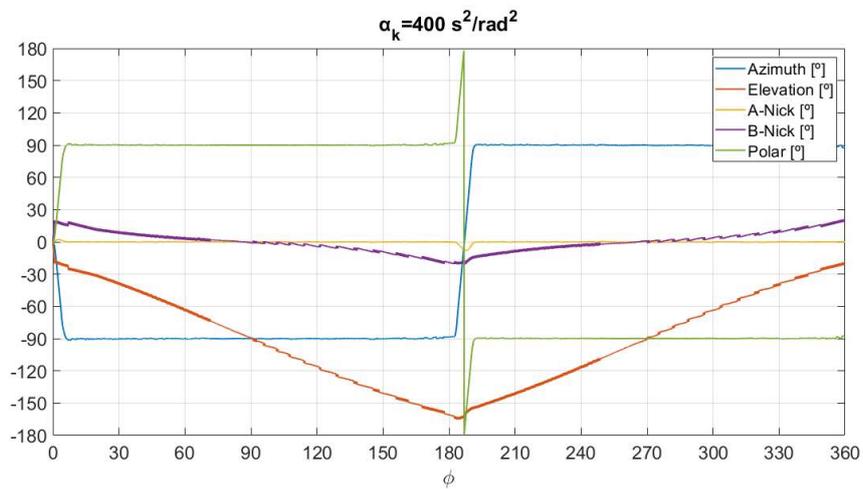


Fig-C 17: WLN for a Roll Rotation with $\alpha_k = 2 \text{ s/rad}$

D Inverse Kinematics Manifold for the Example Trajectory

In this appendice, the inverse kinematics solution for the example trajectory tested in Chapter 7 is presented. A discretization has been performed in the trajectory with a step of 15° . The following figures illustrate the associated 5-D Manifolds to each of these points.

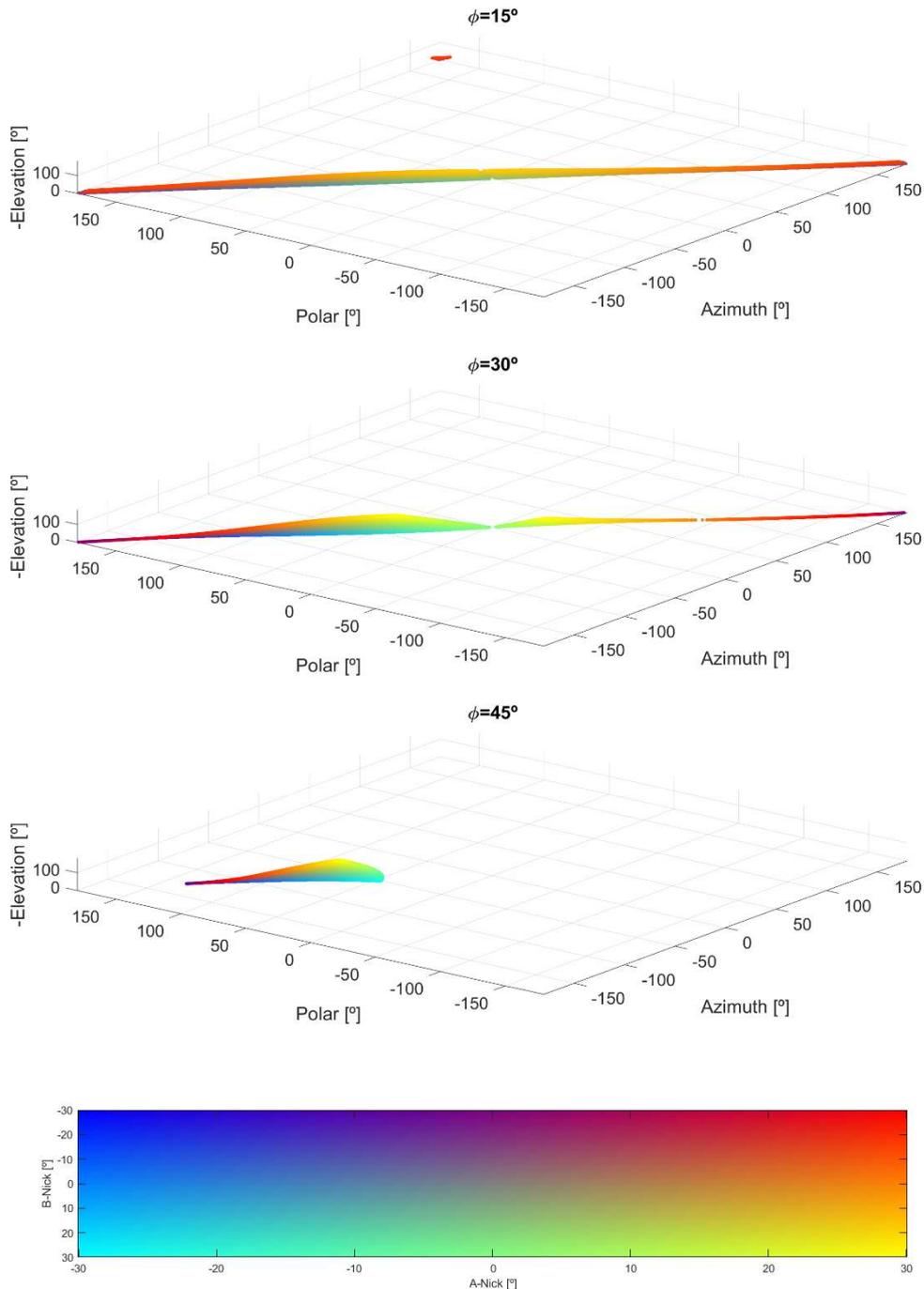


Fig-D 1: Solutions from $\phi = 15^\circ$ to $\phi = 45^\circ$

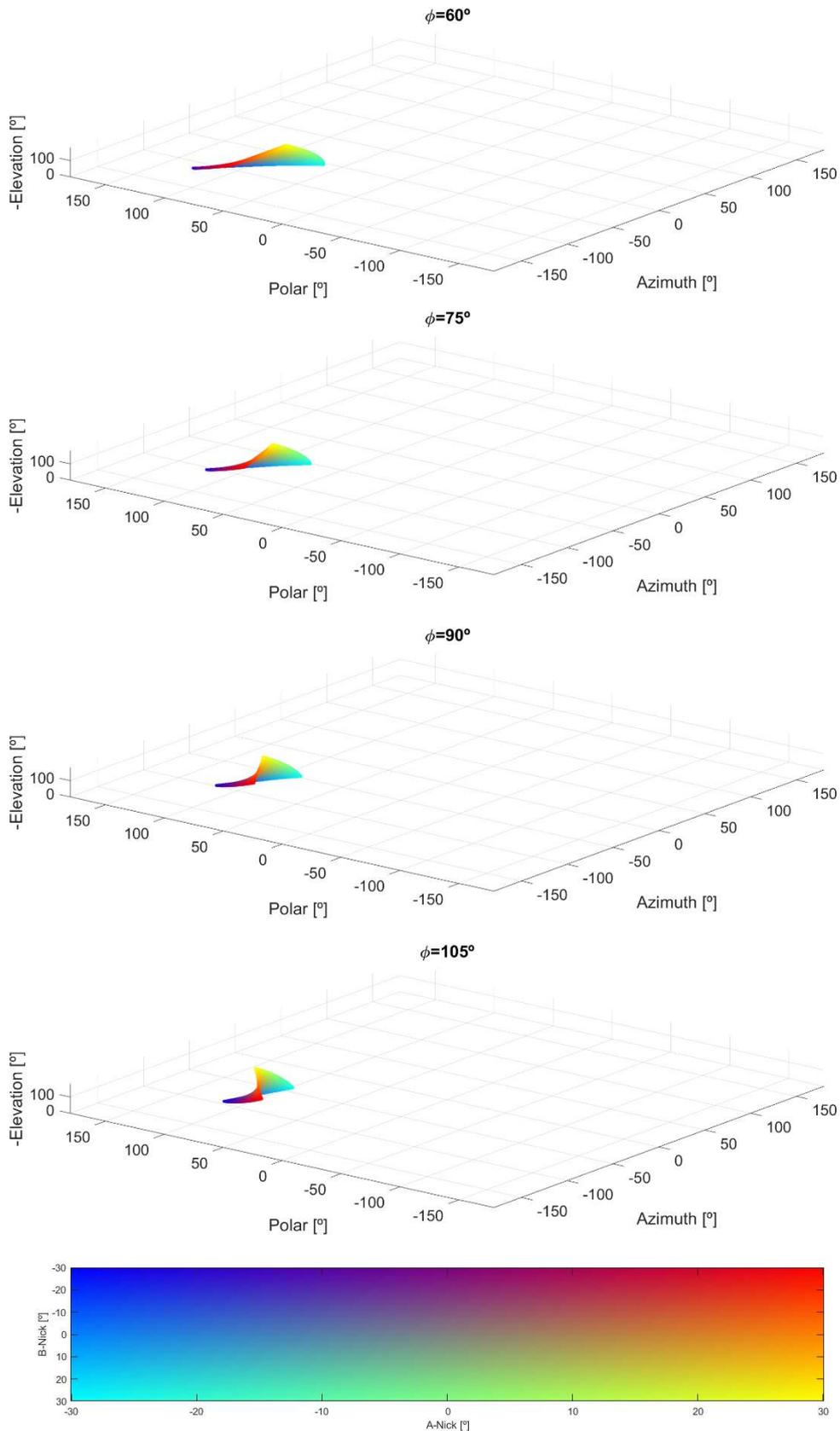


Fig-D 2: Solutions from $\phi=60^\circ$ to $\phi=105^\circ$

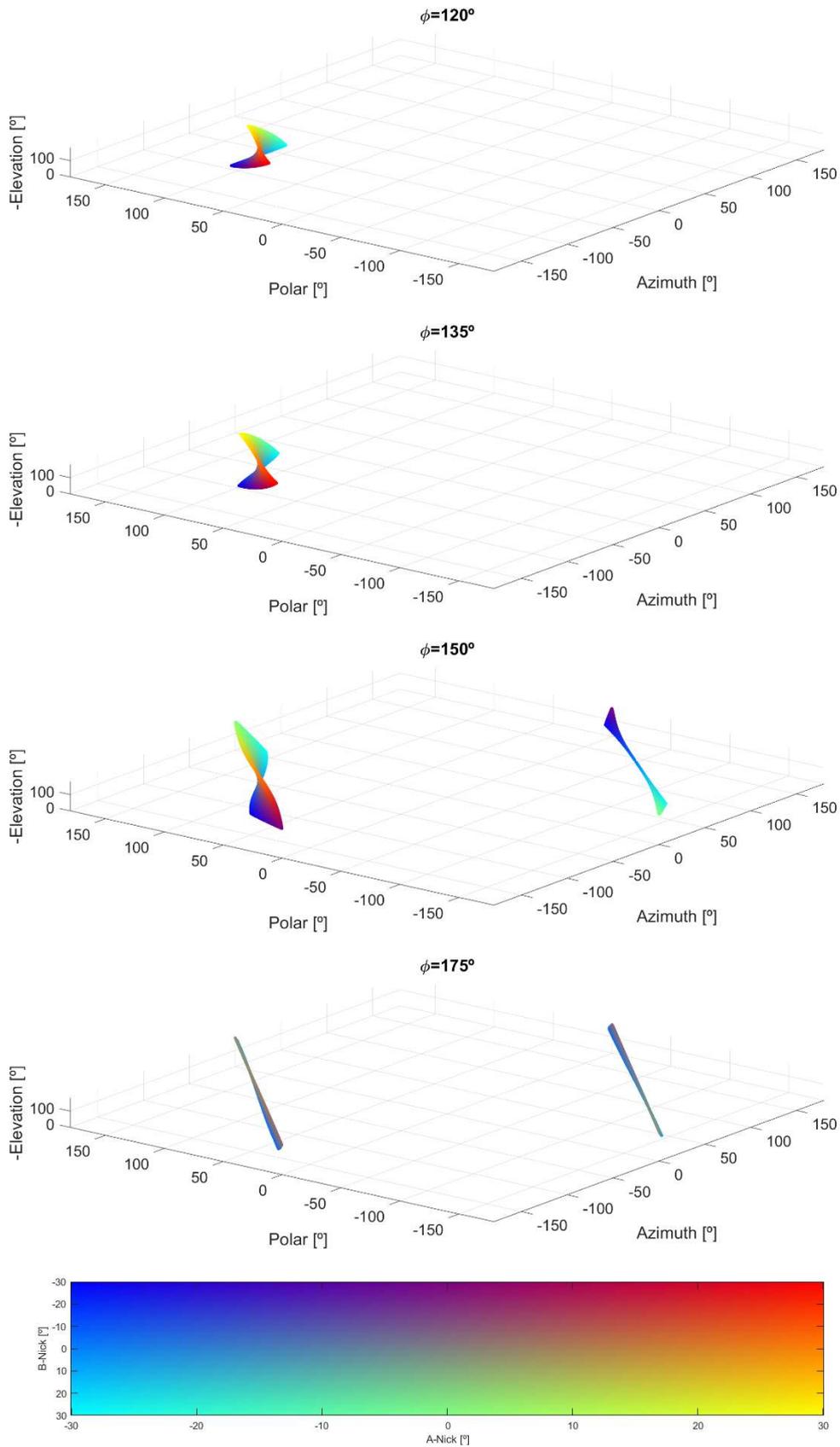


Fig-D 3: Solutions from $\phi=120^\circ$ to $\phi=175^\circ$

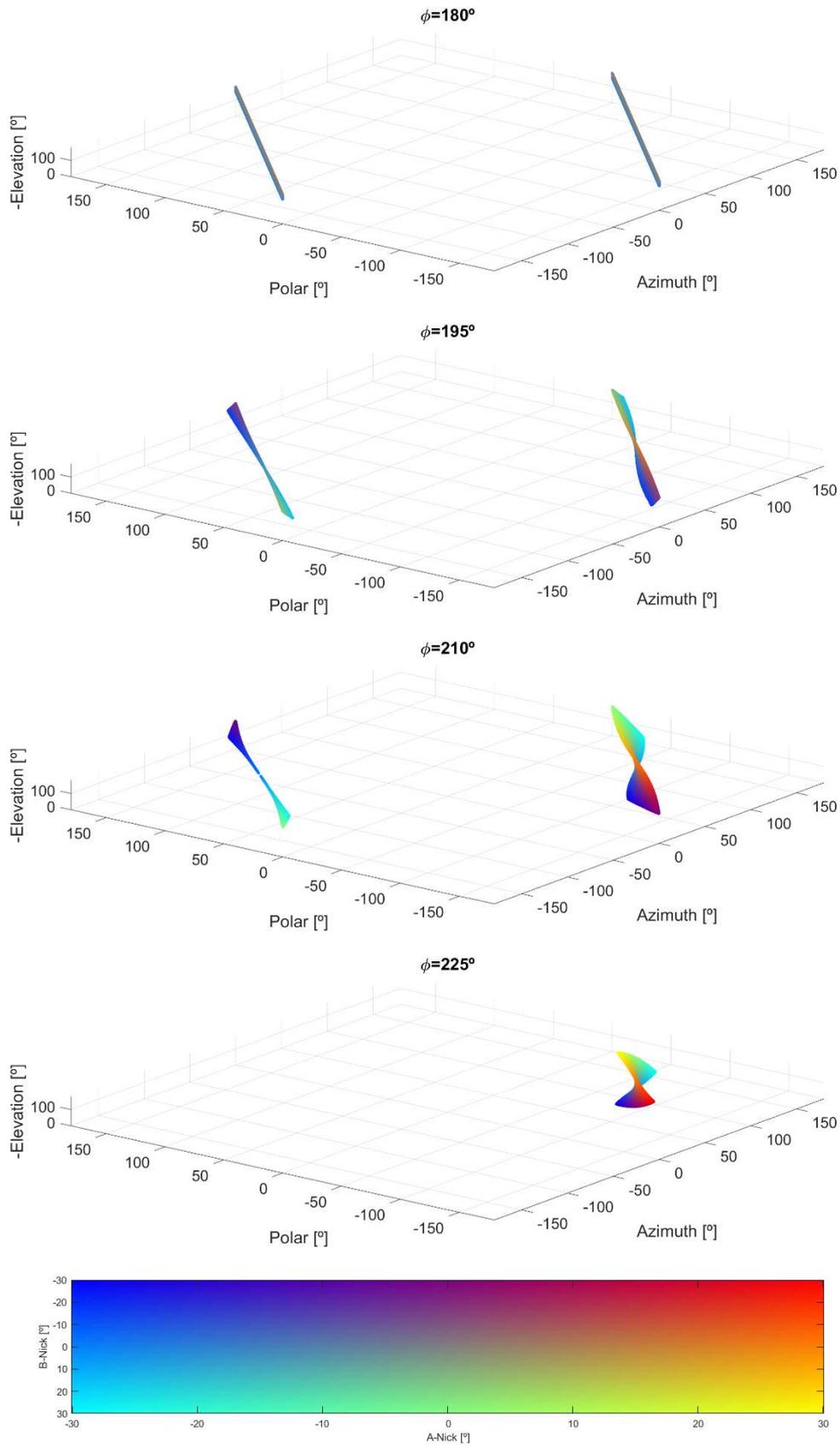


Fig-D 4: Solutions from $\phi=180^\circ$ to $\phi=225^\circ$

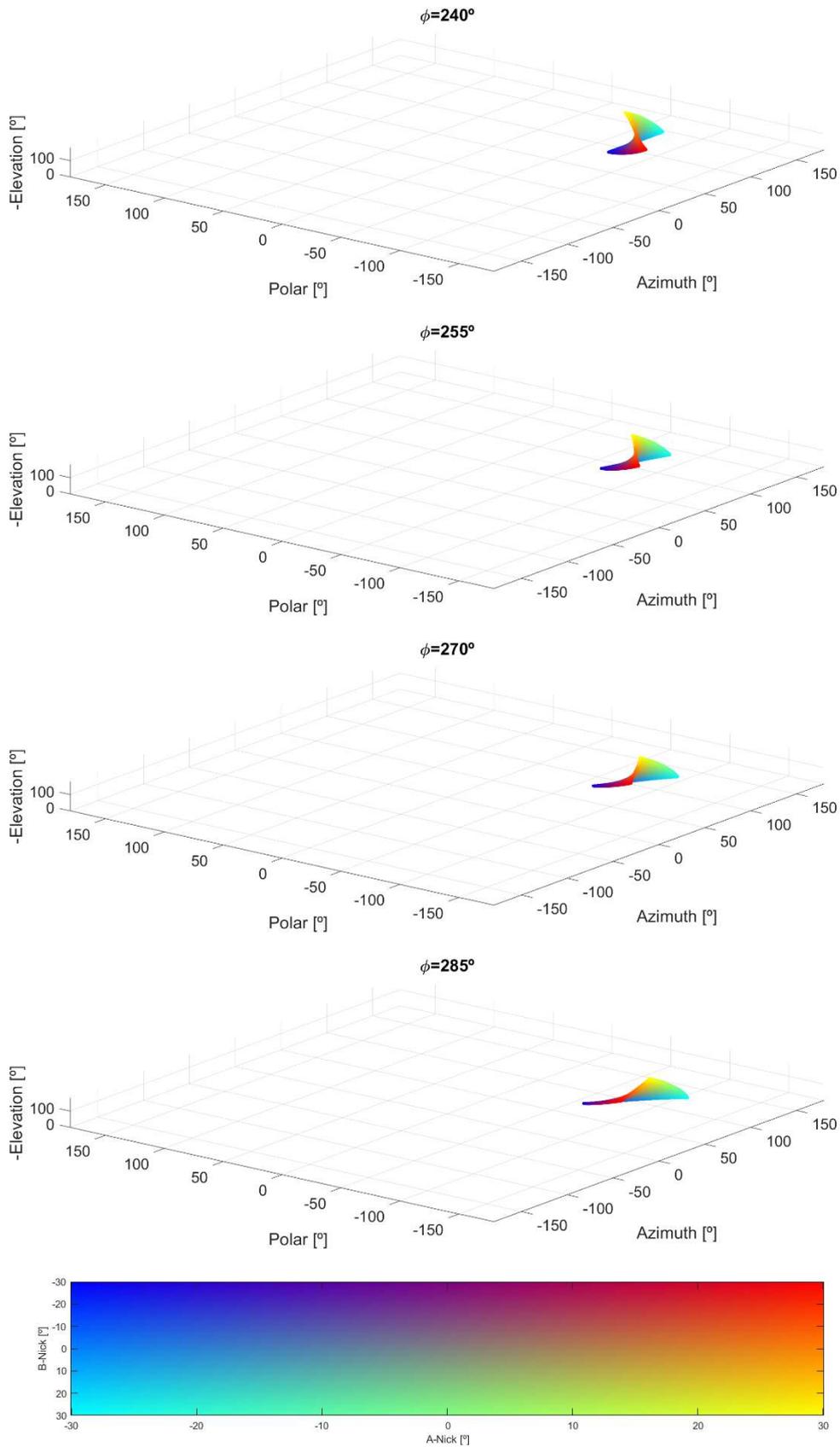


Fig-D 5: Solutions from $\phi=240^\circ$ to $\phi=285^\circ$

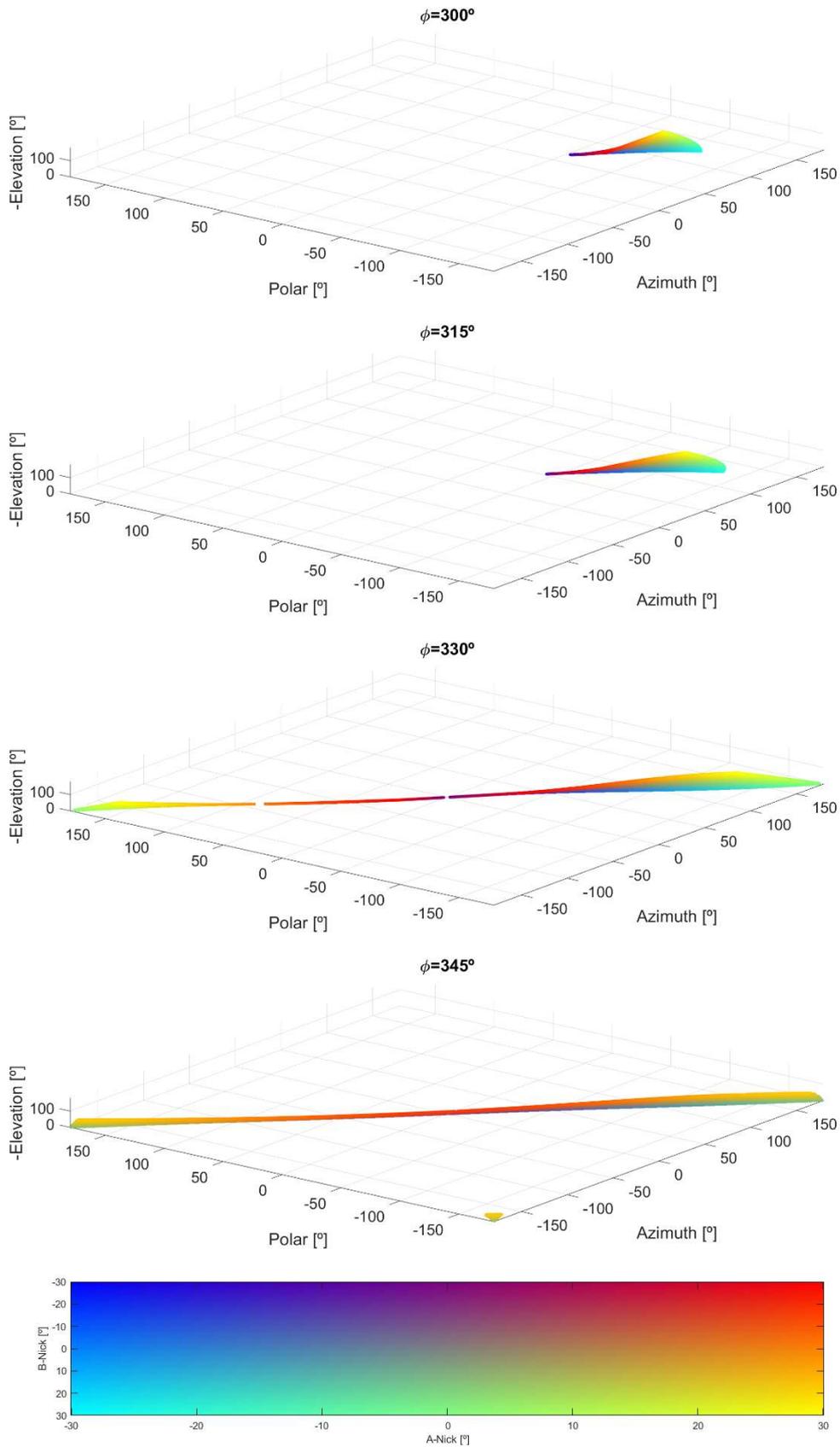


Fig-D 6: Solutions from $\phi=300^\circ$ to $\phi=345^\circ$

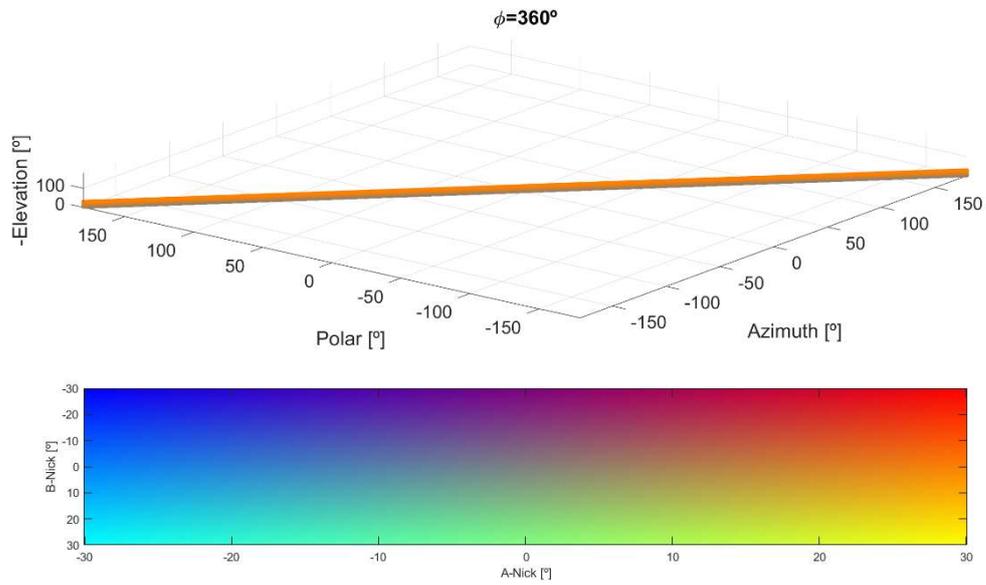


Fig-D 7: Solution for $\phi=360^\circ$

E Tuning of the Relaxed Isokinematic Centroid Tracking

In this Appendice the intermediate results of the tuning tuning process for the Relaxed Isokinematic Centroid Tracking are displayed. These results consist on both the output solution to the inverse kinematics for a roll rotation and the associated instantaneous efficiency for a given η_{min} . The effects of this parameter's variations are presented with a spacing of a 5%. The maximum value which proved to provide a valid trajectory was found to be a 50%.

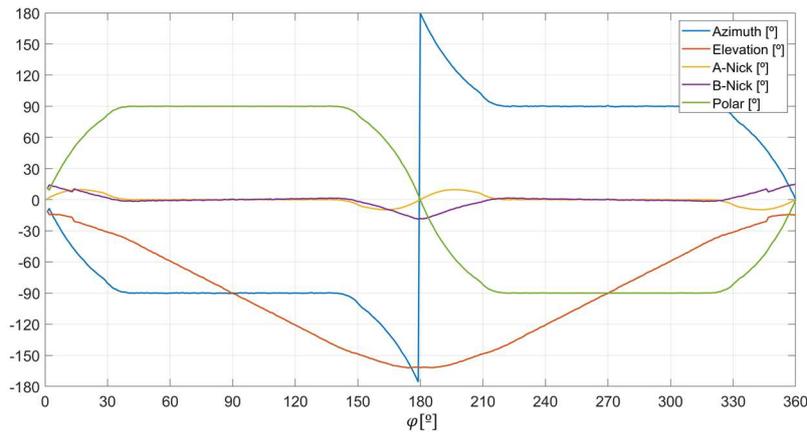


Fig-E 1: Relaxed Isokinematic Centroid Tracking Trajectory with $\eta_{min}=20\%$

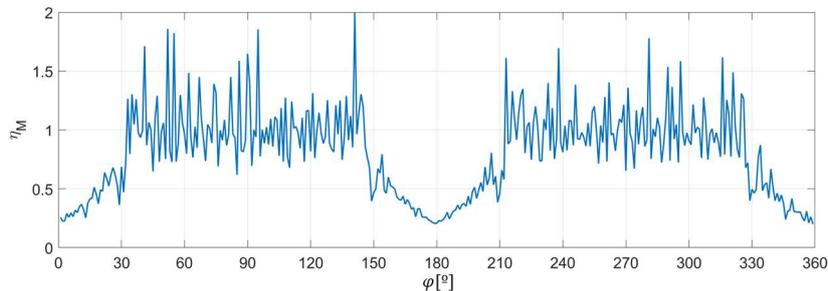


Fig-E 2: Efficiency of Relaxed Isokinematic Centroid Tracking Algorithm with $\eta_{min}=20\%$

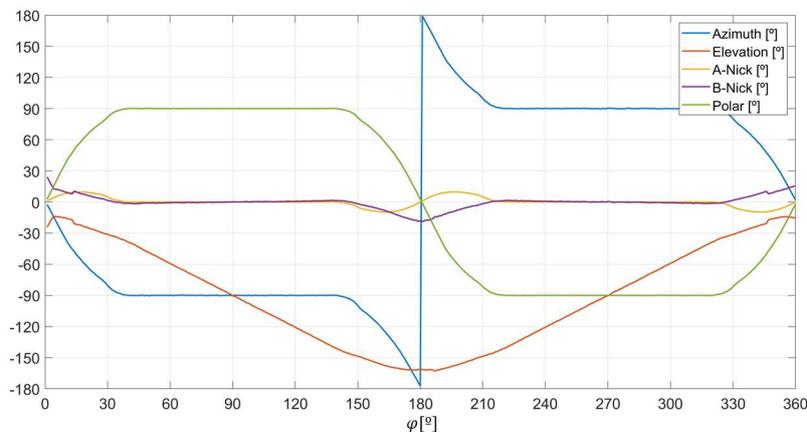


Fig-E 3: Relaxed Isokinematic Centroid Tracking Trajectory with $\eta_{min}=25\%$

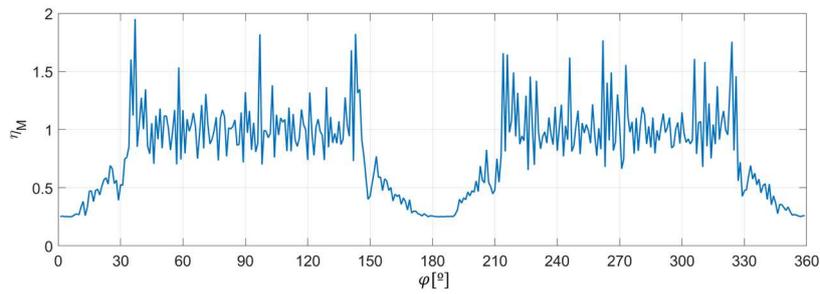


Fig-E 4: Efficiency of Relaxed Isokinematic Centroid Tracking Algorithm with $\eta_{\min}=25\%$

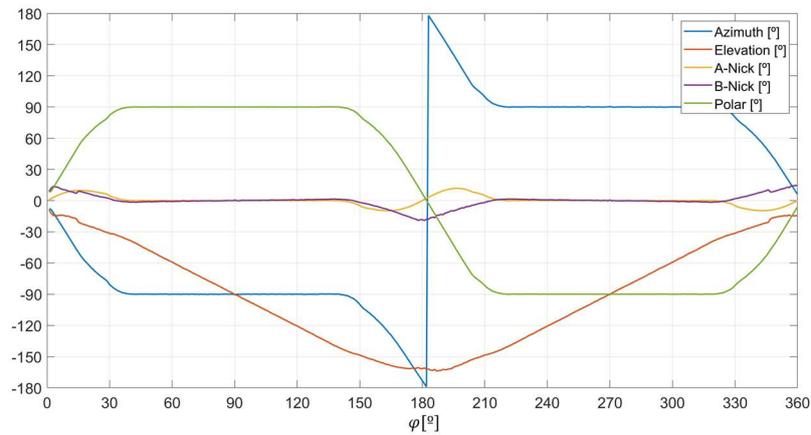


Fig-E 5: Relaxed Isokinematic Centroid Tracking Trajectory with $\eta_{\min}=30\%$

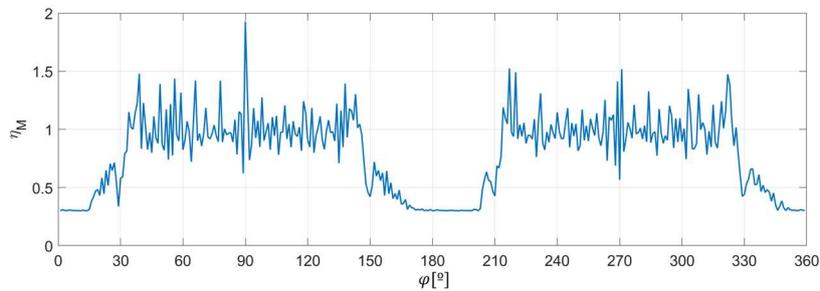


Fig-E 6: Efficiency of Isokinematic Centroid Tracking Tracking Algorithm with $\eta_{\min}=30\%$

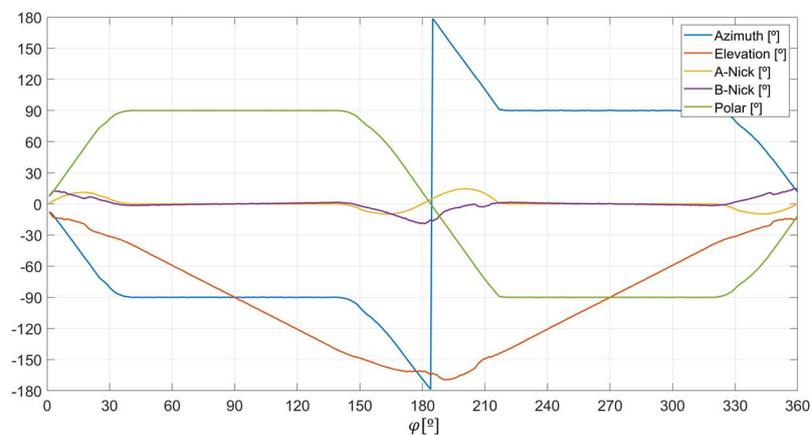


Fig-E 7: Relaxed Isokinematic Centroid Tracking Trajectory with $\eta_{\min}=35\%$

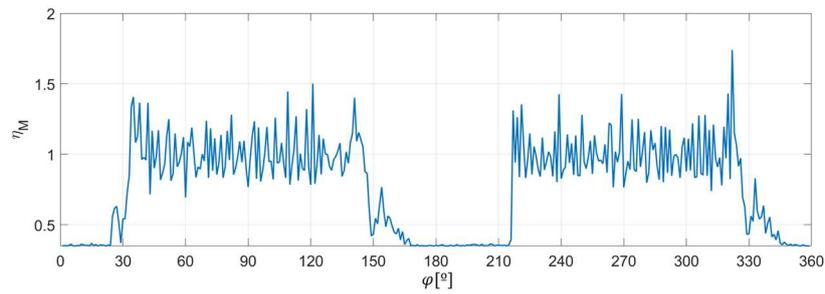


Fig-E 8: Efficiency of Relaxed Isokinematic Centroid Tracking Algorithm with $\eta_{\min}=35\%$

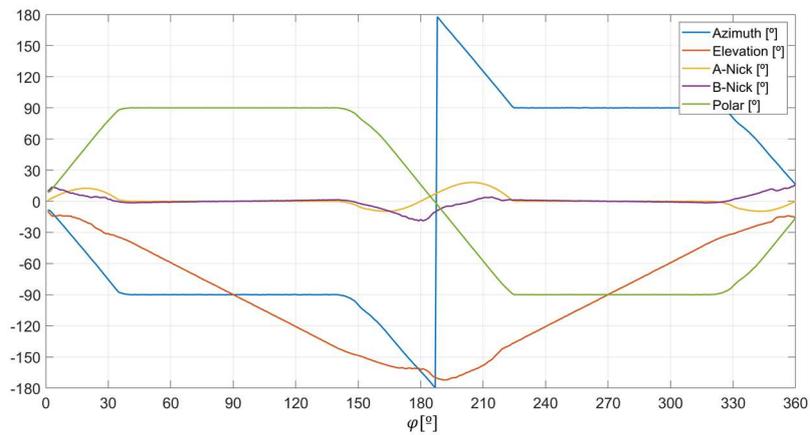


Fig-E 9: Relaxed Isokinematic Centroid Tracking Trajectory with $\eta_{\min}=40\%$

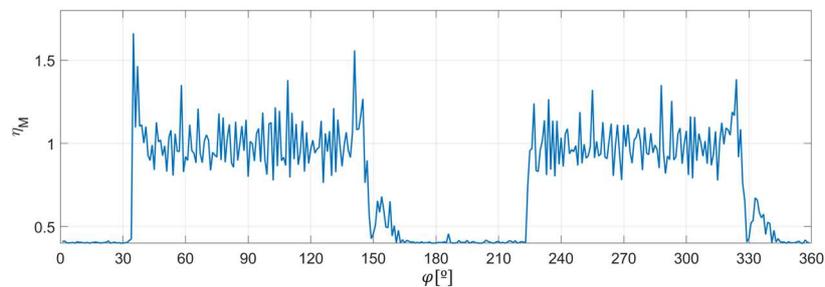


Fig-E 10: Efficiency of Relaxed Isokinematic Centroid Tracking Algorithm with $\eta_{\min}=40\%$

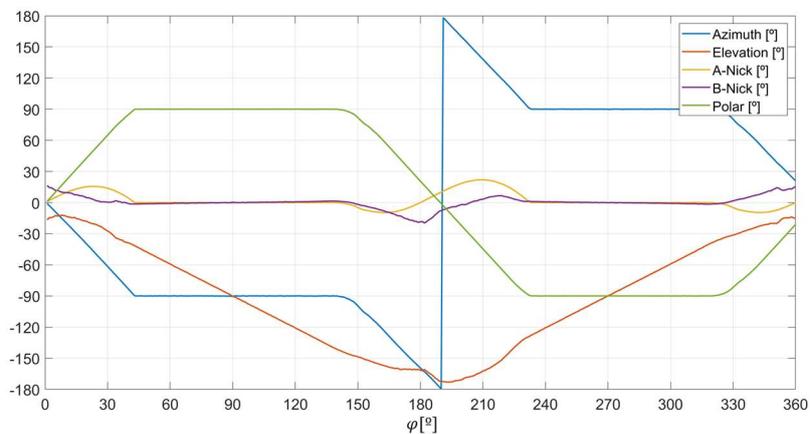


Fig-E 11: Relaxed Isokinematic Centroid Tracking Trajectory with $\eta_{\min}=45\%$

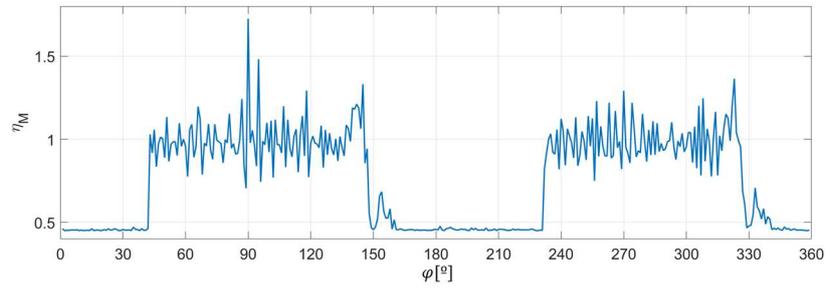


Fig-E 12: Efficiency of Relaxed Isokinematic Centroid Tracking Algorithm with $\eta_{\min}=45\%$

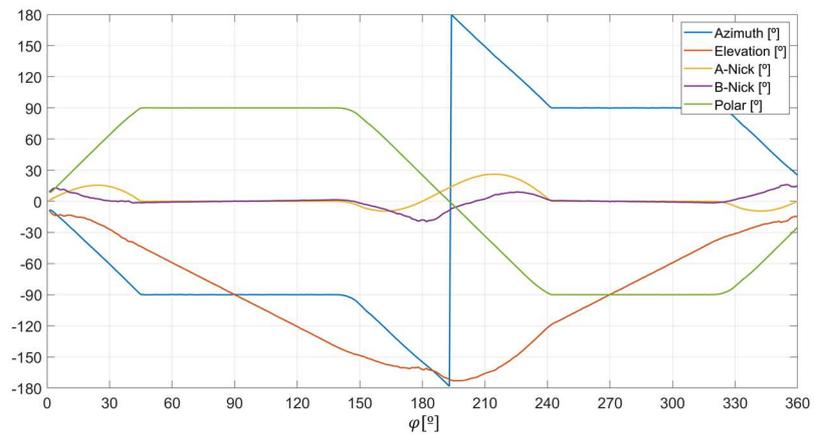


Fig-E 13: Relaxed Isokinematic Centroid Tracking Trajectory with $\eta_{\min}=50\%$

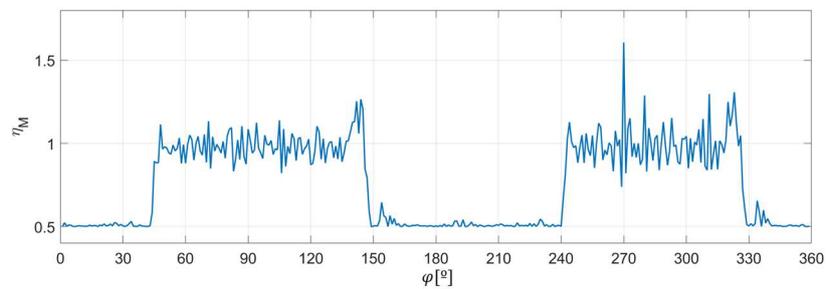


Fig-E 14: Efficiency of Relaxed Isokinematic Centroid Tracking Algorithm with $\eta_{\min}=50\%$

F MOVE-II Simulation

In this appendice the main results of the MOVE-II simulation are presented. The first section provides the computed Euler angles and Euler angles rates obtained from the gyroscopic data. The Second section illustrates the trajectories output by several inverse kinematic algorithms.

F.1 Attitude Data

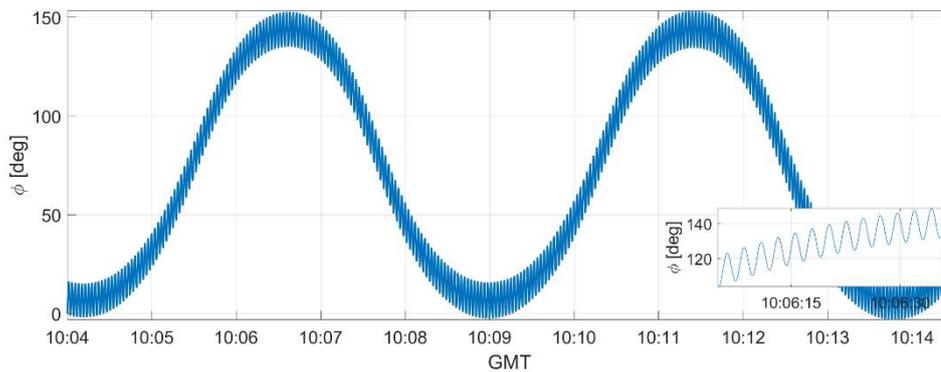


Fig-F 1: Roll Angle of MOVE-II over the Study Time

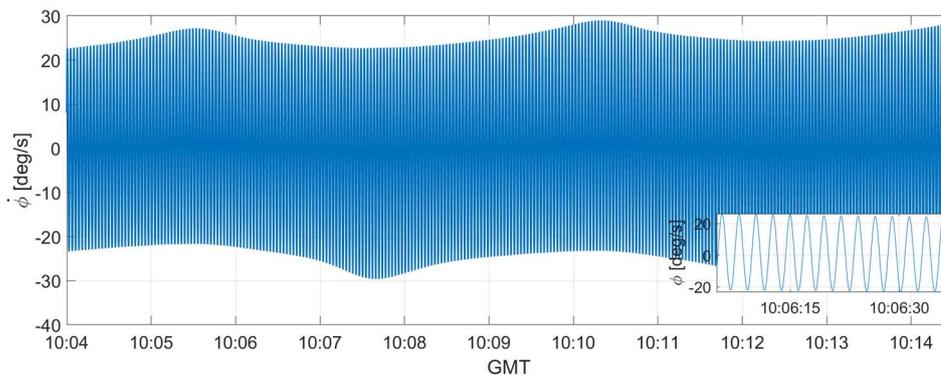


Fig-F 2: Roll Rate of MOVE-II over the Study Time

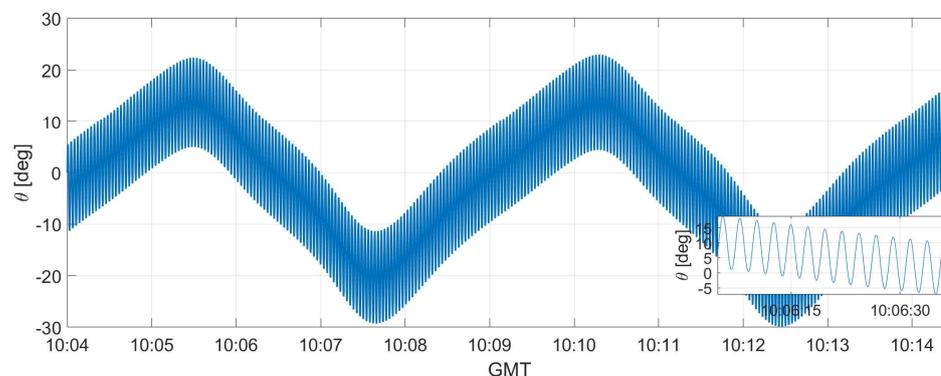


Fig-F 3: Pitch Angle of MOVE-II over the Study Time

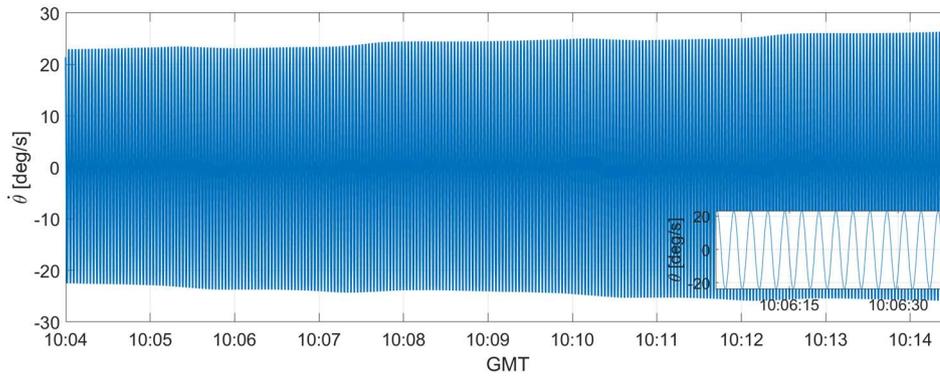


Fig-F 4: Pitch Rate of MOVE-II over the Study Time

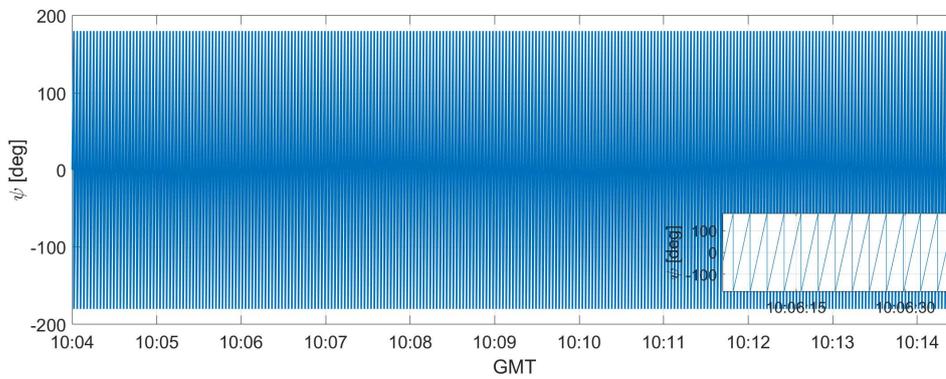


Fig-F 5: Yaw Angle of MOVE-II over the Study Time

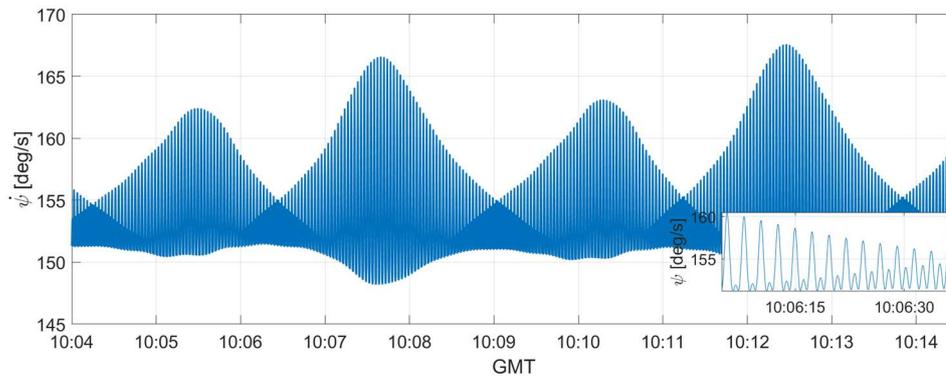


Fig-F 6: Yaw Rate of MOVE-II over the Study Time

F.2 Results of Inverse Kinematics Algorithms

Automatic Supervisory Control

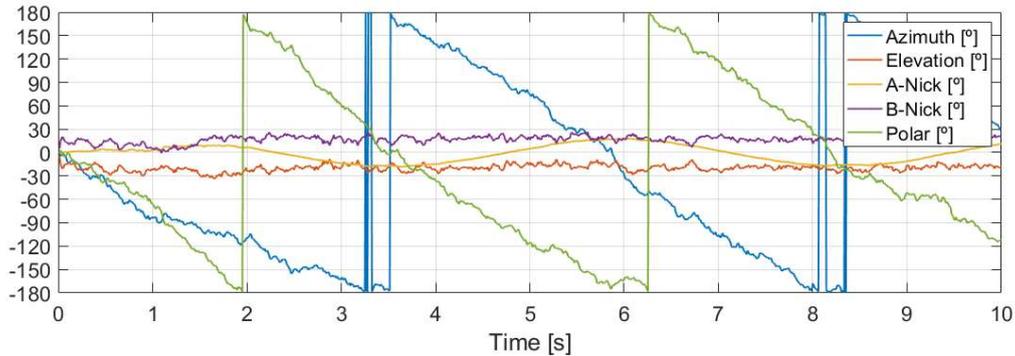


Fig-F 7: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with ASC and $\alpha_k=60$ s/rad (I)

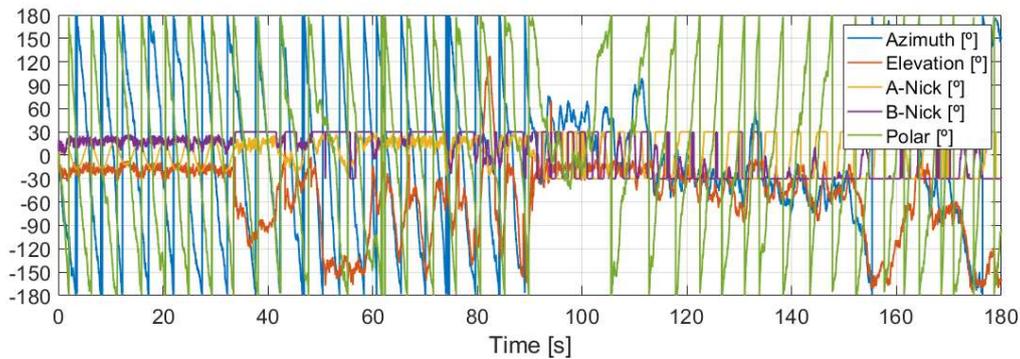


Fig-F 8: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with ASC and $\alpha_k=60$ s/rad (II)

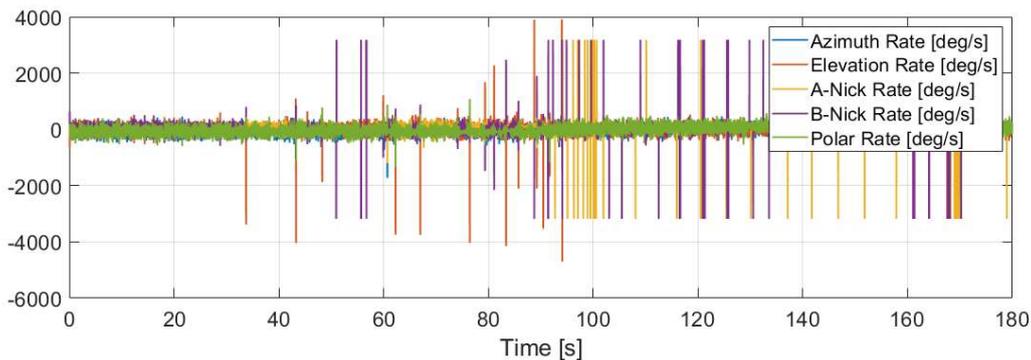


Fig-F 9: Angular Velocities of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with ASC and $\alpha_k=60$ s/rad

Relaxed Isokinematic Centroid Tracking

Values of η_{\min} that Led to Locking

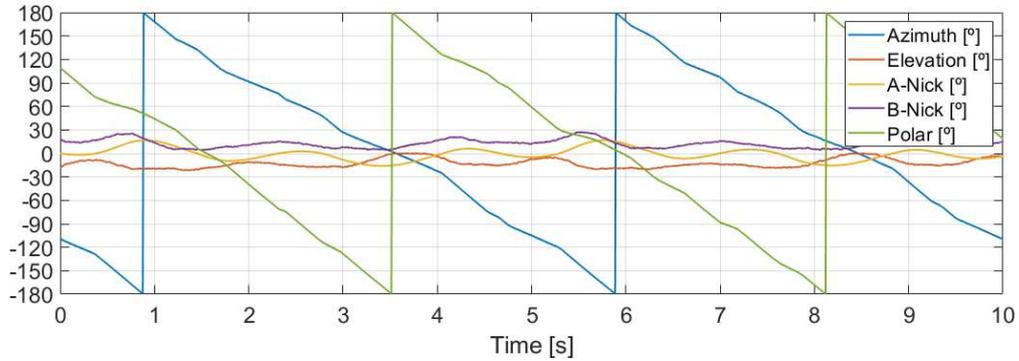


Fig-F 10: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=180\%$

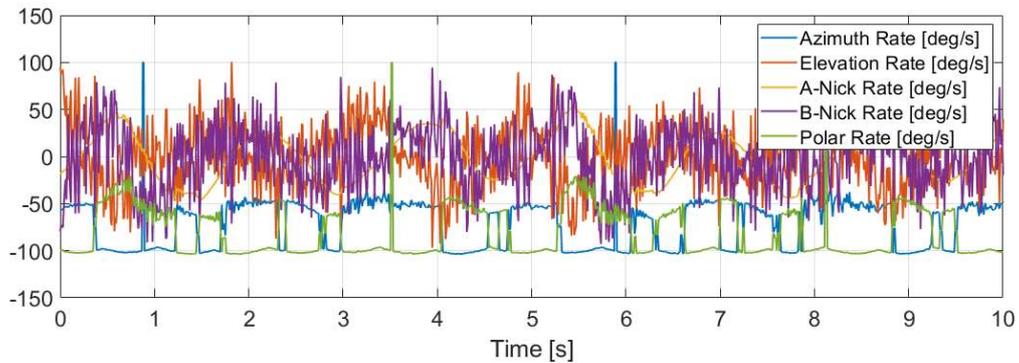


Fig-F 11: Angular Velocities of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=180\%$

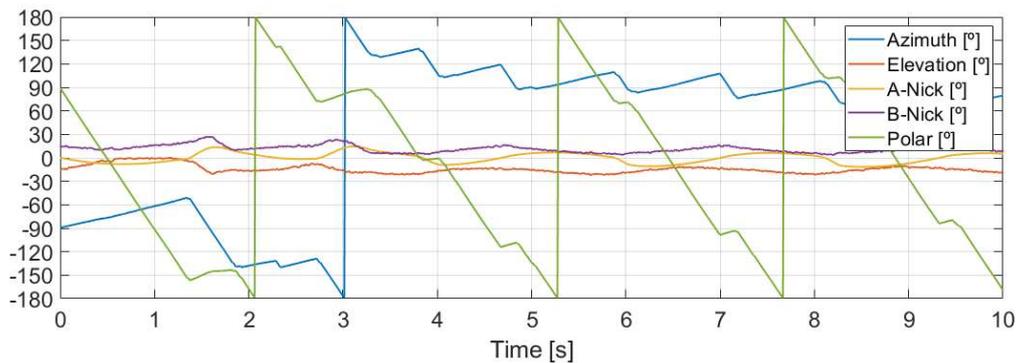


Fig-F 12: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=150\%$

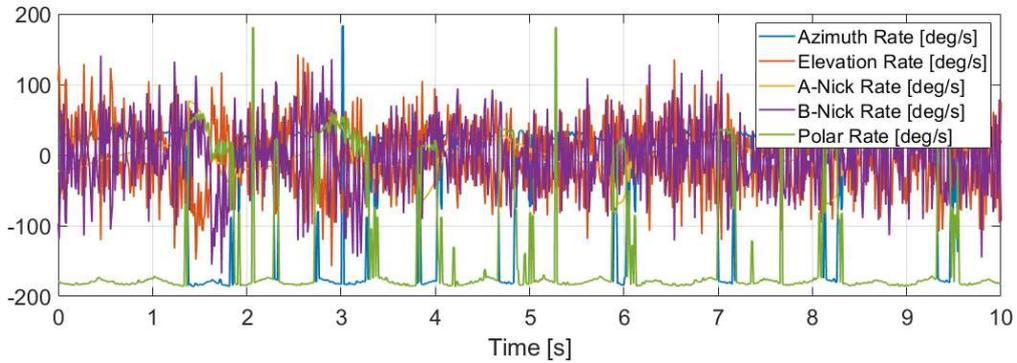


Fig-F 13: Angular Velocities of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=150\%$

Values of η_{\min} that provided Convergence

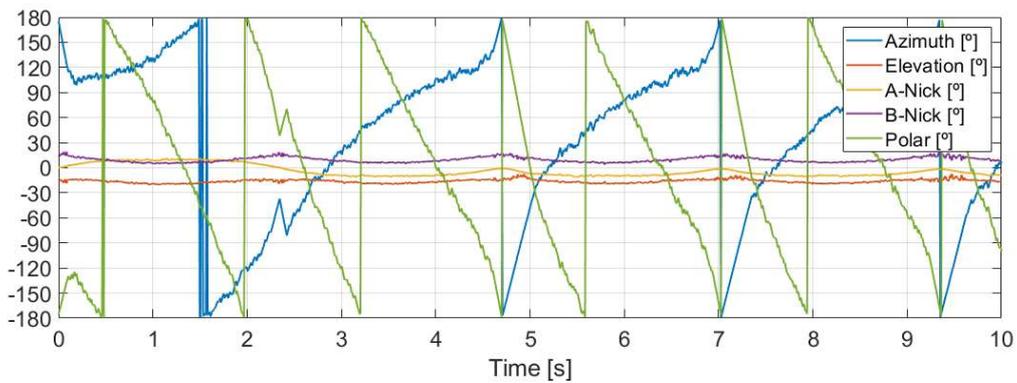


Fig-F 14: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=27\%$ (I)

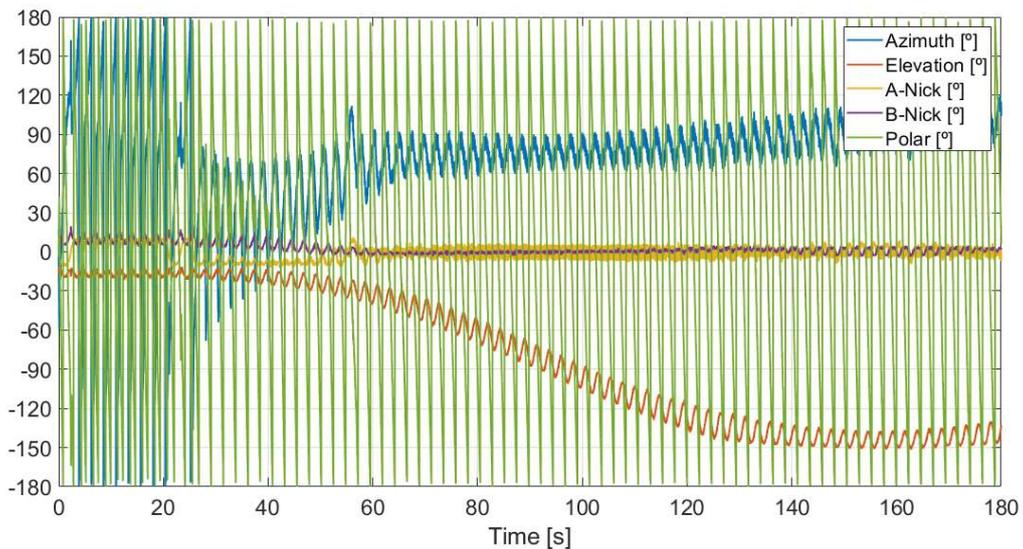


Fig-F 15: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=27\%$ (II)

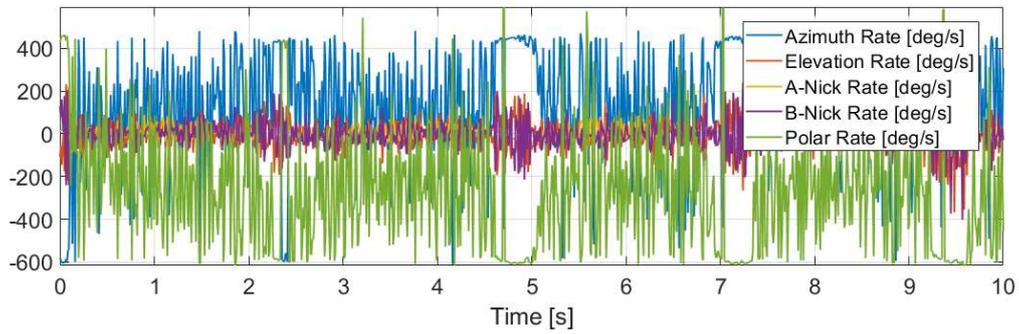


Fig-F 16: Angular Velocities of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=27\%$ (I)

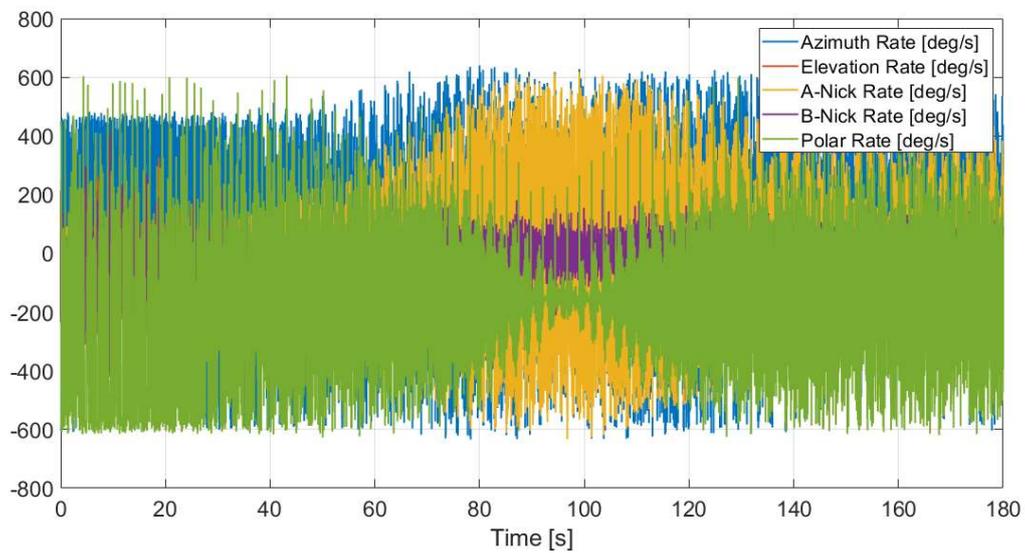


Fig-F 17: Angular Velocities of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=27\%$ (II)

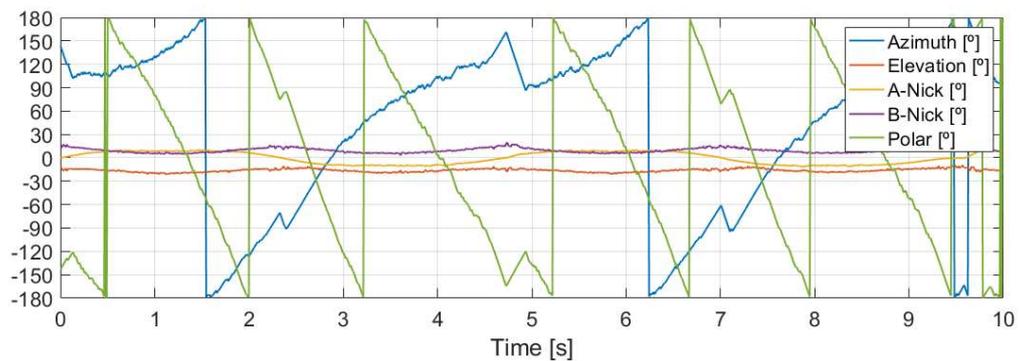


Fig-F 18: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=50\%$ (I)

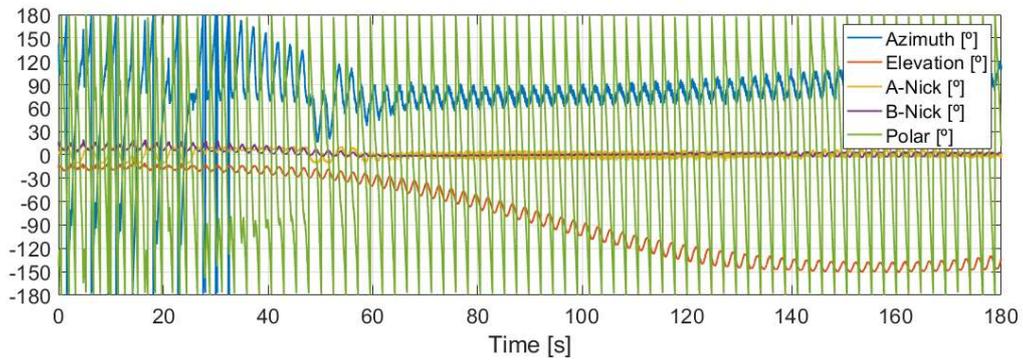


Fig-F 19: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=50\%$ (II)

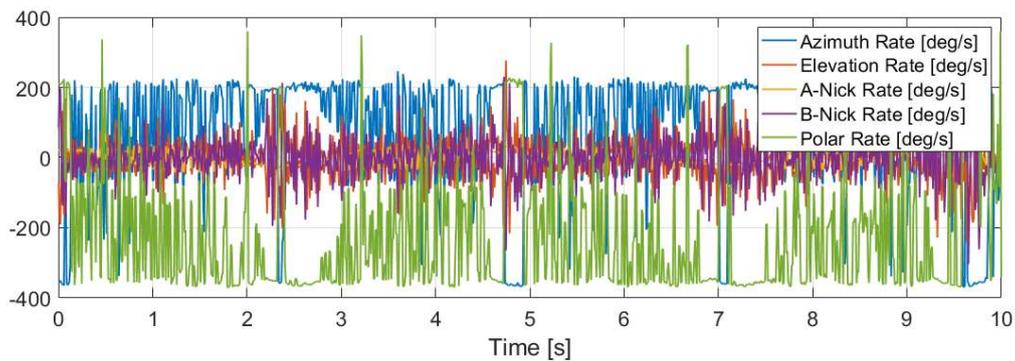


Fig-F 20: Angular Velocities of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=50\%$ (I)

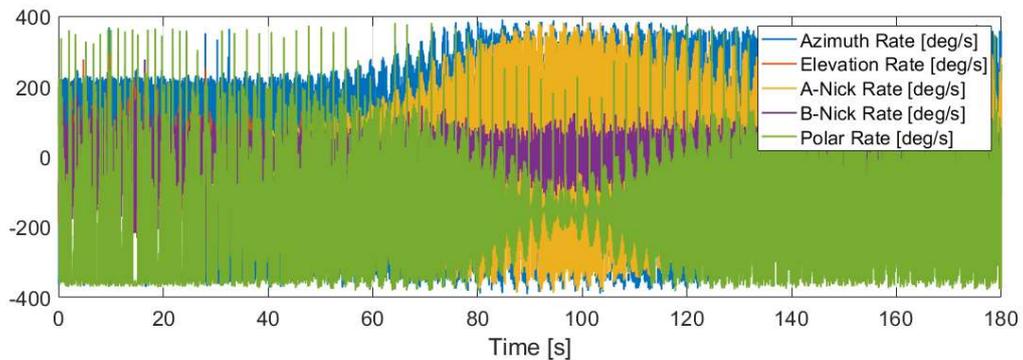


Fig-F 21: Angular Velocities of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=50\%$ (II)

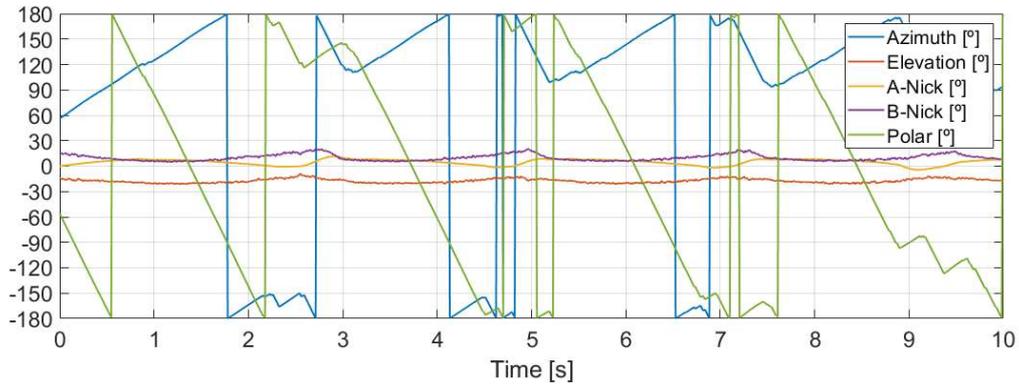


Fig-F 22: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=70\%$ (I)

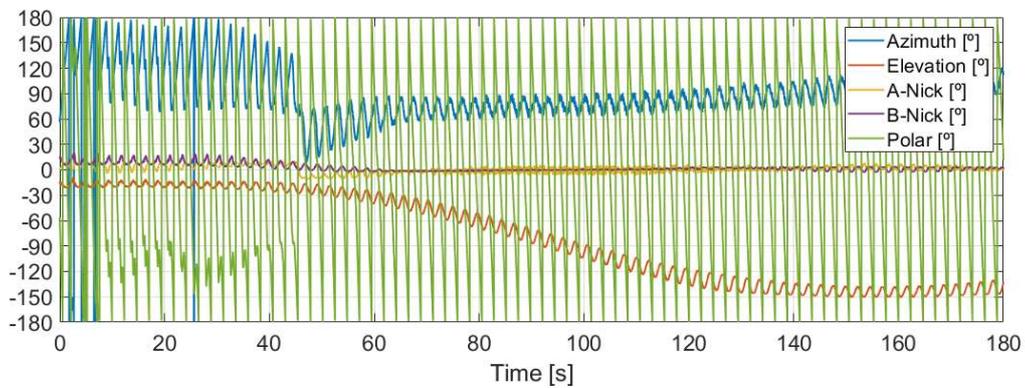


Fig-F 23: Configuration of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=70\%$ (I)

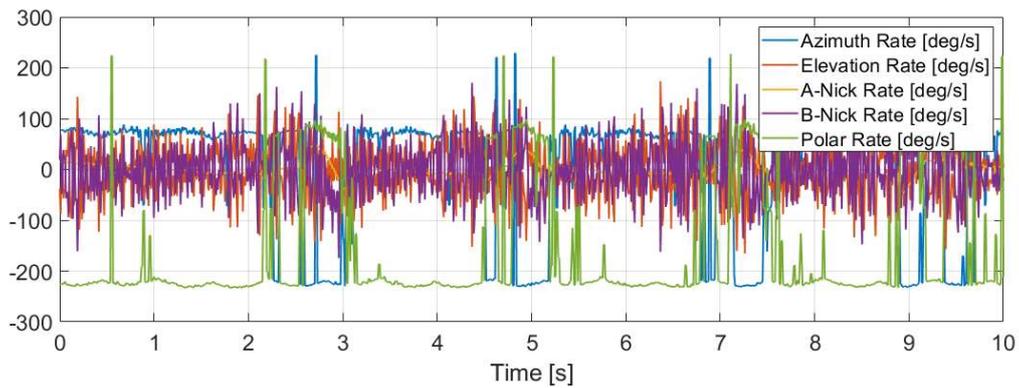


Fig-F 24: Angular Velocities of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=70\%$ (I)

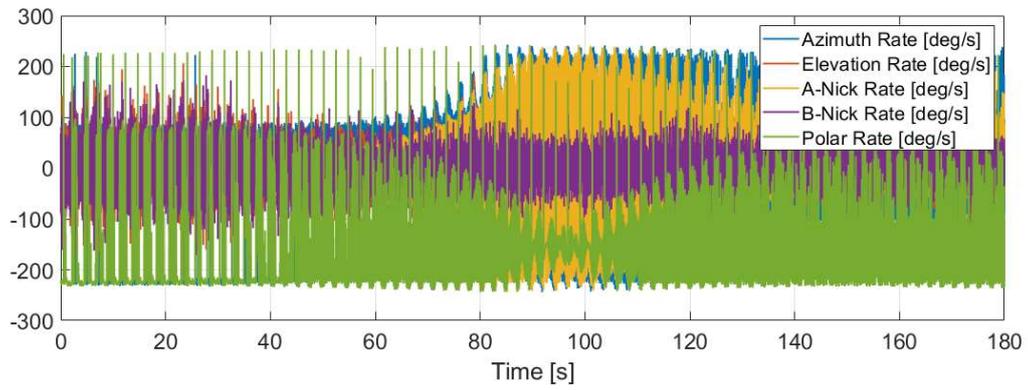


Fig-F 25: : Angular Velocities of the RACOON-Lab's Joints for Simulating MOVE-II Attitude History with RIKCT and $\eta_{\min}=70\%$ (II)

