



TRABALHO DE GRADUAÇÃO

**Trajectory Control of Anthropomorphic
Compliant Manipulator with Dual Quaternion Based
Kinematic Controllers**

Marcos da Silva Pereira

Brasília, novembro de 2016



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

Trajectory Control of Anthropomorphic Compliant Manipulator with Dual Quaternion Based Kinematic Controllers

Marcos da Silva Pereira

*Relatório submetido como requisito parcial de obtenção
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Geovany Araújo Borges, ENE/UnB
Orientador

Profa. Mariana Costa Bernardes Matias,
FGA/UnB
Co-Orientadora

Prof. João Yoshiyuki Ishihara, ENE/UnB
Examinador Interno

Dr. Luis Felipe da Cruz Figueiredo, ENE/UnB
Examinador Interno

Brasília, novembro de 2016

FICHA CATALOGRÁFICA

PEREIRA, MARCOS DA SILVA

Trajectory control of anthropomorphic compliant manipulator with dual quaternion based kinematic controllers,

[Distrito Federal] 2016.

xiv, 140p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2016). Trabalho de Graduação – Universidade de Brasília.Faculdade de Tecnologia.

1. Controle cinemático

2.Robótica

3. Quaternios duais

I. Mecatrônica/FT/UnB

II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

PEREIRA, M. S., (2016). Trajectory control of anthropomorphic compliant manipulator with dual quaternion based kinematic controllers. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº019, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 140p.

CESSÃO DE DIREITOS

AUTOR: Marcos da Silva Pereira

TÍTULO DO TRABALHO DE GRADUAÇÃO: Trajectory control of anthropomorphic compliant manipulator with dual quaternion based kinematic controllers.

GRAU: Engenheiro

ANO: 2016

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Marcos da Silva Pereira

SQN 314, Bloco J, Apartamento 508.

70767-100 Brasília – DF – Brasil.

Dedicatória

Aos meus pais, José Poli e Dácia Ibiapina.

Marcos da Silva Pereira

Agradecimentos

Ao longo dos meus anos de graduação muitas pessoas estiveram junto comigo.

Primeiro, agradeço aos meus orientadores que me guiaram durante esse trabalho. Agradeço à Profª. Mariana Bernades pela companhia, por ter estado sempre presente compartilhando seus conhecimentos e ideias, tanto nos caminhos em que tudo deu certo quanto pelo apoio e compreensão nos obstáculos que surgiram. Também sou eternamente grato pela companhia e amizade do Luis Figueiredo, que com seu doutorado tornou possível esse trabalho de graduação, que pacientemente me ensinou conceitos do mais real ao mais imaginário e dual e que me fez entender como um pesquisador enfrenta desafios. Agradeço ao Prof. Geovany que contribuiu com conhecimentos fundamentais para a minha formação durante a graduação e compartilha sua sabedoria apoiando seus alunos tanto em pesquisa quanto em competições de robótica e sempre prezou pela qualidade do ambiente de trabalho no laboratório LARA.

Agradeço pela amizade e suporte a todos que me acompanharam nessa jornada, meus colegas da melhor turma da Mecatrônica de todos os tempos, a Vigésima Sétima Turma de Engenharia Mecatrônica da Universidade de Brasília, especialmente, aos amigos Eric Torlig, De Hong Jung, Cristiana Miranda, Pedro Hecksher, Daniel Vincentin, Francisco Gomes, Vitor Caveira, Gabriel Cataldi, Matheus Takahashi, Filipe Caixeta, Marco Emílio, Jessé Barreto, Rodrigo Teixeira e Rafael Lima. Esse caminho foi extremamente divertido por causa de vocês. Agradeço também a todos membros do LARA, em particular, Lucas Rato, Roberto Baptista, Gabriel Moises, Vinicius Galvão e Thiago Rocha que de alguma forma me ajudaram ao longo desse trabalho. E agradeço a primeira equipe de futebol de robôs da UnB, a UnBall, onde aprendi os conceitos básicos da robótica e conheci pessoas que me levaram a sempre enfrentar desafios novos, particularmente, George Brindeiro, Tiago Pimentel, Lucas Levy, Matheus Portela e Guilherme Anselmo.

Agradeço todos meus amigos da vida, especialmente, ao Tiago Fonseca que conheço desde sempre e ao Vitor Barata Ribeiro que me inspirou a estudar Mecatrônica.

E acima de tudo, agradeço a todos da minha família que são essenciais para mim, especialmente, aos meus pais José Poli e Dácia pelos conselhos frequentes e por acreditar nas minhas escolhas e às minhas irmãs Joana e Cecília que estão sempre dispostas e presentes me apoiando e tornando a vida muito mais descontraída e agradável.

Marcos da Silva Pereira

RESUMO

A visão de robôs interagindo e ajudando pessoas a executar tarefas existe há bastante tempo. A evolução da tecnologia inseriu os primeiros robôs manipuladores nas fábricas para realizar tarefas repetitivas e perigosas para humanos. Apesar disso, as tarefas e o ambiente foram adaptados para o robô. Além disso, por questões de segurança, eles foram isolados dos humanos. O uso de robôs em fábricas trouxe vários benefícios, mas alguns desafios surgiram. As capacidades de repetibilidade e precisão dos robôs são elevadas, mas eles não são bons em se adaptar para novas tarefas e precisam sempre serem reprogramados. Por outro lado, os humanos têm alta capacidade de adaptação, mas não é recomendado trabalharem longos períodos realizando a mesma tarefa podendo sofrer lesões por esforço repetitivo. No futuro as máquinas vão ter que trabalhar cada vez mais perto dos humanos em ambientes não estruturados, logo, é preciso ser seguro para humanos trabalharem próximo delas. Nesse sentido, o campo de interação humano-robô (IHR) surgiu para integrar humanos e robôs no ambiente de trabalho e tornar possível realizar tarefas que nenhum dos dois conseguiria realizar sozinhos. Para aumentar a segurança, robôs manipuladores complacentes que permitem ajuste da rigidez das juntas são utilizados. No entanto, o uso seguro e o controle eficiente deles ainda é um desafio a ser resolvido em robótica. Logo, neste trabalho controladores de trajetória recentemente desenvolvidos na literatura são implementados e testados em um robô manipulador complacente e um ambiente de trabalho (*framework*) para utilizar o robô é desenvolvido.

O controle de um robô para realizar tarefas requer sua descrição cinemática de corpo rígido. Existem diferentes métodos para representações de corpos rígidos nas quais a orientação e a translação são tratadas de forma desacoplada. Apesar disso, tais representações não consideram o acoplamento cinemático completo. Isso pode gerar descrições incompletas causando problemas como, por exemplo, singularidades no caso dos uso dos ângulos de Euler. Matrizes de transformação homogênea (MTH) são uma das ferramentas mais utilizadas que acoplam orientação e translação. Entretanto, elas têm um custo computacional elevado. Nesse sentido, os quatérnios duais vêm ganhando popularidade nas últimas quatro décadas. Eles realizam a maioria das operações envolvendo movimentações de corpos rígidos com menor custo computacional se comparado com a MTH. Além disso, eles podem ser usados diretamente nas leis de controle sem necessitar converter para outra representação como é o caso da MTH. Logo, quatérnios duais foram utilizados neste trabalho para representar corpos rígidos e implementar controladores de trajetória.

Os manipuladores complacentes são uma solução atrativa para aumentar a segurança e aproximar robôs de humanos. No entanto, em aplicações práticas, é preciso que eles entendam tarefas requisitadas pelos humanos, ou seja, precisam seguir trajetórias descritas no espaço de tarefas. Em outras palavras, se uma pessoa for pegar uma caneca de café em uma mesa, ela se preocupa para qual posição sua mão deve se mover e não como cada junta do seu braço deve se deslocar. Logo, os robôs devem deslocar seu efetuador para a posição especificada pelo humano no espaço de tarefas. Portanto, o presente trabalho implementou técnicas de controle cinemático projetadas no espaço de tarefas.

Dessa forma, foram implementados seis controladores cinemáticos baseados em quaternários duais para analisar seu desempenho em um robô real e contribuir para seu futuro uso em aplicações de IHR. Eles foram avaliados extensamente no manipulador complacente antropomórfico A2 Arm da Meka Robotics com 7 graus de liberdade disponível no Laboratório de Automação e Robótica (LARA) da Universidade de Brasília (UnB). Os controladores foram implementados em C++ e a estabilidade numérica foi verificada no simulador *Virtual Robot Experimentation Platform* (V-REP). A integração das plataformas robô, computador e simulador foi feita com ROS. As especificações técnicas do robô A2 Arm foram detalhadas ao longo deste trabalho e o framework para facilitar o desenvolvimento de trabalhos futuros foi estabelecido.

Dois controladores proporcionais com e sem termo de antecipação apresentaram desempenho aceitável em relação a norma do erro com um ajuste simples de ganho, apesar disso só possuem um parâmetro que otimiza o erro sem considerar as velocidades das juntas e do efetuador. Entretanto, em IHR os manipuladores trabalham próximos aos humanos que devem se sentir confortáveis próximo dos robôs, logo, eles devem realizar trajetórias mais suaves com menores picos de velocidade. Portanto, foram implementados dois controladores do tipo regulador quadrático linear (RQL) que permitiram balancear entre velocidade do efetuador e esforço de controle além de ter apresentado erro inferior comparado aos controladores proporcionais. Por fim, para considerar perturbações externas e incertezas de modelo, dois controladores H_∞ foram implementados e forneceram o melhor desempenho de erro sem gerar velocidades excessivamente altas. A desvantagem dos controles H_∞ e RQL foi o ajuste de ganhos que em algumas situações precisou variar durante a trajetória. Os controladores também foram avaliados do ponto de vista do período de amostragem utilizado. São apresentados histogramas do período de amostragem para 20 ms e 8 ms. Percebeu-se que o módulo de controle conseguiu manter um período próximo do desejado. Entretanto, em trabalhos futuros podem ser explorados os módulos de controle de baixo nível do robô para facilitar a escolha do período de amostragem utilizados nos módulo de controle de trajetória de alto nível implementados.

Logo, este trabalho verificou que os controladores cinemáticos podem ser aplicados em um manipulador complacente real e forneceu resultados para comparar controladores sofisticados com outros métodos clássicos de controle. Além disso, gerou um ambiente de programação que facilitará futuros trabalhos no LARA envolvendo o robô A2 Arm. Portanto, conclui-se que, apesar da implementação dos controladores e do *framework* poderem ser melhoradas no futuro, houve sucesso no desenvolvimento do ambiente de trabalho e no uso dos controladores de trajetória para um robô real complacente.

Palavras Chave: robôs complacentes, controle cinemático, quaternários duais, interação humano-robô

ABSTRACT

This work implements six dual quaternion based kinematic control algorithms recently developed in literature to analyze their performance on a real robot and contribute to their use in future human-robot interaction (HRI) applications. They were extensively evaluated on the A2 Arm anthropomorphic compliant manipulator from Meka Robotics with 7 degrees of freedom available in the Robotics and Automation Laboratory (LARA) at the University of Brasília. The motivation for this work was to control a compliant manipulator which can work safely alongside humans in HRI tasks. Two proportional controllers with and without feedforward term showed a satisfactory performance regarding the error norm and, as expected, with simple gain adjustment. Nonetheless, they have only one parameter which seeks to reduce the error disregarding the overall joint velocities and end effector velocities along the trajectory. However, in HRI applications, manipulators will work near humans which must feel comfortable and safe near the robotic arms. Therefore, smoother end effector trajectories with lower velocity peaks are desired. Hence, two linear quadratic regulators (LQR) controllers which enable to reach a trade-off between trajectory error and end effector velocities are implemented. In addition to lower velocity peaks, they also had a better error performance compared to the proportional controllers. Lastly, to consider exogeneous disturbances and uncertainties, two H_∞ controllers are implemented and delivered the best error performance without an excessive increase in control effort. Their disadvantage together with the LQRs controllers was their parameter selection which in some cases needed to vary along the trajectory. The controllers are implemented in C++ and called in a ROS node. Thus they can be applied across platforms running ROS. In order to check the controllers numerical stability before testing them on the real robot, the controllers are tested on a Kuka LBR iiwa 7 R800 in the V-REP simulator. Consequently, this work combines development and evaluation results. The work verified the kinematic controllers can be applied to a compliant robot. Moreover, the contributions derived herein provides enough results to compare sophisticated control methods with classic controllers and also developed a programming framework for future projects at LARA involving the A2 Arm robot.

Keywords: compliant robots, kinematic control, dual quaternions, human-robot interaction

SUMÁRIO

1	INTRODUCTION	1
1.1	HISTORIC CONTEXTUALIZATION AND MOTIVATION	2
1.1.1	COMPLIANT ROBOT CONTROL	4
1.2	PROBLEM DESCRIPTION & PROJECT GOAL	5
1.2.1	WORK OVERVIEW	6
1.2.2	MATHEMATICAL AND FRAMEWORK TOOLS	7
1.2.3	RESULTS	8
2	MATHEMATICAL BACKGROUND	9
2.1	DUAL QUATERNIONS	9
2.1.1	QUATERNIONS	10
2.1.2	DUAL NUMBERS	11
2.1.3	DUAL QUATERNIONS	11
2.2	KINEMATIC MODELING, CONTROL AND ERROR CRITERIA	12
2.2.1	KINEMATIC CONTROL	13
2.2.2	DUAL QUATERNION KINEMATIC CONTROL	14
2.2.3	INVARIANT ERROR FUNCTION	15
2.3	CONCLUSION	16
3	KINEMATIC CONTROLLERS	17
3.1	PROPORTIONAL CONTROLLER	17
3.1.1	DECOPLED PROPORTIONAL CONTROLLER	19
3.2	PROPORTIONAL CONTROLLER WITH FEED-FORWARD TERM	20
3.3	DUAL QUATERNION BASED ROBUST H_∞ KINEMATIC CONTROL	20
3.3.1	H_∞ CONTROL IN DUAL QUATERNION SPACE	23
3.3.2	DUAL QUATERNION H_∞ CONTROLLERS	24
3.4	LINEAR QUADRATIC OPTIMAL CONTROL (LQR)	26
3.4.1	OPTIMAL DUAL QUATERNION BASED CONTROLLER IN \mathbb{R}^8 MANIFOLD	27
3.4.2	OPTIMAL QUADRATIC CONTROLLER IN DUAL QUATERNION SPACE	29
3.5	CONCLUSION	31
4	DEVELOPMENT AND IMPLEMENTATION	33
4.1	VIRTUAL ROBOT EXPERIMENTATION PLATFORM (V-REP)	33

4.2	A2 ARM FROM MEKA ROBOTICS	34
4.2.1	ACTUATORS	35
4.2.2	SOFTWARE	36
4.2.3	CONTROLLERS	37
4.3	ROBOT OPERATING SYSTEM (ROS)	38
4.4	DQ ROBOTICS AND EIGEN	39
4.5	PROGRAMMING FRAMEWORK	40
4.6	HOW TO START WORKING WITH THE MEKA ROBOT	43
4.7	CONCLUSION	43
5	EVALUATION AND RESULTS	45
5.1	TASKS DEFINITION	46
5.2	EVALUATION CRITERIA	47
5.3	PARAMETERS SELECTION	48
5.3.1	PROPORTIONAL CONTROLLER WITH AND WITHOUT FEEDFORWARD TERM	48
5.3.2	LQR CONTROLLERS	48
5.3.3	DUAL QUATERNION H_∞ CONTROLLERS	49
5.4	LINEAR QUADRATIC OPTIMAL CONTROL (LQR) PERFORMANCE ANALYSIS	49
5.5	DUAL QUATERNION H_∞ CONTROLLERS PERFORMANCE ANALYSIS	52
5.6	CIRCULAR TRAJECTORY	56
5.7	SPIRAL TRAJECTORY	60
5.8	SAMPLING TIME HISTOGRAM	66
5.9	TRAJECTORY CONTROL SIMPLE APPLICATION: ROBOT LIGHT PAINTING	68
5.10	CONCLUSION	70
6	CONCLUSION AND FUTURE WORKS	72
REFERÊNCIAS BIBLIOGRÁFICAS		74
ANEXOS		78
A	RIGID MOTIONS AND HOMOGENEOUS TRANSFORMATIONS	79
A.1	REPRESENTING POSITIONS	79
A.2	REPRESENTING ROTATIONS	82
A.3	PARAMETRIZATION OF ROTATIONS	84
A.4	HOMOGENEOUS TRANSFORMATION MATRIX	86
B	ROTATION MATRIX OF A QUATERNION	86
C	CIRCULAR AND SQUARED TRAJECTORY	87
D	LQR IN \mathbb{R}^8 IN DEPTH ANALYSIS PERFORMANCE	92
E	TABLES OF RESULTS	95
E.1	TABLES FROM CIRCULAR TRAJECTORY WITH $r = 0.001, q = 1$ —SECTIONS 5.4 AND 5.6	95
E.2	TABLES FROM CIRCULAR TRAJECTORY WITH $r = 0.01, q = 10$ —SECTIONS 5.4 AND 5.6	98

E.3	TABLES FROM CIRCULAR TRAJECTORY WITH $r = 0.1, q = 100$ —SECTIONS 5.4 AND 5.6.....	101
E.4	TABLES FROM CIRCULAR TRAJECTORY WITH $r = 1, q = 1000$ —SECTIONS 5.4 AND 5.6.....	104
E.5	TABLES FROM DUAL QUATERNION H_∞ CONTROLLERS PERFORMANCE ANAL- YSIS—SECTION 5.5.....	107
E.6	TABLES FROM SPIRAL TRAJECTORY—SECTION 5.7.....	109
E.7	TABLE FROM SAMPLING TIME HISTOGRAM—SECTION 5.8	139
	IV DESCRIÇÃO DO CONTEÚDO DO CD	140

LISTA DE FIGURAS

1.1	(a) Robot working on foundry and (b) ABB welding robots (source: ABB)	2
1.2	(a) Baxter from Rethink Robotics working together with worker (b) Kuka robot assisting welding process	3
1.3	Work diagram	7
4.1	Kuka LBR iiwa 7 used for simulations in V-REP	34
4.2	Meka robot available at LARA at UnB (A2 Arm from Meka Robotics)	35
4.3	Illustration of where the SEA are located at the robot. Note that the SEA in this figure is a general SEA example and not the specific SEA which the Meka robot uses.	36
4.4	Meka robot: continuously flow of information between higher level controllers (left) and lower level controllers (right).....	37
4.5	System modules overview.....	39
4.6	ROS computation graph integrating the Meka robot, the control node and the V-REP simulation environment.	40
4.7	Overview of the robot controlling node, the use of the DQ Robotics library and the Drawer class.....	42
5.1	Coordinate axes on the Meka robot used on the experiments.....	45
5.2	Gains adjusted to 20% (left) and 100% (right)	55
5.3	Circular trajectory with $q = 1$ and $r = 0.001$	59
5.4	Best gain selection: $k = 125$, $q = 270$, $r = 0.01$, $\gamma_T = 0.372$, $\gamma_O = 0.377$	65
5.5	Histogram of the time difference between each trajectory step for a sampling time of 20 ms. Mean values (left to right): 0.020, 0.020, 0.020, 0.021, 0.020, 0.020. Std. Dev.: 0.001 (See Appendix E.7 Table 61).	67
5.6	Histogram of time difference between each trajectory step for a sampling time of 8 ms. Mean values: 0.008, Std. Dev. values: 0.001 (See Appendix E.7 Table 61)	68
5.7	Kinematic controllers applied to light painting tasks.....	69
1	Two coordinate frames, a point p and vectors \mathbf{v}_1 and \mathbf{v}_2	79
2	Baxter robot with world coordinate frame w (source: Rethink Robotics).	80
3	Two coordinate frames, a point p and vectors \mathbf{v}_1^0 , \mathbf{v}_2^0 , \mathbf{v}_1^1 , \mathbf{v}_2^1	80
4	Kuka LBR iiwa 7 on simulation environment.	81
5	Circular trajectory with $r = 1$ and $q = 1000$	90
6	Square trajectory with $r = 1$ and $q = 1000$	91

7	LQR parameters surface on Kuka LBR iiwa 7 on V-REP	93
8	LQR parameters surface on Meka robot	94

LISTA DE TABELAS

3.1	Kinematic controllers equations overview	32
5.1	Overview of all experiments.....	46
5.2	Overview of experiment to evaluate the LQR controller performance	50
5.3	Trajectory results obtained on Meka performing the circular end effector trajectory described by (5.7) with the LQR in \mathbb{R}^8 : $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s ²]	51
5.4	Trajectory results obtained on Meka performing the circular end effector trajectory described by (5.7) with the LQR in DQ: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s ²]	51
5.5	Overview of experiment to evaluate the H_∞ controller performance	52
5.6	Trajectory results obtained on Meka performing the spiral end effector trajectory described by (5.10) with the HIR controller: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s ²]	53
5.7	Trajectory results obtained on Meka performing the spiral end effector trajectory described by (5.10) with the HIRT controller: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s ²]	54
5.8	Overview of the experiment to compare all controllers using a circular trajectory.....	56
5.9	Trajectory results obtained on Meka performing the circular end effector trajectory described by (5.7): $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s ²]	57
5.10	Overview of the experiment to compare all controllers using a spiral trajectory	60
5.11	LQR in \mathbb{R}^8 controller gain variations	61
5.12	H_∞ gains variation	61
5.13	Trajectory results obtained on Meka performing the spiral end effector trajectory described by (5.12) with all controllers set to $(\ln 2)^2$ of performance: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s ²]	62
5.14	Trajectory results obtained on Meka performing the spiral end effector trajectory described by (5.12) with all controllers set to $\ln 2$ of performance: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s ²]	62
5.15	Trajectory results obtained on Meka performing the spiral end effector trajectory described by (5.12) with all controllers set to the best performance defined: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s ²]	63
5.16	Overview of experiments sampling time	66
5.17	Overview of controllers evaluation.....	71

1	Overview of the circular trajectory experiment to compare the proportional controllers with the LQR in \mathbb{R}^8 controller.....	87
2	Overview of the square trajectory experiment to compare the proportional controllers with the LQR in \mathbb{R}^8 controller.....	88
3	Simulation results obtained in V-REP performing two different end effector trajectories on the Kuka LBR iiwa 7.....	88
4	Trajectory results obtained on Meka performing two different end effector trajectories.	89
5	Overview of the circular trajectory experiment to evaluate the LQR in \mathbb{R}^8 controller behavior on the Meka robot.....	92
6	LQR controller with $r = 0.001, q = 1$: Integral norm of the end-effector invariant error.....	96
7	LQR controller with $r = 0.001, q = 1$: Control output $\dot{\theta}$	96
8	LQR controller with $r = 0.001, q = 1$: Joints Control Effort Norm (Eff.).....	97
9	LQR controller with $r = 0.001, q = 1$: End effector linear velocity (L. Vel.).....	97
10	LQR controller with $r = 0.001, q = 1$: End effector linear acceleration (L. Acc.).....	98
11	LQR controller with $r = 0.01, q = 10$: Integral norm of the end-effector invariant error.....	99
12	LQR controller with $r = 0.01, q = 10$: Control output $\dot{\theta}$	99
13	LQR controller with $r = 0.01, q = 10$: Joints Control Effort Norm (Eff.).....	100
14	LQR controller with $r = 0.01, q = 10$: End effector linear velocity (L. Vel.).....	100
15	LQR controller with $r = 0.01, q = 10$: End effector linear acceleration (L. Acc.).....	101
16	LQR controller with $r = 0.1, q = 100$: Integral norm of the end-effector invariant error.....	102
17	LQR controller with $r = 0.1, q = 100$: Control output $\dot{\theta}$	102
18	LQR controller with $r = 0.1, q = 100$: Joints Control Effort Norm (Eff.).....	103
19	LQR controller with $r = 0.1, q = 100$: End effector linear velocity (L. Vel.).....	103
20	LQR controller with $r = 0.1, q = 100$: End effector linear acceleration (L. Acc.).....	104
21	LQR controller with $r = 1, q = 1000$: Integral norm of the end-effector invariant error.....	105
22	LQR controller with $r = 1, q = 1000$: Control output $\dot{\theta}$	105
23	LQR controller with $r = 1, q = 1000$: Joints Control Effort Norm (Eff.).....	106
24	LQR controller with $r = 1, q = 1000$: End effector linear velocity (L. Vel.).....	106
25	LQR controller with $r = 1, q = 1000$: End effector linear acceleration (L. Acc.).....	107
26	H_∞ controller behavior: Integral norm of the end-effector invariant error.....	108
27	H_∞ controller behavior: Control output $\dot{\theta}$	108
28	H_∞ controller behavior: Joints Control Effort Norm (Eff.).....	108
29	H_∞ controller behavior: End effector linear velocity (L. Vel.)	108
30	H_∞ controller behavior: End effector linear acceleration (L. Acc.).....	109
31	Integral norm of the end-effector invariant error: spiral trajectory with $(\ln 2)^2$ of the best performance gains.....	110
33	Control output $\dot{\theta}$: spiral trajectory with $(\ln 2)^2$ of the best performance gains.....	112

35	Joints Control Effort Norm (Eff.): spiral trajectory with $(\ln 2)^2$ of the best performance gains.....	114
37	End effector linear velocity (L. Vel.): spiral trajectory with $(\ln 2)^2$ of the best performance gains.....	116
39	End effector linear acceleration (L. Acc.): spiral trajectory with $(\ln 2)^2$ of the best performance gains.....	118
41	Integral norm of the end-effector invariant error: spiral trajectory with $(\ln 2)$ of the best performance gains.....	120
43	Control output $\dot{\theta}$: spiral trajectory with $(\ln 2)$ of the best performance gains.....	122
45	Joints Control Effort Norm (Eff.): spiral trajectory with $(\ln 2)$ of the best performance gains.....	124
47	End effector linear velocity (L. Vel.): spiral trajectory with $(\ln 2)$ of the best performance gains.....	126
49	End effector linear acceleration (L. Acc.): spiral trajectory with $(\ln 2)$ of the best performance gains.....	128
51	Integral norm of the end-effector invariant error: spiral trajectory with best performance gains.....	130
53	Control output $\dot{\theta}$: spiral trajectory with best performance gains.....	132
55	Joints Control Effort Norm (Eff.): spiral trajectory with best performance gains.....	134
57	End effector linear velocity (L. Vel.): spiral trajectory with best performance gains.....	136
59	End effector linear acceleration (L. Acc.): spiral trajectory with best performance gains.....	138
61	Sampling time histograms mean values and standard deviation.....	139

LISTA DE SÍMBOLOS

Latin Symbols

\mathcal{O}	Coordinate frame
R	Rotation matrix
F, H	cost functions

Greek Symbols

γ	upper bound of the induced norm in the context of robust H_∞ control
γ_T	upper bound of the translational error induced norm
γ_O	upper bound of the orientation error induced norm
ω	rigid body twist
v	uncertainties and disturbances

Subscripts

E	variable related to end effector
-----	----------------------------------

Superscripts

\rightarrow	represent vector form of variable
\cdot	time derivative
$^+$ or \dagger	the Moore-Penrose pseudoinverse
$^{-1}$ or inv	inverse of a matrix

Basic Symbols and notations

a, b, c, \dots	scalars are represented by lowercase plain letters
$\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$	vectors are represented by lowercase bold letters
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \dots$	matrices are represented by uppercase bold letters
\mathbf{I}	the identity matrix with appropriate dimensions
$sgn(\bullet)$	the signal of a given scalar
$\ \bullet\ $	the generic norm of a function
$\ \bullet\ _2$	the induced norm $p = 2$ of a function (also called Euclidean norm)
$diag(\mathbf{A}_1, \dots, \mathbf{A}_n)$	block diagonal matrix with the diagonal elements being the matrices $\mathbf{A}_1, \dots, \mathbf{A}_n$

Quaternion and dual quaternions symbols and notations

$\hat{i}, \hat{j}, \hat{k}$	ordinary complex imaginary units (quaternionic units)
ε	nilpotent dual unit
$\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$	quaternions and pure quaternions are represented by lower case bold letters
$\underline{\mathbf{a}}, \underline{\mathbf{b}}, \underline{\mathbf{c}}, \dots$	dual quaternions and pure dual quaternions are represented by lowercase bold underlined letters
$\mathbf{a}^*, \underline{\mathbf{a}}^*$	the conjugate of quaternion \mathbf{a} and dual quaternion $\underline{\mathbf{a}}$, respectively
$\underline{\mathbf{e}}$ and $\underline{\mathbf{x}}_e$	invariant error function and dual quaternion spatial difference, respectively
$\underline{\mathbf{x}}_E$	forward kinematic model in the unit dual quaternion group
$Re(\bullet)$	the real component of the quaternion $\mathbf{a} = Re(\mathbf{a}) + Im(\mathbf{a})$ or of the dual quaternion $\underline{\mathbf{a}}$
$Im(\bullet)$	the imaginary component of the quaternion $\mathbf{a} = Re(\mathbf{a}) + Im(\mathbf{a})$ or of the dual quaternion $\underline{\mathbf{a}}$
$vec_3\mathbf{a}$,	the vector space isomorphism mapping from pure quaternions and quaternions to \mathbb{R}^3
$vec_4\mathbf{a}$	the vector space isomorphism mapping from pure quaternions and quaternions to \mathbb{R}^4
$vec_6\underline{\mathbf{a}}, vec\underline{\mathbf{a}}$	the vector space isomorphism mapping from pure dual quaternions and dual quaternions to \mathbb{R}^6 and \mathbb{R}^8 , respectively
\underline{vec}	the inverse vector space mapping from \mathbb{R}^n
$\mathbf{H}^\dagger(\underline{\mathbf{a}}) \mathbf{H}(\underline{\mathbf{a}})$	the Hamilton operator (matrix form of the algebra product) for a dual quaternion $\underline{\mathbf{a}}$

Jacobians

\mathbf{J}_w	task-space analytical jacobian
\mathbf{J}	task-space jacobian where $\mathbf{J} \in \mathbb{R}^{8 \times n}$
\mathbf{J}_ω	task-space dual quaternion Jacobian

Kinematic controllers

K, K+FF	proportional controller without and with feedforward term, respectively
LQR in \mathbb{R}^8	linear quadratic optimal controller based in \mathbb{R}^8 manifold
LQR in DQ	linear quadratic optimal controller based in dual quaternion space
HIR	H-infinity Robust controller
HIRT	H-infinity Robust controller with tracking term

Acronyms and Abbreviations

HRI	<i>Human-Robot Interaction</i>
Meka	A2 Arm from Meka Robotics
V-REP	<i>Virtual Robot Experimentation Platform</i>
SEA	Series Elastic Actuator
HTM	<i>Homogeneous Transformation Matrix</i>
FKM	Forwards Kinematic Model
DOF	Degrees of Freedom
LARA	<i>Robotics and Automation Laboratory at the University of Brasília</i>
UnB	<i>University of Brasília</i>

Chapter 1

Introduction

The vision of robots interacting and helping people to execute tasks is around here for a long time already. Technological evolution brought the first robot manipulators to factories to perform jobs which are repetitive and dangerous for humans. In order to deploy robots in the factory floor, tasks and environment were adapted for them. However, in the future, machines will have to work even closer to humans in an unpredictable environment. Therefore, this work is focused on one of the many challenges involved in making robot arms execute tasks in the presence of a human.

Innovation in robotics is encouraged by a wide number of sources and one of them is human-robot interaction (HRI) which aims factory and non-factory environments. New technological improvements appear even in educational contexts. Robots are also used by teachers to motivate students through ludic activities. The field of medical robotics also emerged where robots are developed on a collaboration between roboticists and medical doctors to be applied in different processes, for instance, in rehabilitation. In this sense, today, robots are being developed in many fields and applied in distinct situations and not only by specialized factory technicians to be used in factory environments [1]. Therefore, it is also necessary for them to interact safely with humans and one of the ways to do it is to use compliant joints and motors in the manipulators [2].

In applications which require HRI for object manipulation it is common to use anthropomorphic robotic arms because they are designed to move the same way humans do. Thus, there are fewer modifications to be done in the environment for them to accomplish their tasks.

Although the use of robots in manufacturing comes with benefits, they are still seen, in some cases, as a replacement for human workers which could reduce the number of jobs in factory. The HRI research field appeared to integrate both humans and robots in the workplace and turn possible to do things which a robotics-only or human-only team would not be able to do. The robots used for HRI require robust methods and tools in order to deliver acceptable results. Hence, improvements make easier the implementation of robotic manipulator controllers and integrate them to computer vision, force and positioning sensors.

Section 1.1 will briefly review the historic context and motivation for introducing robots in the human workplace. Afterwards, the challenge and goals of this project are presented in Section 1.2, followed by a description of the framework used in this work and the obtained results.

1.1 Historic Contextualization and Motivation

Since the term robot appeared, robotics has been imagined as robots cooperating and coexisting with humans in everyday life. However, they have been restricted for a long time to factories and laboratories. If you visit one of them today you will still encounter robot manipulators isolated from researchers or workers. They may be too dangerous to be around with and need to be kept at a safe distance [3].

In 1961, the first industrial robot from Unimation Inc. was installed for extracting parts from a die-casting machine at an industrial plant [4]. Then, hydraulically actuated robots were used for workpiece handling and spot-welding of car bodies [5]. The results were good enough for each application and have shown robots could offer reliability and quality together with uniformity. From then on, other industrial companies began to implement robots on their production and an innovating process started. Although reliability had been verified, there were still safety problems to be solved.



(a)Kuka KR500¹

(b)

Figure 1.1: (a) Robot working on foundry and (b) ABB welding robots (source: ABB)

One of the motivations for using robots in factories was to pull out humans from hazardous work environments. Moreover, robots achieve high accuracy and speed, and work long periods without tiring if correct maintenance procedures are periodically done. Nonetheless, safety concerns also emerged because of robots potential to cause accidents involving humans. Therefore, early technical texts about robotic safety recommended isolating robots from personnel, but new issues arise with this approach.

Although robots are reliable and precise, there are still tasks which robots cannot do because of their minimal sensing and interpreting capability in comparison to humans. For every new task they must be reprogrammed which is not always straightforward. Meanwhile, humans adapt easily to new situations and problems. For instance, the products on the supermarket shelves are constantly changing places, sizes and positions and a person can easily rearrange a wide range of different objects while a robot must be reprogrammed for each new object. For this reason,

¹<http://www.roboworld.com/high-temperature>

people still injure themselves in daily activities because of repetitive material handling which causes work-related musculoskeletal disorders [2]. If robots are safe enough to be around with and easily reprogrammed by people, the flexibility in the work environment can be clearly increased. For instance, in factories where different materials constantly arrive in large quantities, a human could always adapt the robot manipulator to unload each of them instead of repeating a lot of similar tiresome unloading tasks. In this sense, the field of physical human-robot interaction (HRI or pHRI) appeared.

The HRI robots will empower humans to develop new ideas and accomplish them with the accuracy of robots without worrying about repetitive procedures. HRI robots are focused on extending workers ability and not on replacing them. Thus, they help to overcome the skepticism about robots replacing human jobs. In order to achieve this, safety become of paramount importance.

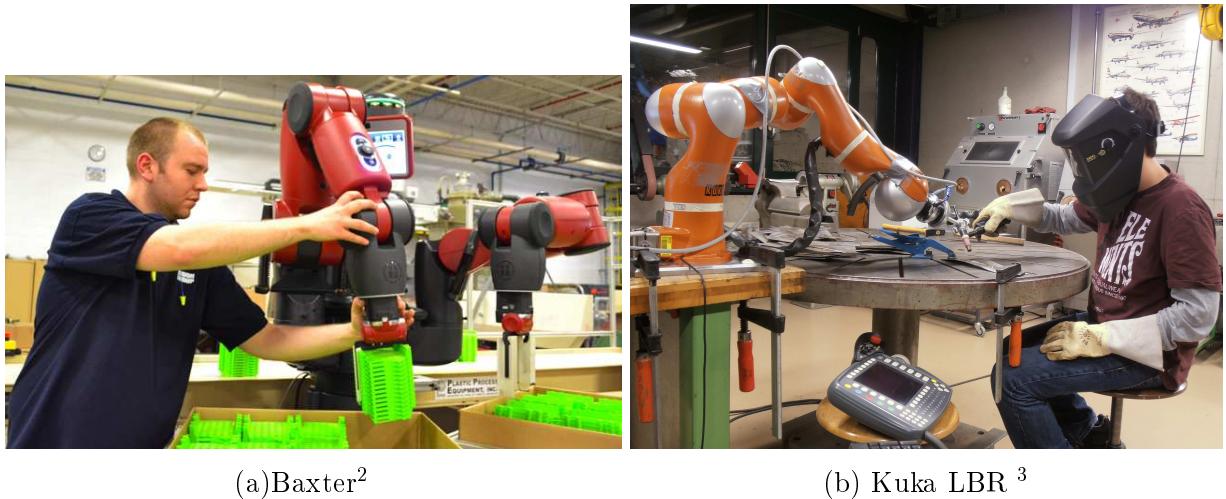


Figure 1.2: (a) Baxter from Rethink Robotics working together with worker (b) Kuka robot assisting welding process

Safety in robotics involve a lot of considerations. In [6], an extensive review was made of published literature on various aspects of robot reliability and safety. In HRI, two distinct cases are commonly mentioned: first, when physical contact occurs occasionally or by accident, and second, when physical interaction is an intentional part of planned operations [2].

This paradigm shift from precision, speed and error performance to safety comes together with new challenges in the design of control methods. Classic industrial robots are made to be as stiffer as possible in order to be more precise without the need of complex control methods. However, one way to increase safety is to increment the energy-absorbing properties of the robot structure, hence, decreasing stiffness. This has been done since the early 1980s by equipping the manipulator with protective layers and designing active force controllers or, stiffness and impedance control with respect to sensed interactions. However, this approach does not work for portions of the robot manipulator which do not have sensors [2]. Therefore, safety use and control of robotic manipulators is still a challenge to be efficiently solved in robotics.

²<http://www.rethinkrobotics.com/blog/meet-sally-applin-humans-and-robots/baxter-teach-onexia/>

³http://lasa.epfl.ch/tutorial_erra/welding_assist_kuka.jpg

It is recognized in robotics that the controller has intrinsic limitations to what it can do to change a robot arm behavior. This is closely related to the manipulator mechanical properties. Although possible, it is not simple to make a heavy robot gently, safe and precise, if realistic circumstances are considered. With this in mind, lightweight robots for service applications began to appear, first the whole-arm manipulator (WAM) from the Barret Technology Inc. and later the lightweight robot (LWR) from the german aerospace center (Deutsches Zentrum für Luft- und Raumfahrt - DLR).

Lightweight robot arms use high-performance actuators and integrated torque sensors. Significant joint compliance is usually present because of design choices such as use of cable transmissions, harmonic drives and joint torque sensors. Therefore, introducing mechanical compliance into the design of HRI robots has turned to be an effective way to increase safety for service application robot arms and has been used in numerous implementations in different contexts in robotics.

1.1.1 Compliant Robot Control

One primary skill that robots lacks compared to biological systems is adaptable compliance or variable stiffness. The dynamic decoupling of robots actuators rotor inertia from the links whenever an impact occur complements the low-inertia design of manipulators links and soft coverings. This helps to mimic the desired mechanical compliance. The HRI tasks usually require the combination of motions which vary from fully stiff to fully compliant. They may vary depending on the goal. In [7], Ahmed divided the types of motions in three categories:

1. Stiff motion: refers to robot movement in free space referred as unconstrained free work space. The manipulator trajectory within its work-space is done by position and velocity control. It has no compliance variation and, hence, is not good enough for executing constrained motions with HRI.
2. Soft motion: relates to robot movement constrained by an environment referred as constrained work-space. The sudden, unexpected intrusion of an obstacle makes collision unavoidable. Therefore, the robot behavior is changed from fully stiff to fully compliant.
3. Compliant motion: motions which incorporate all transitions between stiff and soft motion. It appears in HRI tasks, where a human want to impose its motion over the robot's specified motion. In this case, variable compliance in the robot is required and achieved through compliant mode.

Then, in order to effectively use compliant robots, some issues have to be solved. Compliant transmissions can reduce performance, causing slow response, increased oscillations and longer settling times. Nevertheless, for practical applications, positioning accuracy and velocity of task execution are crucial. The control of soft manipulators quickly and accurately is one of the challenges open in front of HRI research.

In [2], a extensive review of robotics literature regarding mechanical joint designs to achieve compliant control of robot manipulators was done. The control of robots with elastic joints was ex-

plored in [8, 9]. Robot safety achieved through safety oriented design using series elastic actuators appeared in [10]. Adaptive control methods were also employed to endure uncertain inertia caused by elastic joints. Moreover, other compliant actuation mechanisms were developed such as distributed macro-mini actuation [11] which separates torque generation in low- and high-frequency components and sum them in parallel to deliver a better performance while reducing the manipulators inertia.

Even though elastically actuated arms have high enough compliance to be safe, there are intrinsic limitations to what can be achieved by such design although optimal controlling methods exists for them. With this in mind, it was also explored in [2] techniques which allow mechanical compliance of motion transmission devices to vary during the execution of tasks. In other words, the robot joints of a manipulator may be adjusted to be as stiff or soft as needed at a given moment. This is comparable to what humans do to behave in a safe manner: the human arm moves slowly when it is stiff and compliantly when moving fast.

Variable compliance enables to achieve the safety-performance trade-off using mechanical means and control strategies. In this sense, numerous implementations have emerged such as the mechanical impedance adjuster (MIA) in [12] which enables to choose the compliance level in a wide range of values accordingly to the desired task. On the other hand, the compliance must be chosen before the execution of the task and remains constant during motion. Another solution was proposed in [13] which allows variable-impedance actuation (VIA) during task execution. This is achieved through a combination of mechanical and control design allowing rapid and continuous variations of transmission impedance. Therefore, the injury risk during all motion may be reduced by adjusting stiffness, damping and gear ratio while minimizing negative effects on control performance. Furthermore, it allows to adjust the overall performance to accomplish as quickly and accurately as possible to achieve the desired task specifications. In this sense, VIA design suits well the challenge of executing compliant motions adjusting between stiff and soft motions as needed.

1.2 Problem Description & Project Goal

Compliant robot control techniques have been a compelling way to increase safety and bring robot manipulators closer to humans. However, in order to use robots effectively in practical applications, one must also be able to devise tasks for the robot. It is natural for humans to execute trajectories in the task-space instead of the joint space. In other words, when a person is picking up a coffee cup of a table, the person focus to which position in space their hand must move and not on how each arm joint will move. The same goes for robots working alongside humans. They must be able to work with their human work partners solving tasks in a similar way.

This brings up the problem investigated in this work, which is how to make a robot manipulator execute a trajectory specified in the task-space. Different kinematic control techniques for robot manipulators have been presented in the literature. However, they are still being evaluated on real robots. Recent kinematic control techniques in the task-space were developed in [14] which were based on the complete framework for robotic kinematic modeling formalized in [15]. However, the

controllers designed in [14] were only fully explored in simulation environments. Moreover, some of the experiments were done in simulation tools without a physics engine to simulate rigid body dynamics.

In this sense, the goal of this work is to implement and evaluate for the first time on a real compliant anthropomorphic manipulator some of the kinematic control techniques in the task-space developed in [14]. In other words, evaluate them in the real world scenario with multiple sources of disturbances and uncertainties and, in addition to compare their performance with other classic control methods. Furthermore, they will be tested under high amount of compliance which turns kinematics-only control strategies way more challenging and demanded. The kinematic control techniques use only the robot joint encoders position as input and no force feedback. Impedance and force control helps to solve this issue [16, 17, ch. 9], but they need the robot dynamic model which is seldom straightforward to obtain [16, 17, ch. 7].

For all the above reasons and challenges posed, this work contributes to the use and further development of kinematic control techniques for compliant robots in practical applications.

1.2.1 Work Overview

The present chapter brought up the HRI scenario and its challenges. The first section explained how robots emerged in the industrial environment to execute repetitive and dangerous tasks and then were seen as a solution to help and empower humans in the workplace, but with a lot of safety issues to be solved. As a consequence, the compliant robots concept appeared as a solution. Then, the problem description and goal of this work was defined.

Including this introduction chapter, this work is organized in six chapters. In Chapter 2, the mathematical background of the dual quaternions is presented, a basic kinematic controller is derived and the invariant error norm is described. The Chapter 3 shows the mathematical derivation of the kinematic controllers evaluated in this work. Chapter 4 describes the robotic tools and development of the programming framework used for implementing and testing and how to use it. In Chapter 5, the tasks definition procedure, evaluation criteria and parameter selection are explained. Then, the experiments are described and their results are given and analyzed. Lastly, Chapter 6 contains a conclusion and future work ideas to enhance the kinematic controllers performance on the A2 Arm.

For the sake of giving a clearer understanding of how this text will be developed, the following diagram in Figure 1.3 was devised:

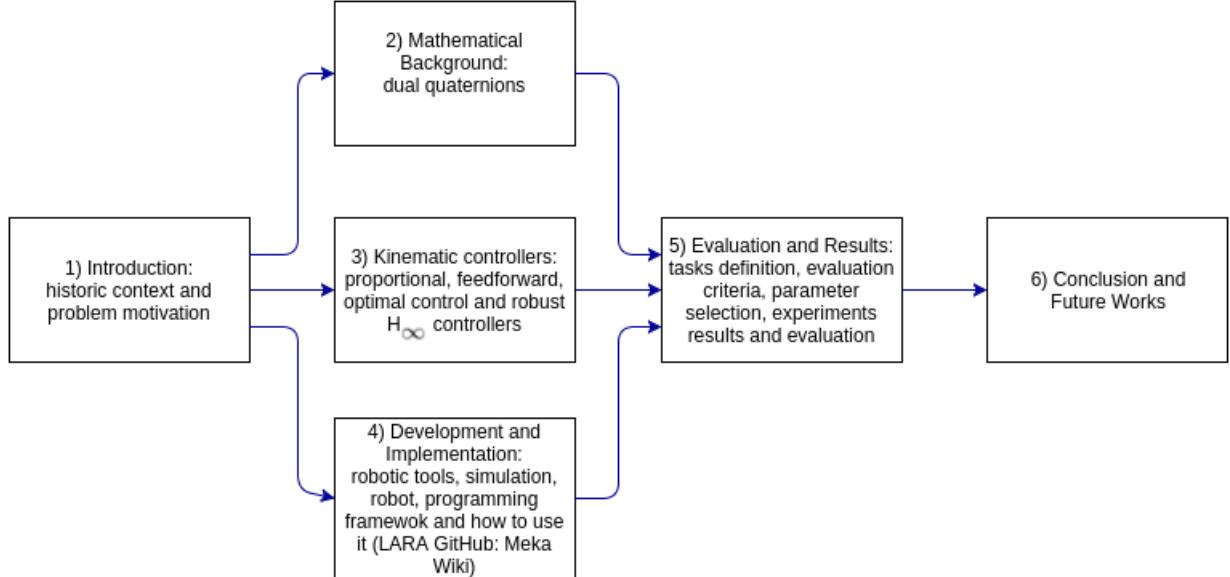


Figure 1.3: Work diagram

1.2.2 Mathematical and Framework Tools

To achieve good results with compliant control techniques, one must first solve the robot kinematic control problem. In this sense, an efficient representation for rigid body transformation must also be chosen. In this work, the dual quaternions were used due to its advantages in comparison to the homogeneous transformation matrices (HTM). For instance, as presented in [15], most of the mathematical operations of rigid body motion with dual quaternions are achieved with lesser calculations. Moreover, they do not have representation singularity problems in comparison to the Euler angles representation. Furthermore, they can be directly used in the control law design without the need of extracting parameters as it is the case of the HTM. The dual quaternions will, therefore, be detailed in Chapter 2.

Closely intertwined with the rigid body mathematical representation, the control method used to solve forward- and inverse kinematics also plays an important role. Classic proportional and feedforward controllers for robot manipulators deliver acceptable results. However, their performance depends only on their one-parameter gain adjustment. Hence, the control effort together with the robot joints velocity and, consequently, the end effector linear velocity are bounded to this one gain value.

In other words, if one wish to reduce the robot end effector trajectory error, one may increase the proportional gain. The robot motion velocity and the motors power will also increase. Then it is not possible to adjust each performance specification isolated. For this reason, this work implements two linear-quadratic optimal tracking controllers which allows to minimize end effector velocities in the sense of joints control effort while improving trajectory error [18].

Although the classic proportional and feedforward controllers are simple to be implemented, they do not consider the influence of multiple external disturbances acting on the system which

also causes uncertainties in the kinematic model of the robot. Therefore, robust H_∞ kinematic controllers were also implemented and investigated. Their goal is to reduce the noise-to-output influence and, therefore, reduce the effect of external disturbances while keeping the system internal stability.

The real robot experimentation platform used for this work was the anthropomorphic compliant manipulator A2 Arm from Meka Robotics available at the Robotics and Automation Laboratory (LARA) at the University of Brasília (UnB). In order to prevent causing damage to the robot, the control methods were first run on the simulation environment Virtual Robot Experimentation Platform (V-REP). Finally, to integrate all the systems the Robot Operating System (ROS) was adopted. This tools will be explained in Chapter 4.

1.2.3 Results

This work outcomes are divided in development and evaluation results. The programming framework used to implement and test the dual quaternion based kinematic controllers turns easier future works and applications involving the A2 Arm robot to be implemented at LARA. The main controller ROS node written in C++ enables to load a desired trajectory specified in the task-space from a *.txt* file or to get a discrete trajectory specified point-to-point in Cartesian coordinates. Afterwards, the trajectory Cartesian coordinates can be used as the input of the kinematic controllers which deliver the robot joints trajectory output. The programming framework documentation can be found in [19], that is, the Wiki created at the LARA GitHub¹ page for the Meka project². More details on how to start working with the framework will be presented in Chapter 4.

As a result of the control techniques evaluation on the real robot, it was seen the dual quaternion based kinematic controllers can be used for trajectory control on a compliant robot manipulator. They enable the A2 Arm robot to be used in applications which accepts precision of 1cm. Depending on the tasks which must be done, each controller has its advantages. Proportional controllers can do multiple lower precision tasks without the need of online gain tuning. Optimal controllers can be adjusted for repetitive similar tasks which demand higher precision with lower end effector velocities. Robust H_∞ controllers can be adjusted for high precision tasks without an excessive increase in robot joints velocities but they need a more cautious gain adjustment including online gain variation.

¹<https://github.com/>

²<https://github.com/lara-unb/Meka/wiki>

Chapter 2

Mathematical Background

In order to plan and control robot motions to perform specified tasks, it is needed to model them mathematically. A lot of rigid motion mathematical representations detach and individually address attitude and translational kinematics or, in other words, orientation and position of the rigid body are treated separately [14, ch.2]. Hence, they do not consider the coupling between the full rigid motion kinematics which include both orientation and position. This may yield improper descriptions of rigid body motion. The homogeneous transformation matrices (HTMs) are one of the most used ones to represent rigid body motion which couples orientation and position. Still, they are computational demanding. Another representation which has been gaining attention in the last 30 to 40 years are the quaternions and dual quaternions due to their properties. The dual quaternions, more specifically, do most of the rigid body motion operations with lesser calculations and it is straightforward to extract geometric parameters while HTMs requires additional calculations in order to extract control parameters [14, ch. 2]. Moreover, they do not have representation singularity problems as in the case of Euler angles. They also allow to use the same set of variables to represent the forwards kinematic model and perform robot control [15], henceforth, they were adopted in this work.

In this chapter the dual quaternions are explored and associated with rigid body motion of robotic manipulators. Afterwards, the basic concepts of differential kinematic control for robot manipulators using unit dual quaternions are presented. A background on rigid motion and HTMs can be found in Appendix A.

2.1 Dual Quaternions

Dual quaternions are dual numbers in which the primary and dual parts are quaternions [15]. Before getting into details about them, a brief introduction to dual numbers and rigid body motion using quaternions is done.

2.1.1 Quaternions

Quaternions were invented by Hamilton in the nineteenth century and are an extension of the complex numbers [20]. They have three imaginary components \hat{i} , \hat{j} and \hat{k} which define the quaternions set, such that

$$\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1.$$

A general quaternion $\mathbf{h} = h_1 + \hat{i}h_2 + \hat{j}h_3 + \hat{k}h_4$ and its conjugate is defined as $\mathbf{h}^* \triangleq h_1 - h_2\hat{i} - h_3\hat{j} - h_4\hat{k}$. The norm of a quaternion is $\|\mathbf{h}\| = \sqrt{\mathbf{h}\mathbf{h}^*}$.

The rotation of an angle θ around an axis $\mathbf{n} = n_x\hat{i} + n_y\hat{j} + n_z\hat{k}$ can be described by the unit quaternion $\mathbf{r} = \cos(\frac{\theta}{2}) + \mathbf{n}\sin(\frac{\theta}{2}) = \eta + \mathbf{n}\epsilon$ where $\|\mathbf{r}\| = 1$. Sequential rotations $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, \mathbf{r}_n$ are described by sequential quaternion multiplications $\mathbf{r}_1 \cdot \mathbf{r}_2 \cdot \mathbf{r}_3 \cdot \dots, \mathbf{r}_n$.

Analogous to the rotation matrices $R \in SO(3)$ (Appendix Section A.2), the rotation quaternion \mathbf{r} also has geometrical meanings [14, ch. 2, p.21]:

1. It represents a coordinate transformation relating the coordinates of a point \mathbf{p} in two different frames. Considering the pure quaternions \mathbf{p}^0 and \mathbf{p}^1 represents \mathbf{p} with respect to frames 0 and 1, respectively. If frame 1 is obtained by rotating frame 0 by a quaternion \mathbf{r}_1^0 , the representation of the point in frame 1 is given by the frame rotation transformation,

$$\mathbf{p}^1 = \mathbf{r}_1^{0*} \mathbf{p}^0 \mathbf{r}_1^1.$$

2. If \mathbf{p}_0^0 is a point with respect to a coordinate frame 0. The representation of the new point \mathbf{p}_1^0 after rotating within the same coordinate frame is given by the point rotation transformation,

$$\mathbf{p}_1^0 = \mathbf{r}_1^0 \mathbf{p}_0^0 \mathbf{r}_1^{0*}.$$

3. The quaternion \mathbf{r}_1^0 gives the frame rotation between frames 0 and 1.

The challenges which arise when using the Euler angle/axis representation mentioned in the end of Appendix Section A.3 can be overcome using the unit quaternion representation. A rotation by $-\theta$ about $-\mathbf{r}$ yields the same quaternion as that associated with a rotation by θ about \mathbf{r} . This solve the nonuniqueness problem of the angle/axis description [17, ch. 2, section 2.6]. On the other hand, the unit quaternion representation is liable to the unwinding phenomenon where the attitude of the rigid body may contrast with the antipodal representation yielding in unnecessary rotation. It is interesting to highlight that tiny representation presents topological issues.

Unit quaternions are then a four-parameter rotation representation constrained to unit norm. They have a unique algebra and enable to represent all rotations with lesser parameters than a rotation matrix. Moreover, the representation singularities which interfere in the use of minimal rotation representations are solved [21, p. 24].

2.1.2 Dual Numbers

Dual numbers were introduced by Clifford in the nineteenth century, who proposed the dual unit ε to create a new algebra over the real numbers in which, ε has the following properties:

$$\begin{aligned}\varepsilon &\neq 0 \\ \varepsilon^2 &= 0\end{aligned}$$

In a dual number $\underline{a} = a + \varepsilon a'$, a is the primary part and a' the dual part. Both parts are of the same type of elements. The usual operations of sum, subtraction and multiplication consider the ε operator and are defined [22]. The dual numbers are not a division algebra, hence, the inverse is only defined when $a \neq 0$ [22].

2.1.3 Dual Quaternions

Dual quaternion are dual numbers in which the primary and dual parts are quaternions. They are the building blocks of the kinematic control theory used in this work.

A three dimensional translation $\mathbf{t} = t_x \hat{i} + t_y \hat{j} + t_z \hat{k}$ can be combined with a rotation \mathbf{r} to represent a unit dual quaternion rigid body motion as $\underline{\mathbf{h}} = \mathbf{r} + \frac{1}{2} \varepsilon \mathbf{t} \mathbf{r}$ [22, ch. 2]. Furthermore, the unit dual quaternion conjugate is $\underline{\mathbf{h}}^* \triangleq \mathbf{r}^* + \frac{1}{2} \varepsilon (\mathbf{t} \mathbf{r})^*$. Therefore, a unit dual quaternion represent rotations and translations simultaneously using 8 parameters while HTMs represent rigid body motions with the use of 12 free parameters and are also more computationally demanding.

A general dual quaternion is composed of eight elements $\underline{\mathbf{g}} = g_1 + g_2 \hat{i} + g_3 \hat{j} + g_4 \hat{k} + \varepsilon (g_5 + g_6 \hat{i} + g_7 \hat{j} + g_8 \hat{k})$. The vec operator is used to map it into an eight-dimensional column real vector; i.e., $\vec{\underline{\mathbf{g}}} \triangleq [g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8]^T$. Throughout this work, the following matrix

$$\mathbf{C}_8 \triangleq \text{diag}(1, -1, -1, -1, 1, -1, -1, -1) \quad (2.1)$$

such that

$$\text{vec} \underline{\mathbf{h}}^* = \mathbf{C}_8 \text{vec} \underline{\mathbf{h}}. \quad (2.2)$$

Finally, given dual quaternions $\underline{\mathbf{g}}_1, \underline{\mathbf{g}}_2$, the Hamilton operators $\bar{\mathbf{H}}(\cdot), \dot{\mathbf{H}}(\cdot)$ are transformation matrices satisfying the following relation [15]:

$$\text{vec}(\underline{\mathbf{g}}_1 \underline{\mathbf{g}}_2) = \dot{\mathbf{H}}(\underline{\mathbf{g}}_1) \text{vec} \underline{\mathbf{g}}_2 = \bar{\mathbf{H}}(\underline{\mathbf{g}}_2) \text{vec} \underline{\mathbf{g}}_1. \quad (2.3)$$

The set of dual quaternions $\mathbb{H} \otimes \mathbb{D}$ forms a group under dual quaternion multiplication [15]. Quaternions and dual quaternions are associative and distributive, but non-commutative. The Hamilton operators are used as a way to describe the multiplication operation of the dual quaternion algebra using matrix form which allows to commute the terms [15].

2.2 Kinematic Modeling, Control and Error Criteria

The first steps to mathematically describe the kinematic modeling of rigid body motion applied to robotic serial manipulators is to group the pose or Cartesian coordinate variables of the manipulator end effector in a vector \vec{x} of size m . Then, the position of the manipulator joints are grouped in $\vec{\theta}$ of size n , where n is the number of degrees of freedom (DOF) of the robot. One of the ways to solve the kinematic problem is to work in the joint space and find

$$\vec{\theta} = g(\vec{x}) \quad (2.4)$$

which means how the robot joints $\vec{\theta}$ must be set in order that the end effector is located at the position \vec{x} . This is also known as the inverse kinematic problem. Let the search for g begin by using a already known vector valued function

$$\vec{x} = f(\vec{\theta}). \quad (2.5)$$

In (2.5), f is the forward kinematics model (FKM). The FKM can be found for any serial link robotic manipulator using its Denavit-Hartenberg(DH) parameters [17, 16]. A solution for g would be to obtain $f^{-1}(\vec{x})$. Yet, the inversion of f is rarely straightforward. Additionally, if $\vec{\theta}$ has higher order than \vec{x} there could be infinite solutions for this problem. This is the case for robots with more than six DOF [16] because they usually can reach an arbitrary point in the three dimensional space with different poses.

The inverse kinematic problem admits closed-form solutions only for manipulators having a simple kinematic structure. For complex structures it is not possible to relate the end effector pose to different sets of joint variables, or else the manipulator is redundant. The highly nonlinear relationship between joint space variables and task space variables causes this limitations [16, ch. 3].

Another approach to solve the inverse kinematic problem is to use differential kinematics which represents a linear mapping between the joint velocity and the task space velocities, although it changes for each joint configuration. In the pioneer work [23], Whitney differentiated (2.5) to obtain

$$\dot{\vec{x}} = \frac{\partial f(\vec{\theta})}{\partial \vec{\theta}} \frac{\partial \vec{\theta}}{\partial t}, \quad (2.6)$$

in which

$$\mathbf{J}_w(\vec{\theta}) \triangleq \frac{\partial f(\vec{\theta})}{\partial \vec{\theta}} = \begin{bmatrix} \frac{\partial f_1(\vec{\theta})}{\partial \theta_1} & \dots & \frac{\partial f_1(\vec{\theta})}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\vec{\theta})}{\partial \theta_1} & \dots & \frac{\partial f_n(\vec{\theta})}{\partial \theta_n} \end{bmatrix}, \quad (2.7)$$

and then

$$\dot{\vec{x}} = \mathbf{J}_w(\vec{\theta}) \dot{\vec{\theta}} \quad (2.8)$$

which is a linear relation and \mathbf{J}_w is the analytical Jacobian. If $m = n$ and $\mathbf{J}_w(\theta)$ is full rank, the basis of kinematic control is obtained as

$$\dot{\vec{\theta}} = \mathbf{J}_w^{-1}(\vec{\theta}) \dot{\vec{x}}. \quad (2.9)$$

Equation (2.9) allows to control the robot joints configuration in order to reach the desired end effector position in the three dimensional space. This happens by feeding the output joint velocities $\dot{\vec{\theta}}$ to the robot manipulator as motor speeds, for instance, in the case of a low level controller for each manipulator joint.

In this approach, linear systems issues are present. If $m > n$, $\mathbf{J}_w^{-1}(\vec{\theta})$ is not defined and (2.9) could have infinite solutions. A manipulator is called redundant whenever this occurs. One of the ways to overcome this problem is by using the Moore-Penrose pseudoinverse, which minimizes the instantaneous norm of the joint velocities. Another problem of (2.9) is the existence of work-space singularities. A robot configuration $\vec{\theta}$ is singular if the analytical Jacobian \mathbf{J}_w is rank-deficient. The issue arises not in the singular configuration itself, but also its neighborhood. Pseudoinverse matrices have ill-conditioning near singularities, in which small task-space coordinate reference velocities may need unreachable velocities in the robot joints. However, redundancy and work-space singularities will not be the main focus in this work¹.

2.2.1 Kinematic Control

The use of (2.9) enables to find the velocity of the manipulator joints and obtain a given velocity of the end effector in world coordinates. Nonetheless, it is still needed to find the joint positions for a desired position of the tool in space. First, an open-loop scheme will be presented to check that, despite it's simplicity, it has no guarantee of convergence, and then, a closed-loop approach is devised. In [21], Marinho used the following solution proposed in [17]: given the initial tool position $\vec{x}(0)$, the final tool position \vec{x}_d and a time frame t_f , find a reference tool velocity trajectory $\dot{\vec{x}}(t)$. Therefore, in a computational system, the goal is to control the manipulator joint velocities with a sampling period T , such that t_f/T is an integer. The result of this method is to evaluate

$$\dot{\vec{\theta}} = \mathbf{J}_w^{inv}(\vec{\theta}(t))\dot{\vec{x}} \quad (2.10)$$

a number of t_f/T steps. The joints position at each time step can be found by integration

$$\int_0^{t_f} \dot{\vec{\theta}}(t) dt = \int_0^{t_f} \mathbf{J}_w^{inv}(\vec{\theta}(t))\dot{\vec{x}}(t) dt \quad (2.11)$$

$$\vec{\theta}(t) = \int_0^{t_f} \mathbf{J}_w^{inv}(\vec{\theta}(t))\dot{\vec{x}}(t) dt + \vec{\theta}(0) \quad (2.12)$$

Assuming the tool velocity was constant during the sampling period and applying Euler integration in (2.12),

$$\vec{\theta}(k) = \sum_{l=0}^{t_f/T} \mathbf{J}_w^{inv}(\vec{\theta}(l))\dot{\vec{x}}(l)T + \vec{\theta}(0) \quad (2.13)$$

If an additive perturbation vector $\vec{b}(k)$ is added containing effects of variations of the tool velocity during the sampling interval and the uncertainties and simplifications errors from the

¹Pseudoinverse matrices will be calculated using the damped least-squares pseudoinverse proposed in [24].

integration,

$$\vec{\theta}(k) = \sum_{l=0}^{t_f/T} \mathbf{J}_w^{inv}(\vec{\theta}(l)) \dot{\vec{x}}(l) T + \sum_{l=0}^{t_f/T} \vec{b}(l) + \vec{\theta}(0). \quad (2.14)$$

This perturbation has many external disturbances sources and also the change of $\mathbf{J}_w(\vec{\theta})$ as $\vec{\theta}$ changes. Thus, (2.14) will not necessarily converge and the tool may not reach \vec{x}_d . For this reason, in [21] Marinho also proposed a closed-loop algorithm. In order to keep a cleaner notation, \mathbf{J}_w will be used in place of $\mathbf{J}_w(\theta)$. Resuming to the suitable inverse \mathbf{J}_w^{inv} and setting $\dot{\vec{x}}$ as a variable velocity of the tool in the direction of the desired \vec{x}_d ,

$$\dot{\vec{\theta}} = \mathbf{J}_w^{inv}[\vec{x}_d - \vec{x}(t)] \quad (2.15)$$

$$\dot{\vec{\theta}} = \mathbf{J}_w^{inv} \mathbf{K}[\vec{x}_d - \vec{x}(t)]. \quad (2.16)$$

This will force the tool to move with variable velocity in the direction $[\vec{x}_d - \vec{x}(t)]$, going slower the closer it is to the desired position. To control the rate of convergence, an arbitrary gain \mathbf{K} was added to (2.15). It can be shown that (2.16) converges for $\mathbf{K} > 0$ [17]. In the discrete case, the closed-loop velocity and position control algorithm of the robot joints gives

$$\dot{\vec{\theta}}(k) = \mathbf{J}_w^{inv} \mathbf{K}[\vec{x}_d - \vec{x}(k)] \quad (2.17)$$

$$\vec{\theta}(k+1) = \vec{\theta}(k) + \mathbf{J}_w^{inv} T \mathbf{K}[\vec{x}_d - \vec{x}(k)] = \vec{\theta}(k) + \mathbf{J}_w^{inv} \tilde{\mathbf{K}}[\vec{x}_d - \vec{x}(k)]. \quad (2.18)$$

The gain $\tilde{\mathbf{K}} = T \mathbf{K}$ enables to see the effect of T in the stability of the control loop. For a stable value of $\tilde{\mathbf{K}}$, as T represents the amount of time the system will take to correct itself, a bigger T should be compensated by a smaller \mathbf{K} in order to keep stability. The influence of the sampling time on the controllers parameters will appear later for other controllers used in this work.

Equations (2.17) and (2.18) make use of both the Jacobian and \vec{x} for the general manipulator case in [16, 17, 25] using HTM or using dual quaternion in [15]. Thus, this closed-loop algorithm solution for the manipulator inverse kinematics is the basic standard for kinematic control.

2.2.2 Dual Quaternion Kinematic Control

Equations (2.17) and (2.18) may be written with the dual quaternion formulation. Substituting \vec{x} in (2.10) for its dual quaternion representation,

$$\dot{\vec{\theta}} = \mathbf{J}^{inv} \text{vec}(\underline{\dot{x}}) \quad (2.19)$$

in which $\underline{\dot{x}}$ is the generalized velocity of the robot pose, i.e. the dual quaternion representation of $\dot{\vec{x}}$. In order to describe correctly the differential relation between arm coordinates and world coordinates, \mathbf{J}_w is altered to $\mathbf{J} \in \mathbb{R}^{8 \times n}$ because of the change to the dual quaternion representation. \mathbf{J} is the analytical Jacobian and can be found algebraically [15]. Although $\mathbf{J} \in \mathbb{R}^{8 \times n}$, the maximum rank of the dual quaternion analytical Jacobian \mathbf{J} is 6, as the dual quaternions have eight terms but two constraints.

Applying the same procedure to (2.19) in order to obtain the unit dual quaternion formulation of (2.15) yields

$$\dot{\vec{\theta}} = \mathbf{J}^{inv} \mathbf{K} \text{vec}(\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(t)) \quad (2.20)$$

where \mathbf{K} is a positive definite matrix, $\underline{\mathbf{x}}_d$ is the desired pose, and $\underline{\mathbf{x}}(t)$ is the dual quaternion FKM of the manipulator at time t which can be obtained in [14, Section 2.7.1]. The stability of (2.20) was proven in Pham's et al work in 2010 [26]. Once again, it is seen that the dual quaternion FKM can be directly used in the control equation. If HTMs were used, it would be possible to obtain the FKM of a manipulator, but it would still be necessary to convert it to another minimal representation to obtain $[\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(t)]$. In [21], Marinho has commented on other control applications besides pose control using the dual quaternions techniques: tool translation using the translation Jacobian \mathbf{J}_p and tool orientation using the orientation Jacobian \mathbf{J}_o [15, p. 77].

It is also possible to obtain the discrete cases (2.17) and (2.18) using the unit dual quaternion formulation. The formulation with the error $[\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(k)]$ were obtained in [26].

$$\dot{\vec{\theta}}(k) = \mathbf{J}^{inv} \mathbf{K} \text{vec}[\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(k)] \quad (2.21)$$

$$\vec{\theta}(k+1) = \vec{\theta}(k) + \mathbf{J}^{inv} \tilde{\mathbf{K}} \text{vec}[\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(k)]. \quad (2.22)$$

2.2.3 Invariant Error Function

As the last mathematical tool of this chapter, it is important to highlight the invariant error criteria. Since the spatial distance in dual quaternion space is described by $\underline{\mathbf{x}}_e = \underline{\mathbf{x}}^* \underline{\mathbf{x}}_d$, also known as the multiplication by the conjugate, then the invariant error function is given by [27]

$$\underline{\mathbf{e}} = 1 - \underline{\mathbf{x}}^* \underline{\mathbf{x}}_d \quad (2.23)$$

It is seen from (2.23) that when $\underline{\mathbf{x}}$ converges to $\underline{\mathbf{x}}_d$, the spatial difference $\underline{\mathbf{x}}_e$ tends to 1 and, hence, the dual quaternion error function $\underline{\mathbf{e}} \rightarrow 0$. In other words, the manipulator end effector will reach the desired pose when $\underline{\mathbf{e}}$ is stabilized to 0. The invariant error $\underline{\mathbf{e}}$ also has convergence properties related to the adopted definition. For instance, if the invariant error with respect to coordinate changes with regard to arbitrary left shifts is considered, that is, $\underline{\mathbf{x}}_e = \underline{\mathbf{x}}^* \underline{\mathbf{x}}_d$, and that both end effector pose and desired set point have been transformed by a coordinate change represented by the unit dual quaternion $\underline{\mathbf{y}}$,

$$\begin{aligned} \underline{\mathbf{x}}' &= \underline{\mathbf{y}} \underline{\mathbf{x}} \\ \underline{\mathbf{x}}'_d &= \underline{\mathbf{y}} \underline{\mathbf{x}}_d. \end{aligned}$$

The invariant error in the new coordinate system is given by

$$\begin{aligned}
\underline{\mathbf{e}}' &= 1 - \underline{\mathbf{x}}'_e = 1 - \underline{\mathbf{x}}'^* \underline{\mathbf{x}}'_d \\
&= 1 - \underline{\mathbf{x}}^* \underline{\mathbf{y}}^* \underline{\mathbf{y}} \underline{\mathbf{x}}_d \\
&= 1 - \underline{\mathbf{x}}^* \underline{\mathbf{x}}_d \\
&= \underline{\mathbf{e}}.
\end{aligned}$$

Therefore, $\underline{\mathbf{e}}$ is invariant with respect to coordinate changes with regard to arbitrary left shifts. If $\underline{\mathbf{x}}_e = \underline{\mathbf{x}} \underline{\mathbf{x}}_d^*$, a similar result is obtained and $\underline{\mathbf{e}}$ is invariant with respect to coordinate changes with regard to arbitrary right shifts. From the control point of view, the controllers are designed to asymptotically stabilize the system with the goal of $\underline{\mathbf{x}}(t) \rightarrow \underline{\mathbf{x}}_d(t)$ as $t \rightarrow \infty$. In this sense the invariant error norm results in pose convergence to the desired pose without regard to the robot base coordinate systems and from coordinate changes [27].

This error form will be used in the unit dual quaternion kinematic controllers implemented in this work in Chapter 3. As the starting point, the control law (2.20) is derived using the invariant error. The distance error $\underline{\mathbf{x}}_d - \underline{\mathbf{x}}$ is changed to $\underline{\mathbf{e}}$ and, consequently, there are some more operations in addition to the inverse Jacobian matrix.

2.3 Conclusion

This chapter began with the introduction of dual quaternions. It was seen how to represent rotations using quaternions and how to describe both translations and rotations with dual quaternions. In comparison to the 12 parameters of the HTMs, unit quaternions have 4 parameters and solve the singularity problem of the rotation matrices parametrization allowing the use of minimal rotation representations. The last section of the chapter presented the kinematic control basic theory which will be explored in Chapter 3. It was also shown that the dual quaternions may be directly used in control laws without the need to convert them to another minimal representation. For all this reasons, the dual quaternions were chosen as the proper mathematical framework in this work to implement kinematic controllers in the task-space for robotic manipulators.

Chapter 3

Kinematic Controllers

This chapter presents a compilation of dual quaternion based kinematic controllers recently proposed in literature which were implemented in this work for the trajectory control of a compliant anthropomorphic robotic arm. The mathematical tools and kinematic control concepts introduced in Chapter 2 are used to derive kinematic control laws. The first one to be obtained is a proportional controller similar to the one explained in Section 2.2.2. Afterward it is extended to a proportional controller with feed-forward term. Then, dual quaternion based H_∞ robust controllers are presented. Lastly, two linear quadratic optimal controllers are devised.

Recalling some concepts presented in Chapter 2, it is known that the forward kinematics model (FKM) of a serial manipulator robot (that is, the mapping between the n-dimensional vector of joint positions $\boldsymbol{\theta} \in \mathbb{R}^n$ and the end effector pose $\underline{\mathbf{x}}$) can be obtained directly in dual quaternion space using algebraic manipulations [15]. In addition, the differential FKM (i.e., the mapping between the joint velocities $\dot{\boldsymbol{\theta}} \in \mathbb{R}^n$ and the generalized end effector velocity $\text{vec}\dot{\underline{\mathbf{x}}} \in \mathbb{R}^8$) can be obtained from (2.19) by multiplying it from the right side by \mathbf{J} which yields

$$\text{vec}\dot{\underline{\mathbf{x}}} = \mathbf{J}\dot{\boldsymbol{\theta}}, \quad (3.1)$$

where $\mathbf{J} \in \mathbb{R}^{8 \times n}$ is the manipulator Jacobian (which is also found algebraically [15, 14, Section 2.7.2-2.7.3] and depends on robot joints configuration).

3.1 Proportional Controller

Since the manipulator differential kinematics (3.1) is a simple linear mapping, it is common practice to design closed-loop controllers based on the pseudoinverse of the Jacobian \mathbf{J} . Proportional controllers can exponentially reduce the error between the current pose $\underline{\mathbf{x}}$ and a desired pose $\underline{\mathbf{x}}_d$.

In Section 2.2.2, a proportional controller was obtained in (2.21) using the $[\underline{\mathbf{x}}_d - \underline{\mathbf{x}}(k)]$ error¹.

¹Although this control law can be easily derived, it must be noted that when representing rigid motions using dual quaternions, the reverse motion operation in unit dual quaternion space is not given by a subtraction, but by the multiplication by the conjugate.

Another way of reaching the proportional controller is to apply a least square minimization to

$$\left\| \text{vec } \underline{\mathbf{e}} - \mathbf{J} \dot{\theta} \right\|^2$$

[28], and obtain the following control law

$$\dot{\theta} = \mathbf{K} \mathbf{J}^\dagger \text{vec } \underline{\mathbf{e}}, \quad (3.2)$$

where \mathbf{K} is a positive definite matrix and \mathbf{J}^\dagger is the pseudoinverse of \mathbf{J} .

Now, the proportional controller will be derived using invariant error form to obtain the result of [27]. Assuming a time varying pose trajectory $\underline{\mathbf{x}}$ and a desired pose trajectory $\underline{\mathbf{x}}_d$ with constant velocity, it is possible to differentiate the invariant error $\underline{\mathbf{e}} = 1 - \underline{\mathbf{x}}^* \underline{\mathbf{x}}_d$ and obtain

$$\dot{\underline{\mathbf{e}}} = -\dot{\underline{\mathbf{x}}}^* \underline{\mathbf{x}}_d - \underline{\mathbf{x}} \dot{\underline{\mathbf{x}}}^*_d = -\dot{\underline{\mathbf{x}}}^* \underline{\mathbf{x}}_d. \quad (3.3)$$

Then, applying the vec operator of (2.3) to $\dot{\underline{\mathbf{e}}}$, considering that $\underline{\mathbf{e}} \triangleq \text{vec}(\underline{\mathbf{e}})$ and $\dot{\underline{\mathbf{e}}} = \text{vec}(\dot{\underline{\mathbf{e}}})$, hence

$$\dot{\underline{\mathbf{e}}} = \text{vec}(-\dot{\underline{\mathbf{x}}}^* \underline{\mathbf{x}}_d) = \bar{\mathbf{H}}(\underline{\mathbf{x}}_d) \text{vec}(-\dot{\underline{\mathbf{x}}}^*). \quad (3.4)$$

From (2.1) and (2.2), it is seen that

$$\dot{\underline{\mathbf{e}}} = -\bar{\mathbf{H}}(\underline{\mathbf{x}}_d) \mathbf{C}_8 \text{vec}(\dot{\underline{\mathbf{x}}}) \quad (3.5)$$

where it is possible to use the differential FKM from (3.1) to obtain

$$\dot{\underline{\mathbf{e}}} = -\bar{\mathbf{H}}(\underline{\mathbf{x}}_d) \mathbf{C}_8 \mathbf{J} \dot{\theta}. \quad (3.6)$$

Defining

$$\mathbf{N} \triangleq \bar{\mathbf{H}}(\underline{\mathbf{x}}_d) \mathbf{C}_8 \mathbf{J}, \quad (3.7)$$

substituting it in (3.6) and solving for $\dot{\theta}$,

$$\begin{aligned} \dot{\underline{\mathbf{e}}} &= -\mathbf{N} \dot{\theta} = \text{vec}(\dot{\underline{\mathbf{e}}}) \\ \dot{\theta} &= -\mathbf{N}^\dagger \text{vec}(\dot{\underline{\mathbf{e}}}), \end{aligned}$$

where \mathbf{N}^\dagger is the pseudoinverse² matrix. Finally, an arbitrary gain k is added in order to control the rate of convergence. From control theory, for an exponential decay, it is considered $\text{vec}(\dot{\underline{\mathbf{e}}}) = -k \text{vec}(\underline{\mathbf{e}})$, $k \in \mathbb{R}^+$. Hence, the result of [27] is achieved for the dual quaternion based proportional controller (K controller) given by

$$\begin{cases} \mathbf{K} = k \mathbf{I}, \text{ where } k \in \mathbb{R}^+ \\ \dot{\theta} = \mathbf{N}^\dagger \mathbf{K} \text{vec } \underline{\mathbf{e}}. \end{cases} \quad (3.8)$$

The arbitrary gain matrix \mathbf{K} in (3.8) controls the rate of convergence. It is shown in [16] and in [27] that (3.8) converges for $\mathbf{K} > \mathbf{0}$ and that it is stable.

²As it was mentioned in chapter 2, pseudoinverse matrices are being calculated using the damped least-squares pseudoinverse proposed in [24].

3.1.1 Decoupled Proportional Controller

In addition to the classic proportional controller approach, a proportional controller which decouples rotation and translation of the end effector was also derived. Since there is no desire to control both orientation and position of the end effector simultaneously in this method and the goal is mainly to follow a desired trajectory, this would be the most simple way to control the end effector position. Comparing it with the proportional controller (3.8), the control law would be the same, but there would be modifications in the Jacobian \mathbf{J} and in the error vector \vec{e} .

First, instead of using the analytical Jacobian, \mathbf{J} is redesigned. The translation Jacobian \mathbf{J}_p is obtained from \mathbf{J} and then merged with the rotational Jacobian [15, p. 77]. A new Jacobian is obtained in which the first four rows are related with the translation and the last four rows are related to orientation.

The error criteria is also modified. The first four rows of the error vector \vec{e} are the translation error given by the difference of the translational part of $\underline{\mathbf{x}}_d$ and of $\underline{\mathbf{x}}$ derived in [15, p. 29], while the last four rows are the invariant error norm of the primary part of $\underline{\mathbf{x}}_d$ and $\underline{\mathbf{x}}$.

This controller is the least complex one in comparison to all the other controllers in this work. However, it does not consider the orientation in which the end effector follows the desired trajectory. In other words, the end effector pose may reach the desired end effector pose with the wrong orientation, hence, the orientation will be corrected which may incur in pose error then the translational and rotational error may not converge smoothly and simultaneously to zero. Therefore, some issues may arise when this controller is used in practical applications.

One of them is that the orientation of the end effector tool may be wrong in order to execute the desired task. For example, consider that a robot must weld a electronic component on a computer motherboard. The end effector should approach the motherboard from a specific side to reach the desired position and start welding. It can not approach it from the other side, even though the end effector would reach the same position.

The robot arm configuration will not be restricted during the trajectory, only its end effector position that must follow the desired trajectory. Therefore the controller may generate robot configurations $\vec{\theta}$ along the trajectory which will bring the robot to a singular pose. Hence, it is possible to occur damages to the robot due to robot joint limitations. Furthermore, this work uses the Meka robot which is a anthropomorphic robot manipulator with joint limitations similar to the human arm.

An experimental evaluation for this controller was done in [29]. It can be seen that the decoupled controller does not succeed in following a desired trajectory with a desired orientation. Moreover, it has higher end effector velocities and accelerations peaks along the trajectory and as a consequence the joints control effort is also higher. For all the reasons mentioned in this subsection, the decoupled proportional controller will not be explored in details in this work.

3.2 Proportional Controller with Feed-forward Term

The proportional controller yields an easy to implement and efficient solution for setpoint control. Still, it neglects the influence of a moving reference and the pose variation along the trajectory. To better address the context of a end effector tracking a desired trajectory instead of a fixed point, it is added a feed-forward term (K+FF) to the proportional controller (3.8).

Considering a time-varying reference $\underline{\mathbf{x}}_d = \underline{\mathbf{x}}_d(t)$, then the derivative of (2.23) is given by

$$\dot{\underline{\mathbf{e}}} = -\dot{\underline{\mathbf{x}}}^* \underline{\mathbf{x}}_d - \underline{\mathbf{x}}^* \dot{\underline{\mathbf{x}}}_d. \quad (3.9)$$

Applying the *vec* operator to (3.9), it is obtained

$$\dot{\underline{\mathbf{e}}} = \text{vec}(-\dot{\underline{\mathbf{x}}}^* \underline{\mathbf{x}}_d) - \text{vec}(\underline{\mathbf{x}}^* \dot{\underline{\mathbf{x}}}_d). \quad (3.10)$$

Using (2.2), (2.3), (3.1) and (3.7) in (3.10) yields

$$\begin{aligned} \dot{\underline{\mathbf{e}}} &= -\bar{\mathbf{H}}(\underline{\mathbf{x}}_d) \text{vec} \dot{\underline{\mathbf{x}}}^* - \text{vec}(\underline{\mathbf{x}}^* \dot{\underline{\mathbf{x}}}_d) \\ &= -\bar{\mathbf{H}}(\underline{\mathbf{x}}_d) \mathbf{C}_8 \dot{\vec{\theta}} - \text{vec}(\underline{\mathbf{x}}^* \dot{\underline{\mathbf{x}}}_d) \\ &= -\mathbf{N} \dot{\vec{\theta}} - \text{vec}(\underline{\mathbf{x}}^* \dot{\underline{\mathbf{x}}}_d). \end{aligned} \quad (3.11)$$

To obtain an exponentially stable closed loop system, it is defined the following dual quaternion based proportional controller with the feed-forward term, solving (3.11) for $\dot{\vec{\theta}}$,

$$\begin{cases} \mathbf{K} = k \mathbf{I}, \text{in which } k \in \mathbb{R}^+ \\ \dot{\vec{\theta}} = \mathbf{N}^\dagger (\mathbf{K} \underline{\mathbf{e}} - \text{vec} \underline{\mathbf{x}}^* \dot{\underline{\mathbf{x}}}_d). \end{cases} \quad (3.12)$$

Hence, $\dot{\underline{\mathbf{e}}} = k \underline{\mathbf{e}}$ where k defines the convergence rate of an exponential error decay. Since the end effector pose must follow a desired trajectory instead of a fixed point, the feedforward term works as a feedback term which compensates the pose and velocity variation along the trajectory.

3.3 Dual Quaternion Based Robust H_∞ Kinematic Control

The FKM maps the rigid body joints configuration $\vec{\theta} \in \mathbb{R}^n$ to the end-effector pose configuration $\underline{\mathbf{x}}_E$ which can be given by

$$\underline{\mathbf{x}}_E = \underline{\mathbf{x}}_1^0 \underline{\mathbf{x}}_2^1 \dots \underline{\mathbf{x}}_n^{n-1}, \quad (3.13)$$

and the differential FKM which maps the robot joints velocities $\dot{\vec{\theta}} \in \mathbb{R}^n$ to the end-effector velocity $\text{vec} \dot{\underline{\mathbf{x}}}_E \in \mathbb{R}^8$ given by $\dot{\underline{\mathbf{x}}}_E = \sum_{i=0}^{n-1} \underline{\mathbf{x}}_i^0 \dot{\underline{\mathbf{x}}}_{i+1}^i \underline{\mathbf{x}}_n^{i+1}$. It is known that $\underline{\mathbf{x}}_{i+1}^i$ is a function of θ_i , i.e. $\underline{\mathbf{x}}_{i+1}^i = \underline{f}_i(\theta_i)$, where $\underline{f}_i : \mathbb{R} \rightarrow \text{Spin}(3) \ltimes \mathbb{R}^3$, therefore $\dot{\underline{\mathbf{x}}}_{i+1}^i = \underline{f}'_i(\theta_i) = \underline{\omega}_i \dot{\theta}_i \underline{f}_i(\theta_i)$ with $\underline{\omega}_i \in \mathbb{H}_0 \otimes \mathbb{D}$. As long as the dual quaternion differential FKM is well defined within the unit dual quaternion group, the differential FKM is described in [14, p. 46] as

$$\begin{aligned} \dot{\underline{\mathbf{x}}}_E &= \frac{1}{2} \sum_{i=0}^{n-1} j_i \dot{\theta}_i \underline{\mathbf{x}}_E \\ &= \frac{1}{2} \text{vec}_6(\mathbf{J}_\omega \dot{\theta}) \underline{\mathbf{x}}_E, \end{aligned} \quad (3.14)$$

where $\underline{vec}_6(\mathbf{J}_{\omega} \dot{\theta})$ describes the end-effector twist in dual quaternion space [14, p. 37], $\vec{\theta} = [\theta_0, \dots, \theta_{n-1}]^T$ is the measured vector of joints variables and \mathbf{J}_{ω} is the Jacobian [14, p. 36-37].

The end-effector pose configuration (3.13) describes the system model in an ideal environment. However, in realistic scenarios, the system is subjected to a lot of external influences. It is desirable to take into account the effects of different disturbances over the system. The trajectory of the end-effector may be affected by exogenous disturbances, inaccurate robot arm parameters and uncertainties in the FKM due to unmodeled viscosity and friction. In order to improve the kinematics accuracy and control performance, Figueredo proposed in [14, p. 46-47] a more complete and accurate kinematics description which considers twist and pose configuration uncertainties.

In his thesis, Figueredo used the following definitions in [14, p. 46-47] for the two mentioned types of uncertainties.

Twist uncertainties v_ω take into account unmodeled twists describing multiple sources of exogenous disturbances, unmodeled time-varying uncertainties and forces acting directly at the pose of the end-effector (e.g. gravity or interaction with the environment) or at different links from the serial manipulator, such that

$$\dot{\underline{\mathbf{x}}}_E = \frac{1}{2} \sum_{i=0}^{n-1} j_i \dot{\theta}_i \underline{\mathbf{x}}_E + \frac{1}{2} v_\omega \underline{\mathbf{x}}_E. \quad (3.15)$$

The differential equation is well-posed as $\underline{v}_\omega \triangleq v_\omega + \varepsilon v'_\omega$, with $v_\omega, v'_\omega \in \mathbb{H}_0$, is in the Lie algebra of $Spin(3) \ltimes \mathbb{R}^3$.

Pose configuration uncertainties considers unforeseen inaccuracies within model parameters and time-varying uncertainties and disturbances on the coordinate base of the manipulator or at the reference frame. They may be described as an unknown transformation in the FKM as

$$\underline{\mathbf{x}} = \underline{\mathbf{x}}_E \underline{\mathbf{c}}, \quad (3.16)$$

where $\underline{\mathbf{c}} \in Spin(3) \ltimes \mathbb{R}^3$ and $\underline{\mathbf{x}}$ denotes the real configuration of the disturbed end-effector. In this sense and with regard to the kinematics of the unknown pose disturbance, $\dot{\underline{\mathbf{c}}} = \frac{1}{2} \underline{\mathbf{c}} \tilde{v}_c$, the differential kinematics yields

$$\begin{aligned} \dot{\underline{\mathbf{x}}} &= \dot{\underline{\mathbf{x}}}_E \underline{\mathbf{c}} + \underline{\mathbf{x}}_E \dot{\underline{\mathbf{c}}} \\ &= \frac{1}{2} \sum_{i=0}^{n-1} j_i \dot{\theta}_i \underline{\mathbf{x}}_E \underline{\mathbf{c}} + \frac{1}{2} v_\omega \underline{\mathbf{x}}_E \underline{\mathbf{c}} + \frac{1}{2} \underline{\mathbf{x}}_E \underline{\mathbf{c}} \tilde{v}_c \\ &= \frac{1}{2} \left(\sum_{i=0}^{n-1} j_i \dot{\theta}_i \underline{\mathbf{x}} + v_\omega \underline{\mathbf{x}} + \underline{\mathbf{x}} \tilde{v}_c \right) \end{aligned} \quad (3.17)$$

where $\tilde{v}_c \in \mathbb{H}_0 \otimes \mathbb{D}$. To improve the readability, \tilde{v}_c is redefined with the adjoint transformation, $\tilde{v}_c \triangleq \underline{\mathbf{x}}^* v_c \underline{\mathbf{x}}$ which does not affect the induced norm, $\| \underline{v}_c \| = \| \tilde{v}_c \|$, such that,

$$\dot{\underline{\mathbf{x}}} = \frac{1}{2} \left(\sum_{i=0}^{n-1} j_i \dot{\theta}_i \underline{\mathbf{x}} + v_\omega \underline{\mathbf{x}} + v_c \underline{\mathbf{x}} \right).$$

The goal of the robust H_∞ kinematic controller is to bring the manipulator to a desired configuration pose reducing the impact of uncertainties and disturbances and keeping stability. Therefore the noise-to-output influence should reduce while the system remains internally stable.

In Chapter 2, the spatial difference in the dual quaternion space invariant to left shifts was defined as $\underline{\mathbf{x}}_e = \underline{\mathbf{x}} \underline{\mathbf{x}}_d^*$. For the H_∞ kinematic controllers, the invariant to right shifts error is used

$$\underline{\mathbf{x}}_e = \underline{\mathbf{x}} \underline{\mathbf{x}}_d^* = \mathbf{r}_e + \varepsilon \frac{1}{2} \mathbf{p}_e \mathbf{r}_e, \quad (3.18)$$

where \mathbf{r}_e denote the orientation error in $Spin(3)$ and \mathbf{p}_e the position error in \mathbb{H}_0 . Hence, the invariant dual quaternion error function (2.23) using (3.18) will be

$$\underline{\mathbf{e}} = 1 - \underline{\mathbf{x}}_e = 1 - \underline{\mathbf{x}}\underline{\mathbf{x}}_d^* = \mathbf{e} + \varepsilon \mathbf{e}' . \quad (3.19)$$

Therefore, the spatial differential kinematics for $\underline{\mathbf{x}}_e$ results in

$$\begin{aligned} \dot{\underline{\mathbf{x}}}_e &= \frac{1}{2} \left(\sum_{i=0}^{n-1} j_i \dot{\theta}_i \underline{\mathbf{x}} + \underline{v}_\omega \underline{\mathbf{x}} + \underline{v}_c \underline{\mathbf{x}} \right) \underline{\mathbf{x}}_d^* \\ &= \frac{1}{2} \left(\sum_{i=0}^{n-1} j_i \dot{\theta}_i + \underline{v}_\omega + \underline{v}_c \right) \underline{\mathbf{x}}\underline{\mathbf{x}}_d^* \\ &= \frac{1}{2} \left(\sum_{i=0}^{n-1} j_i \dot{\theta}_i + \underline{v}_\omega + \underline{v}_c \right) \underline{\mathbf{x}}_e . \end{aligned} \quad (3.20)$$

The uncertainties and disturbances considered in the H_∞ sense assume that

$$v_\omega, v'_\omega, v_c, v'_c \in L_2[0, \infty), \quad (3.21)$$

where L_2 is the Hilbert space of all square-integrable functions. The induced norm of the map $v \rightarrow z$, where $z \in L_2[0, \infty)$ is the desired output state [30], results in the gain that defines the H_∞ norm and represents the supremum of the noise amplification upon the system output

$$\sup \left\{ \frac{\|z\|_2}{\|v\|_2}, v \in L_2 \setminus \{0\} \right\}, \quad (3.22)$$

which means the worst-case influence of the noise over the controlled output. In other words, if, for instance, we consider z as the integral norm of the invariant error along the trajectory, the amount of uncertainty v to which the system is subjected to will limit the error. The H_∞ norm does not require inferences about the statistics of the uncertainties and noises. This is an advantage when working in the space of rigid body transformations because probability density functions are in general not well defined for non-Euclidean spaces [14, p. 48].

To solve the problem of minimization of (3.22), Figueredo [14] introduced the variable γ which upper bounds the induced norm

$$\|z\|_2 \leq \gamma \|v\|_2 . \quad (3.23)$$

Therefore, the smaller the value of γ , the smaller the influence of the uncertainties v over z (i.e. the integral norm of the error). In this sense, γ allow to reduce the effect of the disturbances v on the output z . The aim of the H_∞ control is to reduce the noise-to-output upper bound γ while keeping the system internal stability.

3.3.1 H_∞ Control in Dual Quaternion Space

Figueredo proposed in [14, ch. 3] a setpoint controller and a trajectory tracking controller for robot arms. Both controllers were designed to ensure H_∞ performance to the system. Therefore, they take into account (3.21), (3.22) and (3.23) which considers the worst-case influence of the

noise over the controlled output (i.e. robot joint velocities) and the noise-to-output upper bound γ .

In robotic manipulators the actuation is done in the joint-space. Nonetheless, it is more natural for humans to devise jobs for robotic arms in the task-space. Hence, the H_∞ controller also has another interesting advantage. Due to dual quaternion algebra properties, the control laws were derived in the task-space and exploited to the joint velocity space. Therefore, the manipulator tasks may be defined in the task-space. Moreover, there is a intuitive connection between control effort in task-space, the performance effects over the end-effector trajectory and the influence of uncertainties and disturbance which eases the controller parameter selection.

The invariant error considers coupled orientation and translation in (3.19). Hence, the orientation and translational errors are defined respectively as

$$O(\underline{\mathbf{e}}) \triangleq 1 - \mathbf{r}_e, \quad (3.24)$$

$$T(\underline{\mathbf{e}}) \triangleq \mathbf{p}_e. \quad (3.25)$$

The orientation error is obtained from the primary quaternion part of (3.19), that is, $O(\underline{\mathbf{e}}) = e$ and the translational error is $T(\underline{\mathbf{e}}) = -2e'(1 - e^*)$ [14, ch.3].

To describe the robust performance in the H_∞ sense, in terms of the dual quaternion error (3.19) and the translational and orientation error, Figueredo used the following definition based on [30] .

Definition 3.1 (Definition 3.3 in [14, p. 62]) For prescribed positive scalars $\gamma_{O_1}, \gamma_{O_2}, \gamma_{T_1}, \gamma_{T_2}$, the robust control performance is achieved in the H_∞ sense if the following hold

1. The error (3.19) is exponentially stable for $\underline{v}_\omega \equiv \underline{v}_c \equiv 0$;
2. Under the assumption of zero initial conditions, the disturbances influence upon the orientation and translational errors is attenuated below a desired level

$$\| O(\underline{\mathbf{e}}) \|_2 \leq \gamma_{O_1} \| v_\omega \|_2 + \gamma_{O_2} \| v_c \|_2 \quad \forall v_\omega, v_c \in L_2[0, \infty) \setminus 0; \quad (3.26)$$

$$\| T(\underline{\mathbf{e}}) \|_2 \leq \gamma_{T_1} \| v_\omega \|_2 + \gamma_{T_2} \| v_c \|_2 \quad \forall v_\omega, v_c \in L_2[0, \infty) \setminus 0. \quad (3.27)$$

3.3.2 Dual Quaternion H_∞ Controllers

Figueredo designed the controllers ensuring H_∞ performance for control and tracking problems without decoupling the rotational and translational errors. To solve the problem that traditional H_∞ theory does not deal with multiplicative noises as in (3.20), the differential forward kinematics

(3.20) was rewritten as

$$\underline{\mathbf{x}}_e = \frac{1}{2} \left(\sum_{i=0}^{n-1} j_i \dot{\theta}_i + \underline{v}_\omega + \underline{v}_c \right) \underline{\mathbf{x}}_e \quad (3.28)$$

$$= \frac{1}{2} (\underline{\text{vec}}(\mathbf{J}_\omega \dot{\underline{\theta}}) + \underline{v}_\omega + \underline{v}_c) \underline{\mathbf{x}}_e, \quad (3.29)$$

where $\vec{\theta} = [\theta_0 \dots \theta_{n-1}]^T$ is the measured vector of joint variables and $\mathbf{J}_\omega = [\underline{\text{vec}} j_0 \dots \underline{\text{vec}} j_{n-1}]$ is the analytical Jacobian [14, p. 36-37].

The robust exponential stabilization of the dual quaternion error function (3.19) led to the task-space controller that yields the following joint velocity inputs

$$\dot{\vec{\theta}} = \mathbf{J}_\omega^+ [\mathbf{K}_O \underline{\text{vec}}_3^T(Im(O(\underline{\mathbf{e}})) - \mathbf{K}_T \underline{\text{vec}}_3^T(T(\underline{\mathbf{e}}))]^T, \quad (3.30)$$

where $\underline{\mathbf{e}} \in \mathbb{H} \otimes \mathbb{D}$ is defined in (3.19), $\mathbf{K}_O, \mathbf{K}_T \in \mathbb{R}^{3 \times 3}$ are respectively the orientation and translation gain matrices, \mathbf{J}_ω^+ is the pseudoinverse of \mathbf{J}_ω , and the transformation $\underline{\text{vec}}_3 : \mathbb{H}_0 \rightarrow \mathbb{R}^3$ uses the isomorphism from \mathbb{H}_0 and \mathbb{R}^3 to map elements from $Spin(3)$ to the Euclidean vector space. The closed loop system stability was verified in [14, p. 63-64]. In addition to the task-space controller in (3.30), Figueredo also defined the following theorem³.

Theorem 3.1 (Theorem 3.7 in [14, p. 63]) For prescribed positive scalars $\gamma_{O_1}, \gamma_{O_2}, \gamma_{T_1}, \gamma_{T_2}$, the closed-loop system (3.29) with task-space controller that yields joint velocity inputs as defined in (3.30) with

$$\begin{aligned} \mathbf{K}_O &\geq \sqrt{2} \sqrt{\gamma_{O_1}^{-2} + \gamma_{O_2}^{-2}} \mathbf{I} \text{ and} \\ \mathbf{K}_T &\geq \sqrt{\gamma_{T_1}^{-2} + \gamma_{T_2}^{-2}} \mathbf{I} \end{aligned}$$

achieves exponential stability with H_∞ disturbance rejection in the sense of Definition 4.1 with minimum control effort.

In order to use kinematic controllers in practical applications, the proportional controller with feed-forward term was derived in equation (3.12). Similarly, the task-space controller (3.30) is an efficient solution for setpoint control, but in the case of having time-varying reference trajectories, it is interesting to consider a more general scenario. Figueredo solved this problem in [14, ch. 3] considering the desired end-effector trajectory over time is described by the first order differential kinematic equation of a rigid body,

$$\dot{\underline{\mathbf{x}}}_d = \frac{1}{2} \underline{\omega}_d \underline{\mathbf{x}}_d, \quad (3.31)$$

where $\underline{\omega}_d$ is the twist for the desired pose in inertial frame. Therefore, for varying pose configura-

³Avoidance techniques for kinematics singularities were not the main goal of this work. The Jacobian \mathbf{J}_ω is assumed to be well-posed. Pseudoinverse matrices were calculated using the damped least-squares pseudoinverse proposed in [24].

tions, the error differential kinematics from (3.20) can be rewritten as

$$\begin{aligned}
\dot{\underline{\mathbf{x}}}_e &= \dot{\underline{\mathbf{x}}}\underline{\mathbf{x}}_d^* + \underline{\mathbf{x}}\dot{\underline{\mathbf{x}}}_d^* \\
&= \frac{1}{2} (\underline{\text{vec}}_6(\mathbf{J}_\omega \dot{\theta}) \underline{\mathbf{x}} + \underline{v}_\omega \underline{\mathbf{x}} + \underline{v}_c \underline{\mathbf{x}}) \underline{\mathbf{x}}_d^* + \frac{1}{2} \underline{\mathbf{x}} \underline{\mathbf{x}}_d^* \underline{\omega}_d^* \\
&= \frac{1}{2} (\underline{\text{vec}}_6(\mathbf{J}_\omega \dot{\theta}) + \underline{v}_\omega + \underline{v}_c) \underline{\mathbf{x}} \underline{\mathbf{x}}_d^* + \frac{1}{2} \underline{\mathbf{x}} \underline{\mathbf{x}}_d^* \underline{\omega}_d^* \\
&= \frac{1}{2} (\underline{\text{vec}}_6(\mathbf{J}_\omega \dot{\theta}) + \underline{v}_\omega + \underline{v}_c) \underline{\mathbf{x}}_e - \frac{1}{2} \underline{\mathbf{x}}_e \underline{\omega}_d. \tag{3.32}
\end{aligned}$$

Hence, from Definition 3.1 and Theorem 3.1, the following H_∞ tracking control solution was obtained and its stability proved in [14, p.65].

Theorem 3.2 (Theorem 3.8 in [14, p. 65]) For prescribed positive scalars γ_{O_1} , γ_{O_2} , γ_{T_1} , γ_{T_2} , the task-space controller yielding joint velocity inputs

$$\dot{\theta} = \mathbf{J}_\omega^+ [\mathbf{K}_O \underline{\text{vec}}_3^T(Im(O(\underline{\mathbf{e}})) - \mathbf{K}_T \underline{\text{vec}}_3^T(T(\underline{\mathbf{e}}))]^T + \mathbf{J}_\omega^+ (\underline{\text{vec}}_6(\underline{\mathbf{x}}_e \underline{\omega}_d \underline{\mathbf{x}}_e^*)) \quad (3.33)$$

with the feed-forward term $\underline{\text{vec}}_6(\underline{\mathbf{x}}_e \underline{\omega}_d \underline{\mathbf{x}}_e^*)$ based on the desired end-effector twist $\underline{\omega}_d$ described in (3.31), and control gains

$$\begin{aligned}
\mathbf{K}_O &\geq \sqrt{2} \sqrt{\gamma_{O_1}^{-2} + \gamma_{O_2}^{-2}} \mathbf{I} \text{ and} \\
\mathbf{K}_T &\geq \sqrt{\gamma_{T_1}^{-2} + \gamma_{T_2}^{-2}} \mathbf{I}
\end{aligned}$$

ensures exponential H_∞ tracking performance with disturbance rejection in the sense of Definition 3.3.1 with minimum control effort for the closed-loop system (3.19) and (3.32).

In situations where it is difficult to decouple both sources of uncertainties the induced norm upper bound variable γ may be set the same, that is, $\gamma_T = \gamma_{T_1} = \gamma_{T_2}$ and $\gamma_O = \gamma_{O_1} = \gamma_{O_2}$.

3.4 Linear Quadratic Optimal Control (LQR)

In [26], the stable proportional gain controller was derived, but it showed an intrinsic delay in the presence of time-varying trajectories. A similar controller was also obtained in [27] and presented in this work in (3.8). To reduce the delay, a feed-forward term was added to the controller as in (3.12). Both solutions exceeds in stabilizing the trajectory of the rigid body. Nonetheless, they do not take into account the control effort which inevitably grows along the trajectory time derivative. In this sense, they may lead to higher velocity and acceleration peaks in the manipulator end-effector trajectory and as a consequence higher joint velocities. For these reasons, Figueiredo, Marinho and Adorno proposed in [18] a dual quaternion error mapping to an \mathbb{R}^8 manifold solution to optimize the task-space control trajectory and velocity. Later on, with the goal of increasing the geometric significance of the controller and ease the parameter selection, Figueiredo derived in

[14, ch. 4] an optimal criterion to optimize the dual quaternion error exponential convergence rate in the dual quaternion space.

3.4.1 Optimal Dual Quaternion Based Controller in \mathbb{R}^8 Manifold

In [18], the optimal state-feedback is derived and its computation is discussed. It is shown that the kinematic control with a time-varying reference can be described as a linear time-varying system with an additive perturbation term. Instead of considering $\dot{\vec{\theta}}$ the input signal for the system, it can be considered as input the end effector velocity u using the mapping $\vec{u} = -\mathbf{N}\dot{\vec{\theta}}$ where $\mathbf{N} \triangleq \bar{\mathbf{H}}(\underline{\mathbf{x}}_d)\mathbf{C}_8\mathbf{J}$ as in (3.7). This allows the optimization to be done in task-space variables. Consequently, there are the advantages of not requiring an external inverse kinematics solution and of making direct use of the available robot low level controller instead of redesigning them.

The LQR exploits future knowledge of the desired trajectory, therefore, it calculates the control signals offline for a given desired trajectory. With time varying $\underline{\mathbf{x}}$ and $\underline{\mathbf{x}}_d$, the error derivative is given by (3.9). Hence,

$$\begin{aligned}\underline{\mathbf{e}} &= 1 - \underline{\mathbf{x}}^* \underline{\mathbf{x}}_d \\ \implies \underline{\mathbf{e}} \underline{\mathbf{x}}_d^* &= \underline{\mathbf{x}}_d^* - \underline{\mathbf{x}}^* \\ \implies \underline{\mathbf{x}}^* &= \underline{\mathbf{x}}_d^* - \underline{\mathbf{e}} \underline{\mathbf{x}}_d^*. \end{aligned}\tag{3.34}$$

Using the vec operator on both sides of (3.9) and applying (3.34) to (3.9) yields

$$\dot{\underline{\mathbf{e}}} = -\bar{\mathbf{H}}(\underline{\mathbf{x}}_d)\mathbf{C}_8 vec \dot{\underline{\mathbf{x}}} + \bar{\mathbf{H}}(\underline{\mathbf{x}}_d^* \dot{\underline{\mathbf{x}}}_d) \underline{\mathbf{e}} - vec \underline{\mathbf{x}}_d^* \dot{\underline{\mathbf{x}}}. \tag{3.35}$$

By defining $\mathbf{A} \triangleq \bar{\mathbf{H}}(\underline{\mathbf{x}}_d^* \dot{\underline{\mathbf{x}}}_d)$ and $\vec{c} \triangleq -vec \underline{\mathbf{x}}_d^* \dot{\underline{\mathbf{x}}}$, using (3.1) and (3.7), it follows that

$$\dot{\vec{e}} = -\bar{\mathbf{H}}(\underline{\mathbf{x}}_d)\mathbf{C}_8\mathbf{J}\dot{\vec{\theta}} + \mathbf{A}\vec{e} + \vec{c} \tag{3.36}$$

$$\dot{\vec{e}} = \mathbf{A}\vec{e} - \mathbf{N}\dot{\vec{\theta}} + \vec{c}. \tag{3.37}$$

Then, the goal is to find the optimal controller for the affine time-varying system

$$\dot{\vec{e}}(t) = \mathbf{A}(t)\vec{e}(t) + \vec{u}(t) + \vec{c}(t) \tag{3.38}$$

where $\vec{u}(t) = -\mathbf{N}\dot{\vec{\theta}}(t)$.

Therefore, from the error point-of-view, it is possible to solve the tracking problem for a continuous trajectory using a finite horizon LQR applied to a disturbed system, as the error disturbance caused by the time-varying trajectory is given by $\vec{c}(t)$. Other modeled continuous disturbances can also be grouped into $\vec{c}(t)$ and used in the same solution.

Consider that the manipulator has to track the trajectory during $t \in [0, t_f]$. Then, it is wanted to minimize the following cost function

$$F = \frac{1}{2} \vec{e}(t_f)^T \mathbf{S} \vec{e}(t_f) + \frac{1}{2} \int_0^{t_f} (\vec{e}^T \mathbf{Q} \vec{e} + \vec{u}^T \mathbf{R} \vec{u}) dt, \tag{3.39}$$

given the matrices $\mathbf{S}, \mathbf{Q}(t) \geq 0$ and $\mathbf{R}(t) > 0$ with $\mathbf{S}, \mathbf{Q}, \mathbf{R} \in \mathbb{R}^{8 \times 8}$. The matrix \mathbf{S} is the weight of the final error norm, the time-varying matrix \mathbf{Q} weighs the error cost along the trajectory, and the time-varying matrix \mathbf{R} weighs the control effort in terms of end effector velocity norm. As long as \mathbf{N} is well conditioned, an increase in \mathbf{R} will also cause an overall decrease in joint velocities. The optimization of (3.39) leads to an optimal feedback without excessive expenditure of control energy while keeping the error $\vec{e}(t)$ near zero [31].

To solve the optimization problem, it is introduced the costate variable p , which acts as a Lagrange multiplier for the state equations. Then, using the equality constraint defined in (3.37), the function (3.39) can be rewritten as

$$H = F + \int_0^{t_f} p^T (\mathbf{A}\vec{e} + \vec{u} + \vec{c} - \dot{\vec{e}}) dt. \quad (3.40)$$

Considering [32], distribution theory applied to optimality conditions, and $\partial H/\partial \vec{u} = 0$ and $\partial H/\partial \vec{e} = 0$ as necessary conditions for the optimal trajectory, then,

$$\begin{aligned} \partial H/\partial \vec{u} = 0 &\implies \mathbf{R}\vec{u} + \vec{p} = 0 \implies \vec{u} = -\mathbf{R}^{-1}\vec{p} \\ \partial H/\partial \vec{e} = 0 &\implies \mathbf{Q}\vec{e} + \mathbf{A}^T\vec{p} + \vec{p} = 0 \\ &\implies \dot{\vec{p}} = -(\mathbf{Q}\vec{e} + \mathbf{A}^T\vec{p}) \end{aligned} \quad (3.41)$$

The term $\partial^2 H/\partial^2 \vec{u}$ must be positive to minimize (3.40), which requires $\mathbf{R} > 0$. The system and proposed cost function allow the use of the costate function [32]

$$\vec{p}(t) = \mathbf{P}\vec{e} + \vec{\xi}, \quad (3.42)$$

where \mathbf{P} is a time-varying proportional gain and $\vec{\xi}$ is a weighted feed-forward term. The derivative of (3.42) is given by

$$\dot{\vec{p}} = \dot{\mathbf{P}}\vec{e} + \mathbf{P}\dot{\vec{e}} + \dot{\vec{\xi}} \quad (3.43)$$

Applying (3.42) in (3.41) and using the result in (3.37), and also substituting (3.42) in (3.41), the results are applied to (3.43). It is also considered that (3.43) must hold for any choice of initial state \vec{e} and that both \mathbf{P} and $\vec{\xi}$ do not depend on the initial error, it is needed then simultaneously

$$\begin{cases} \dot{\mathbf{P}} = -\mathbf{PA} - \mathbf{A}^T\mathbf{P} + \mathbf{PR}^{-1}\mathbf{P} - \mathbf{Q} \\ \dot{\vec{\xi}} = -\mathbf{A}^T\vec{\xi} + \mathbf{PR}^{-1}\vec{\xi} - \mathbf{P}\vec{c}. \end{cases} \quad (3.44)$$

where $\vec{u}(t) = -\mathbf{R}^{-1}(\mathbf{P}\vec{e} + \vec{\xi})$.

The system (3.44) is solved by finding the boundary conditions using the final time, t_f , of the trajectory. Set $\vec{\xi}(t_f) = 0$ to find the first boundary condition. From (3.42), with $\vec{\xi}(t_f) = 0$, it is obtained $\partial H/\partial \vec{e}(t_f) = 0$ which yields $\mathbf{P}(t_f) = \mathbf{S}$. It is important to note that $\mathbf{A} \triangleq \bar{\mathbf{H}}(\underline{\mathbf{x}}_d^* \dot{\underline{\mathbf{x}}}_d)$ for all t . Hence, the differential Riccati equation $\mathbf{P}(t)$ can be numerically solved backwards in time. As $\vec{c} \triangleq -\text{vec } \underline{\mathbf{x}}_d^* \dot{\underline{\mathbf{x}}}_d$ is also known for all t and, with the solution of $\mathbf{P}(t)$, $\vec{\xi}(t)$ can be found by also solving it backwards in time. Therefore, from (3.41) and (3.42) the optimal control is given by

$\vec{u}(t) = -\mathbf{R}^{-1}(\mathbf{P}\vec{e} + \vec{\xi})$. Applying as joint velocities it is obtained the dual quaternion LQR based controller in \mathbb{R}^8 manifold [18]

$$\dot{\vec{\theta}} = \mathbf{N}^\dagger \mathbf{R}^{-1}(\mathbf{P}\vec{e} + \vec{\xi}), \quad (3.45)$$

3.4.2 Optimal Quadratic Controller in Dual Quaternion Space

Motivated by the results obtained in [18], Figueredo derived an optimal control law in the dual quaternion space. The controller has the advantage of having a explicitly geometric significance and a well-defined map between the range space of the dual quaternion error dynamics to the space where the actuation takes place. Furthermore, it yields a more intuitive understanding on the performance influence over the end-effector trajectory and eases the implementation process. On the other hand, there are highly and complex nonlinearities within the differential coupled kinematics in the end-effector dual quaternion space.

For this controller, as in the case of the H_∞ robust controllers, the following spatial difference in the dual quaternion space was adopted

$$\underline{\mathbf{x}}_e = \underline{\mathbf{x}}\underline{\mathbf{x}}_d^* = \mathbf{r}_e + \varepsilon \frac{1}{2} \mathbf{p}_e \mathbf{r}_e, \quad (3.46)$$

with a time-varying reference trajectory, that is, a tracking system, the spatial difference kinematics is given by

$$\dot{\underline{\mathbf{x}}}_e = \dot{\underline{\mathbf{x}}}\dot{\underline{\mathbf{x}}}_d^* + \underline{\mathbf{x}}\dot{\underline{\mathbf{x}}}_d^* \quad (3.47)$$

where the end-effector differential kinematics is defined in (3.14) and the desired end-effector trajectory over time is given by the first order kinematic equation of a rigid body in equation (3.31) with $\underline{\omega}_d$ being the twist of the desired pose in inertial frame. Applying (3.14) and (3.31) to the spatial difference kinematics in (3.47) results in

$$\dot{\underline{\mathbf{x}}}_e = \frac{1}{2} \underline{\text{vec}}_6(\mathbf{J}_\omega \dot{\vec{\theta}}) \underline{\mathbf{x}}_e - \frac{1}{2} \underline{\mathbf{x}}_e \underline{\omega}_d, \quad (3.48)$$

where $\vec{\theta} = [\theta_0 \dots \theta_{n-1}]^T$ is the measured vector of joint variables and $\mathbf{J}_\omega = [\underline{\text{vec}} j_0 \dots \underline{\text{vec}} j_{n-1}]$ is the analytical Jacobian [14, p. 36-37].

Although solutions are possible [33, Prop. 2.1], the spatial difference kinematics equation shows the complexity and nonlinearities involved in finding solutions in the real space for $\dot{\vec{\theta}}$ such that (3.48) remains well-posed.

In order to solve the optimal problem, the following dual quaternion based task-space controller with joint velocity vector inputs was assumed in [14, p. 83]

$$\dot{\vec{\theta}} = \mathbf{J}_\omega^+ [\mathbf{u}_c + \underline{\text{vec}}_6(\underline{\mathbf{x}}_e \underline{\omega}_d \underline{\mathbf{x}}_e^*)], \quad (3.49)$$

where

$$\mathbf{u}_c = [\mathbf{K}_O \underline{\text{vec}}_3^T(Im(\mathbf{r}_e)) \quad \frac{1}{2} \mathbf{K}_T \underline{\text{vec}}_3^T(\mathbf{p}_e)]^T, \quad (3.50)$$

and \mathbf{J}_ω^+ is the pseudoinverse of \mathbf{J}_ω , \mathbf{K}_O and \mathbf{K}_T are real valued 3×3 matrices and the transformation $\underline{\text{vec}}_3 : \mathbb{H}_0 \rightarrow \mathbb{R}^3$ defines the isomorphism mapping from \mathbb{H}_0 and \mathbb{R}^3 . If \mathbf{J}_ω is well defined, the solution

is always in the range space of \mathbf{J}_ω . From [14, p. 36], it is possible to define a dual quaternion Jacobian that maps changes in the n -joint configuration to the end-effector twist $\underline{\omega}_E$, that is, a mapping of the form $\mathbb{R}^n \mapsto \mathbb{H}_0 \otimes \mathbb{D}$ from the n -joint configuration velocities to the end-effector twist

$$\underline{\omega}_E = \mathbf{J}_\omega \dot{\vec{\theta}}, \quad (3.51)$$

hence,

$$\dot{\vec{\theta}} = \mathbf{J}_\omega^+ \underline{\omega}_E \quad (3.52)$$

and comparing (3.49) with (3.52), it is seen that

$$\underline{\omega}_E = [\mathbf{u}_c + \text{vec}_6(\underline{\mathbf{x}}_e \underline{\omega}_d \underline{\mathbf{x}}_e^*)]. \quad (3.53)$$

The spatial difference in the dual quaternion space was defined as $\underline{\mathbf{x}}_e = \underline{\mathbf{x}} \underline{\mathbf{x}}_d^*$ in (3.46) and the first order differential kinematic of a rigid body is $\dot{\underline{\mathbf{x}}} = \frac{1}{2}\underline{\omega} \underline{\mathbf{x}}$ with $\dot{\underline{\mathbf{x}}}^* = \frac{1}{2}\underline{\mathbf{x}}^* \underline{\omega}^*$. Hence, the error spatial difference kinematics (3.47) may be written as

$$\begin{aligned} \dot{\underline{\mathbf{x}}}_e &= \dot{\underline{\mathbf{x}}} \underline{\mathbf{x}}_d^* + \underline{\mathbf{x}} \dot{\underline{\mathbf{x}}}_d^* \\ &= \frac{1}{2}\underline{\omega} \underline{\mathbf{x}} \underline{\mathbf{x}}_d^* + \underline{\mathbf{x}} \frac{1}{2}\underline{\mathbf{x}}_d^* \underline{\omega}^* \\ &= \frac{1}{2}\underline{\omega} \underline{\mathbf{x}}_e + \frac{1}{2}\underline{\mathbf{x}}_e \underline{\omega}_d^* \\ &= \frac{1}{2}\underline{\omega} \underline{\mathbf{x}}_e - \frac{1}{2}\underline{\mathbf{x}}_e \underline{\omega}_d. \end{aligned} \quad (3.54)$$

Applying (3.53) in (3.54) and after some manipulation, the closed-loop spatial difference kinematics yields

$$\dot{\underline{\mathbf{x}}}_e = \frac{1}{2}\text{vec}_6(\mathbf{u}_c) \underline{\mathbf{x}}_e. \quad (3.55)$$

In order to obtain a dual quaternion exponentially stable controller, the following theorem was defined and its exponential stability was verified in [14, p.83-84].

Theorem 3.3 (Theorem 4.2 in [14, p. 83]) The closed-loop system (3.48) with dual quaternion task-space controller (3.49)-(3.50) achieves exponential stability with negative definite matrices \mathbf{K}_O and \mathbf{K}_T . Moreover the convergence decay is defined by

$$\begin{aligned} \frac{d}{dt} \| \text{vec}_3\{Im(\mathbf{r}_e)\} \|^2 &\leq \text{vec}_3^T\{Im(\mathbf{r}_e)\} \mathbf{K}_O \text{vec}_3\{Im(\mathbf{r}_e)\}, \\ \frac{d}{dt} \| \text{vec}_3\{Im(\mathbf{p}_e)\} \|^2 &\leq \text{vec}_3^T\{\mathbf{p}_e\} \mathbf{K}_T \text{vec}_3\{\mathbf{p}_e\}. \end{aligned} \quad (3.56)$$

To solve the complex nonlinearities from (3.48), the exponentially stable controller (3.49)-(3.50) and Theorem 3.3 were used to optimize the exponential gains \mathbf{K}_O and \mathbf{K}_T in the algebra of unit dual quaternions. The optimization of the exponential gains was performed at the tangent space of

the dual quaternion task-space. They are then mapped to the geodesic direction for rotation and translation yielding control actions along the geodesic directions for orientation and translational errors.

In this sense, the goal was to optimize the error dynamics and control output vector described in the tangent space as

$$\dot{\underline{\mathbf{e}}} = \underline{\mathbf{u}}, \quad (3.57)$$

where $\underline{\mathbf{u}}$ must satisfy the following constraint

$$\underline{\mathbf{u}} = \text{diag}(\mathbf{K}_O, \mathbf{K}_T)\underline{\mathbf{e}}, \quad (3.58)$$

and the goal is to minimize the cost function

$$F = \frac{1}{2}\vec{e}(t_f)^T \mathbf{S}\vec{e}(t_f) + \frac{1}{2} \int_0^{t_f} (\vec{e}^T(\tau) \mathbf{Q}\vec{e}(\tau) + \vec{u}(\tau)^T \mathbf{R}\vec{u}(\tau)) d\tau, \quad (3.59)$$

given the matrices $\mathbf{S}, \mathbf{Q}(t) \geq 0$ and $\mathbf{R}(t) > 0$ with $\mathbf{S}, \mathbf{Q}, \mathbf{R} \in \mathbb{R}^{6 \times 6}$. The matrix \mathbf{S} is the weight of the final error norm, the time-varying matrix \mathbf{Q} weighs the error cost along the trajectory, and the time-varying matrix \mathbf{R} weighs the control effort in terms of end effector velocity norm. As long as \mathbf{J}_ω is well conditioned, an increase in \mathbf{R} will also cause an overall decrease in joint velocities. The optimization of (3.59) leads to an optimal feedback in unit dual quaternion tangent space without excessive expenditure of control energy while optimizing the exponential convergence of the dual quaternion error function.

After solving the optimization problem, Figueredo obtained the following dual quaternion based task-space controller for the closed loop system (3.48)-(3.50)

$$\begin{aligned} \dot{\vec{\theta}} &= \mathbf{J}_\omega^+ [\underline{\mathbf{u}}_c + \text{vec}_6(\underline{\mathbf{x}}_e \underline{\boldsymbol{\omega}}_d \underline{\mathbf{x}}_e^*)] \\ \underline{\mathbf{u}}_c &= -\mathbf{R}^{-1} \mathbf{P}(t) \underline{\mathbf{e}} \end{aligned} \quad (3.60)$$

$$\dot{\mathbf{P}}(t) = \mathbf{P} \mathbf{R}^{-1} \mathbf{P} - \mathbf{Q}. \quad (3.61)$$

The boundary conditions yields $\mathbf{P}(t_f) = \mathbf{S}$. Therefore, the differential Riccati equation (3.61) can be solved backwards in time. Theorem 4.3 in [14, p. 86] gives the dual quaternion task-space controller gain matrices definition.

3.5 Conclusion

The third chapter of this work recalled the fundamental differential FKM equation described in Chapter 2 in order to derive the kinematic controllers. The first section derived the proportional controller using the invariant error form. However, the proportional controller is not concerned with a moving trajectory reference. Hence, in the second part of the chapter, a feed-forward term was added to the proportional controller and a new control law result was reached. Nonetheless, both controllers do not take into account uncertainties and exogenous disturbances which are present in most practical applications.

With this in mind, two dual quaternion based robust H_∞ kinematic controllers were presented. They use a more complete and accurate kinematics description yielding a controller which consider external influences in more realistic scenarios. All this controllers results are bounded to the influence of gain values which acts mainly on the trajectory error and may lead to higher end effector velocity peaks and, therefore, greater joints control effort along the trajectory. In this sense, they do not allow to adjust separately joint velocities and trajectory error.

Thus, two linear quadratic optimal controllers were presented in which it is possible to reach a balance between joint velocities (i.e. control effort) and trajectory error by setting two gain matrices \mathbf{R} and \mathbf{Q} . The main goal of this work is to implement and evaluate kinematic controllers on compliant manipulators to be used in human-robot interaction environments. Then, controllers which enable to reach a trade-off between trajectory error and end effector velocities enables to make manipulators move slower within the accepted error criteria.

Table 3.1 presents an overview of the controllers equations.

Table 3.1: Kinematic controllers equations overview.

Controller	Equation
Proportional (K)	$\begin{cases} \mathbf{K} = k\mathbf{I}, \text{ where } k \in \mathbb{R}^+ \\ \dot{\vec{\theta}} = \mathbf{N}^\dagger \mathbf{K} \underline{\text{vec}} \mathbf{e} \end{cases}$
Proportional with feedforward term (K+FF)	$\begin{cases} \mathbf{K} = k\mathbf{I}, \text{ in which } k \in \mathbb{R}^+ \\ \dot{\vec{\theta}} = \mathbf{N}^\dagger (\mathbf{K}\mathbf{e} - \underline{\text{vec}} \mathbf{x}_d^* \dot{\mathbf{x}}_d) \end{cases}$
LQR in \mathbb{R}^8	$\begin{cases} \dot{\mathbf{P}} = -\mathbf{PA} - \mathbf{A}^T \mathbf{P} + \mathbf{PR}^{-1} \mathbf{P} - \mathbf{Q} \\ \dot{\vec{\xi}} = -\mathbf{A}^T \vec{\xi} + \mathbf{PR}^{-1} \vec{\xi} - \mathbf{P} \vec{c} \\ \dot{\vec{\theta}} = \mathbf{N}^\dagger \mathbf{R}^{-1} (\mathbf{P} \vec{e} + \vec{\xi}) \end{cases}$
LQR in DQ	$\begin{cases} \dot{\vec{\theta}} = \mathbf{J}_\omega^+ [\mathbf{u}_c + \underline{\text{vec}}_6(\mathbf{x}_e \underline{\omega}_d \mathbf{x}_e^*)] \\ \mathbf{u}_c = -\mathbf{R}^{-1} \mathbf{P}(t) \mathbf{e} \\ \dot{\mathbf{P}}(t) = \mathbf{P} \mathbf{R}^{-1} \mathbf{P} - \mathbf{Q} \end{cases}$
H_∞ (HIR)	$\begin{cases} \dot{\vec{\theta}} = \mathbf{J}_\omega^+ [\mathbf{K}_O \underline{\text{vec}}_3^T (Im(O(\mathbf{e})) - \mathbf{K}_T \underline{\text{vec}}_3^T (T(\mathbf{e})))]^T \\ \mathbf{K}_O \geq \sqrt{2} \sqrt{\gamma_{O_1}^{-2} + \gamma_{O_2}^{-2}} \mathbf{I} \\ \mathbf{K}_T \geq \sqrt{\gamma_{T_1}^{-2} + \gamma_{T_2}^{-2}} \mathbf{I} \end{cases}$
H_∞ with tracking term (HIRT)	$\begin{cases} \dot{\vec{\theta}} = \mathbf{J}_\omega^+ [\mathbf{K}_O \underline{\text{vec}}_3^T (Im(O(\mathbf{e})) - \mathbf{K}_T \underline{\text{vec}}_3^T (T(\mathbf{e})))]^T + \mathbf{J}_\omega^+ (\underline{\text{vec}}_6(\mathbf{x}_e \underline{\omega}_d \mathbf{x}_e^*)) \\ \mathbf{K}_O \geq \sqrt{2} \sqrt{\gamma_{O_1}^{-2} + \gamma_{O_2}^{-2}} \mathbf{I} \\ \mathbf{K}_T \geq \sqrt{\gamma_{T_1}^{-2} + \gamma_{T_2}^{-2}} \mathbf{I} \end{cases}$

Chapter 4

Development and Implementation

The further development of techniques to control robots require not only a strong mathematical background, but also testing them in simulations and in real robots in order to check their usability in practical applications. This chapter describes the tools used, the framework developed to test the kinematic controllers and the programming libraries used to implement the dual quaternion kinematic controllers. The first section introduce the Virtual Robot Experimentation Platform (V-REP) simulator. Then, the Meka Robotics robot and some of its capabilities are presented. Afterward, the Robotics Operating System (ROS) is mentioned and the ROS computation graph of nodes and topics used in this work is shown and explained. In the next section, the DQ Robotics and Eigen libraries are introduced as programming tools to implement the mathematical operations of dual quaternion based kinematic control. Lastly, the programming framework developed in this work is explained and a brief introduction on how to start working with the robot is done.

4.1 Virtual Robot Experimentation Platform (V-REP)

Robotic systems make use of sensors, actuators and controllers. In order to become functional, they require fundamental tools and a strong foundation on kinematics, dynamics, motion planning, computer vision and control techniques [16]. Moreover, each robot or machine in a robotic application environment must often be able to deal with each one of these fields. Therefore, a general-purpose robot simulator needs to integrate them well and make possible to run everything simultaneously.

The Virtual Robot Experimentation Platform (V-REP) from Coppelia Robotics¹ is a robot simulator. V-REP main goal is to unify all the necessary computational demands for complex robotic system simulation scenarios into a versatile and scalable framework. Therefore, it uses a distributed control architecture [34].

As other robot simulation platforms, for instance OpenHRP [35], Gazebo [36] and Webots [37], V-REP offers lots of scene objects: joints, shapes, proximity sensors, vision sensors, force sensors, paths and so on. However, while many simulators offer similar functionalities, not all of them

¹<http://www.coppeliarobotics.com>

provide different options of programming techniques and their simulation models are more difficult to port between hardware or platforms.

V-REP allows the user to choose among various programming techniques simultaneously and even combine them [34]: embedded scripts in Lua, add-ons for customization or porting, plug-ins to interface to specific hardware, ROS and remote API clients via socket communication. In the case of this work, it is being used a C++ ROS node to exchange data with V-REP.

For this work, V-REP is being mainly used to test the kinematic controllers in robot manipulators before running them on the real robot. This approach to the development of controllers is beneficial to the robot. It allows to check for sudden stability problems in controllers due to numerical conditioning, numerical drifting and unexpected issues. Therefore it prevents incurring in damages to the real robot. The virtual model of a Kuka LBR iiwa 7 R800 industrial robot (Figure 4.1) was used in this work.

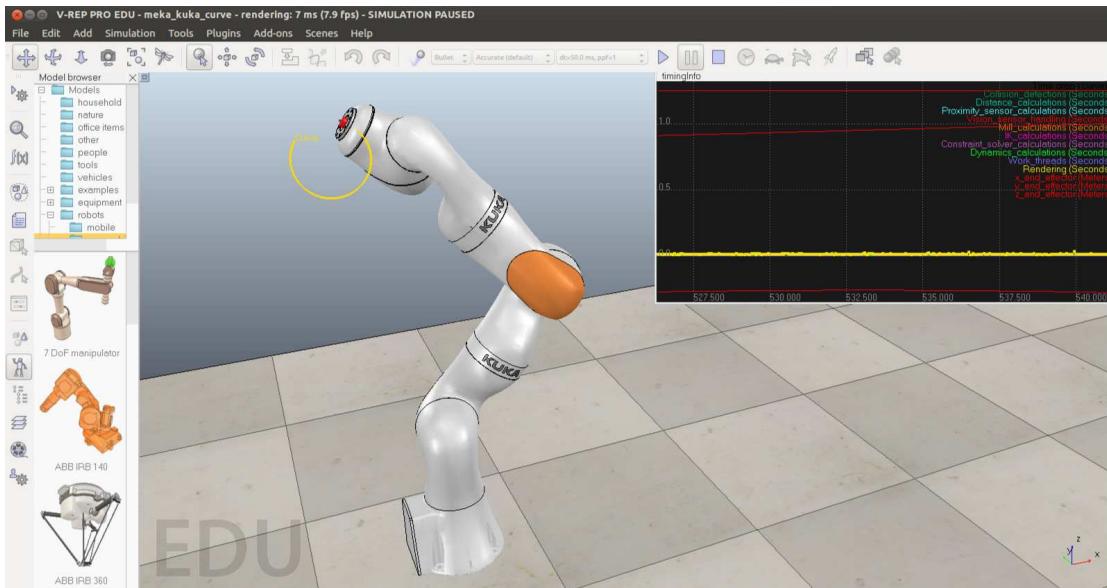


Figure 4.1: Kuka LBR iiwa 7 used for simulations in V-REP

4.2 A2 Arm from Meka Robotics

In a robotic simulation environment it is possible to test the numerical stability of the controllers for robotic manipulators. It also enables to check for sudden instabilities due to the robot dynamic model, the influence of gravity and so on. However, it cannot be fully guaranteed that the result of the control algorithm will have the same behavior in a real robot only by running simulations. Therefore, the kinematic controllers were tested in a compliant robot arm.

The compliant anthropomorphic robotic manipulator A2 Arm from Meka Robotics (acquired by Google), which is available at the Robotics and Automation Laboratory (LARA) at the University of Brasília (UnB) (Figure 4.2), was used as test platform. From now on, the robot manipulator will be referred as Meka. Recalling that the goal of this work is to implement and evaluate kinematic

controllers on compliant robots, the Meka robot suits very well the requirements.



Figure 4.2: Meka robot available at LARA at UnB (A2 Arm from Meka Robotics)

One of the main advantages of using a compliant manipulator is its safety. The Meka robot enables to easily set through software the stiffness parameter of each one of its joints which has the effect of scaling the commanded torque that is fed to the low level controllers. In other words, this means the torques generated for each joint will be adjustable. Thus, there will be no danger if someone accidentally crosses the robot end effector path, while a trajectory is being executed because joint torques are scaled.

In order to better understand the capabilities of Meka, some of its technical specifications are reviewed. All of them were obtained in the Meka User Guide Overview [38] and Controllers [39] documentation provided by Meka Robotics.

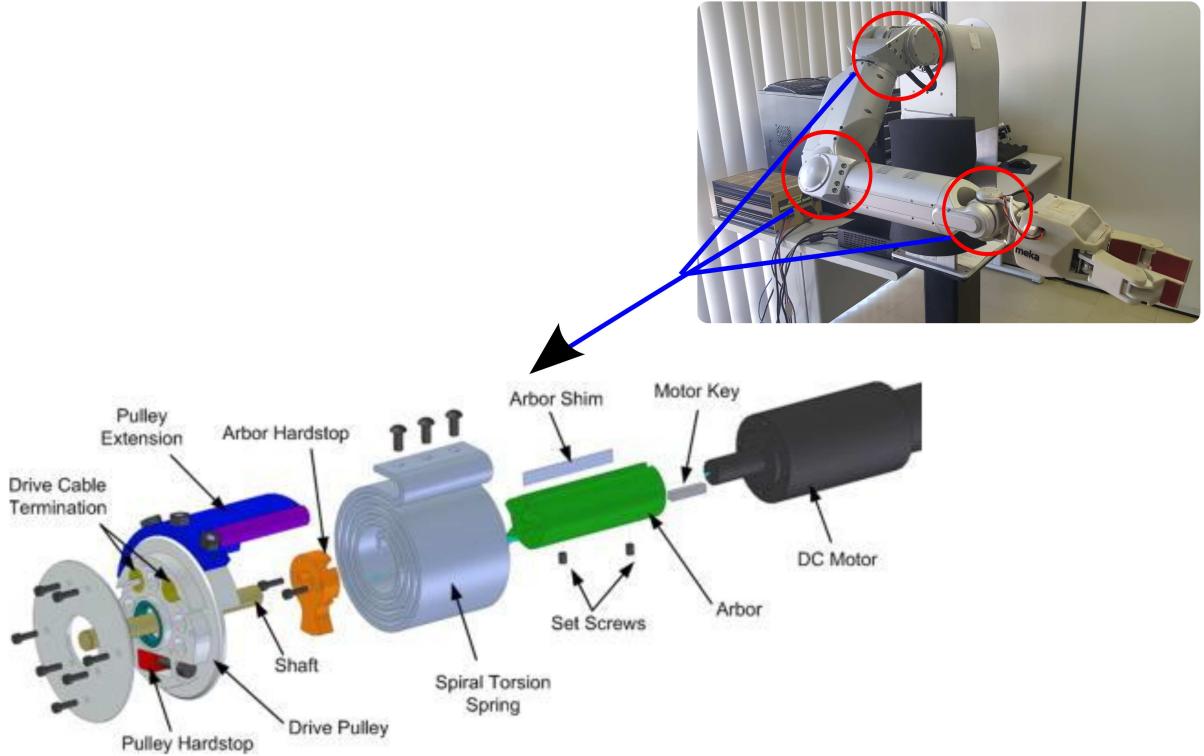
4.2.1 Actuators

The robot makes use of Series Elastic Actuators (SEA) which are compliant actuators with force control characteristics. They utilize BDC/BLDC brushless motors and a Harmonic Drive gearhead for reduction. All joints of Meka are equipped with them. Figure 4.3 shows an illustration of where they are located at Meka.

In [10], Pratt proposes that incorporating series elasticity within an actuator delivers good results and presents the trade-off of low and high stiffness. It is common to say “the stiffer the better” because it increases precision, stability and bandwidth of position-control. Therefore, open-loop positioning or colocated feedback have a decrease in end-point position error under load disturbances. In feedback systems where the position sensor is located at the load side of the interface, the need for corrections in response to load variations is decreased and the resonant frequency of the motor inertia and interface compliance is raised. Hence, the bandwidth of the position control feedback loop is raised while staying stable.

Although high stiffness has some benefits, electric motors have poor torque density and thus can generate high torque density only at high speed. In order to accelerate heavy loads, gear reduction is needed, but introducing friction, backlash, torque ripple and noise. One more negative point of

gear trains is the reflected inertia and high backdrive-friction which can cause damage to objects in contact to the load axis.



(a) SEA² illustration

Figure 4.3: Illustration of where the SEA are located at the robot. Note that the SEA in this figure is a general SEA example and not the specific SEA which the Meka robot uses.

The SEA makes use of an elastic (spring) element between the gear train and the load. Figure 4.3 shows an example of a SEA and illustrate where the Meka SEA are located. The spring deflection under load is measured and used as a force-feedback signal for control. Human safety is therefore enhanced because the spring decouples the motor inertia from the link during impact. As mentioned by Pratt in [10], series elasticity works as a low-pass filter which reduces peak gear forces. Interface elasticity between gear train and load can increase shock tolerance while maintaining small motion bandwidth. Moreover, in a SEA force control, it is transformed into a position control problem enhancing accuracy.

4.2.2 Software

The Meka software comes with an EtherCAT real-time control system using the Meka software package. It handles the joint level controls and dynamics of the arm. In order to interface with the joint level controllers, Python scripting is used. Moreover, it is also possible to use Robot Operating System (ROS) interfaces which allow the use of C++ programming. The robot also has

²<http://mechanicaldesign.asmedigitalcollection.asme.org/data/Journals/JMDEDB/27913/008912jmd3.jpeg>

a pre-configured Real Time PC (RTPC). The pre-installed software is: Ubuntu Linux compiled with RTAI², EtherLab EtherCAT Master, Meka M3 software and the ROS software created by Willow Garage [40] and enhanced since then.

4.2.3 Controllers

Control of the robot can be made at different levels: whole body control of posture, simultaneous control of a group of joints, independent control of each joint, independent calibrated control of each joint, independent calibrated control of each actuator and independent uncalibrated control of each actuator. Status data from the robot high-level posture control of the whole body circles all the way until the actuator low-level DSP PID controllers. All this layers are continuously exchanging data. If desired, a Python Client may publish information to an intermediate component as long as higher level components are disabled and placed in safe operation mode (SAFEOP) mode. Figure 4.4 shows the information flow diagram. For this work, the robot higher level whole body posture control was used which comes with different control modes. The default mode is the OFF mode at start-up and in the case of errors. In this mode the motor amplifier voltage and current is zero.

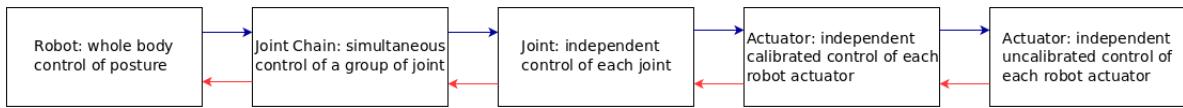


Figure 4.4: Meka robot: continuously flow of information between higher level controllers (left) and lower level controllers (right).

The most simple way of sending an actuation signal for a joint power system is using the PWM mode. It allows to directly set the voltage or current at the desired motor. The value is limited to `pwm_max` found in the lowest-level actuator component .yaml configuration file of the robot.

TORQUE mode allows to control the desired joint torque τ_q using an underlying DSP PID controller. The torque range of each joint is specified in the `min_tq` and `max_tq`. They are also written in an actuator .yaml configuration file.

Similar to TORQUE mode, the TORQUE_GC mode controls the joint torque with the addition of a feed-forward term τ_g which describes the torques due to gravity. To obtain this term, it is used the kinematic and dynamic model of each component provided in configuration files of the robot. The τ_g is subtracted from the commanded joint torque τ_d and passed to the TORQUE mode controller:

$$\tau_q = \tau_d - \tau_g. \quad (4.1)$$

It is possible to include inverse-dynamics velocity and acceleration terms by setting flags in each component configuration file.

For kinematic control, there are higher level controllers. The THETA mode directly control the joint angle using a PID controller. Joint limits are also specified by `min_q` and `max_q` in .yaml

²<https://www.rtai.org/>

configuration files for each joint. The input PID joint angle is slewed to the desired joint angle. The maximum velocity of the joint is set on the slew filter and the maximum slew rate (deg/s) is specified in the q_slew of each joint .yaml configuration file. It comes already tuned for safe operation of the robot.

Analogous to the TORQUE_GC, there is also the THETA_GC mode in which a PID controller computes a desired joint torque τ_s that sets to zero the joint angle error. It also make use of a stiffness parameter s which has the effect of scaling the commanded torque which is delivered to the underlying TORQUE_GC controller:

$$\tau_d = \tau_s s. \quad (4.2)$$

As with THETA mode, THETA_GC mode also utilizes the slew filter on the commanded joint angle.

It is also possible to use a minimum-jerk filter on the commanded joint angle passed to the underlying THETA controller using THETA_MJ mode. This filter allows a smoother trajectory. Instead of setting a slew rate, THETA_MJ requires the setting of a desired velocity. Similar to THETA_GC, there is also the THETA_GC_MJ mode which passes the desired joint position to a THETA_GC controller which allows both smooth and compliant control of the joint.

In this work, the THETA_GC mode was used. According to the robot documentation, it should do the gravity compensation, since it has the TORQUE_GC controller working in the background. However, during the kinematic controllers tests on the robot, the gravity influence was visibly seen acting on the robot. This happens due to the high compliance of the robot inherent to its construction and use of the SEA.

4.3 Robot Operating System (ROS)

Integration of different systems is not an easy task when there is a variety of communication protocols, routines, hardware and software architectures. This is also the case for robots which have a continually scale growth. The ROS³ is a suitable framework created with the goal of integrating large-scale robotics research [40].

In this sense, ROS was used in this work because each robotic tool being used works separately, but needs to exchange information every now and then. Furthermore, both the V-REP simulator and the Meka software allow integration with ROS. Considering also future works, ROS will also be useful to add computer vision systems and other robotic tools and sensors to improve the range of applications for Meka, for instance, the integration with the haptics platform. For each different system and sensor there will be at least one ROS node and one ROS topic.

The main ROS tools used in this project are ROS nodes⁴ and topics⁵. A node can be understood as a process that performs computation and a topic is a bus over which nodes exchange messages.

³<http://www.ros.org/>

⁴<http://wiki.ros.org/Nodes>

⁵<http://wiki.ros.org/Topics>

A node can have publishers which publish messages to a topic and subscribers which subscribe to a topic and receive messages sent via the topic. The messages are strongly typed specifically for the subscriber which must know the message type it will receive otherwise the message transport will not be established.

In the context of this project, ROS is applied to exchange data between the control module, the robot joint actuators, the force sensor and the V-REP simulator. In other words, the control module must know the position of the robot joints of the Meka robot or of the Kuka robot in V-REP at each control loop iteration. On the other hand, the robots need to receive the commanded joints position from the control module. Further sensors (for instance, vision and control interfaces) added in the setup will also need to send information to the control module. Hence, each of the system modules or sensors will have a ROS node which will have publishers and subscribers to transfer messages via topics. A simplified communication graph between the platforms can be seen in Figure 4.5.

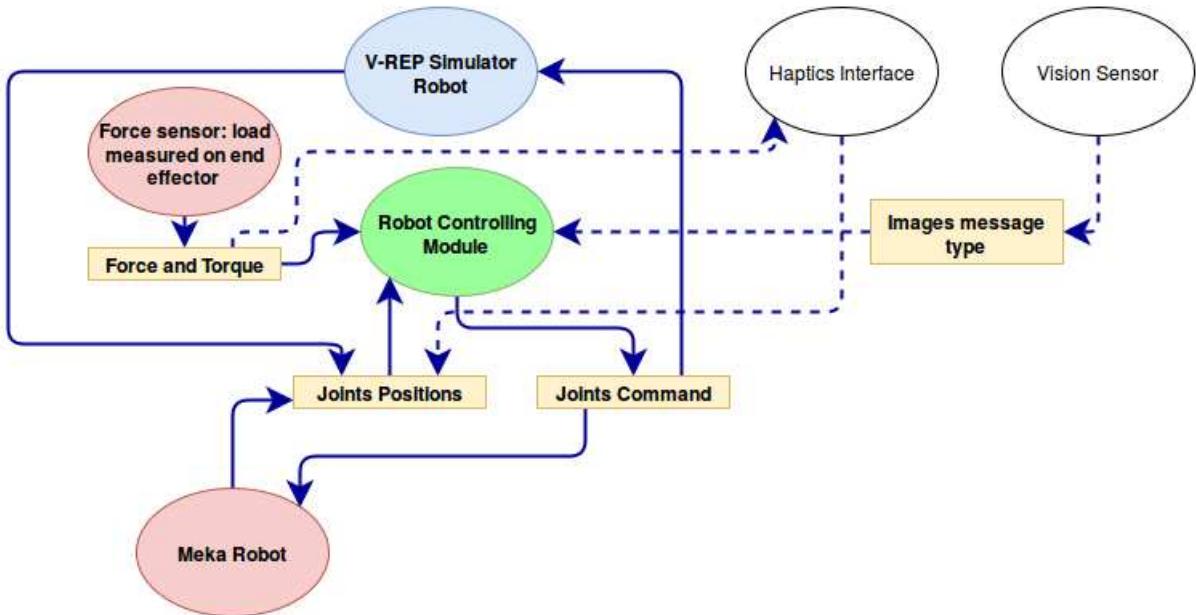


Figure 4.5: System modules overview.

4.4 DQ Robotics and Eigen

The rigid body motions represented using the dual quaternions were introduced in Chapter 2. It was shown that a lot of mathematical operations with vector, matrices and dual quaternions are executed. Therefore, it is needed to choose a suitable way of implementing them. The controllers are being implemented in C++ using ROS, however there are no standard libraries to deal with more complex algebraic operations. In this sense, the DQ Robotics and the Eigen libraries were chosen. DQ Robotics⁶ is a standalone open-source robotics library which provides quaternion algebra and kinematic algorithms for robot modeling and control. It is possible to use it in

⁶<http://dqrobotics.sourceforge.net/>

MATLAB, Python and C++. Therefore it suits very well to integrate with V-REP and the Meka software. Eigen⁷ is a C++ library for linear algebra. It provides matrices, vectors and numerical solvers to operate them. DQ robotics uses a lot of elements from Eigen. Hence it is also a good choice to combine both to implement the kinematic controllers.

4.5 Programming Framework

In Figure 4.6, the ROS computation graph of the system framework used in this work can be seen.

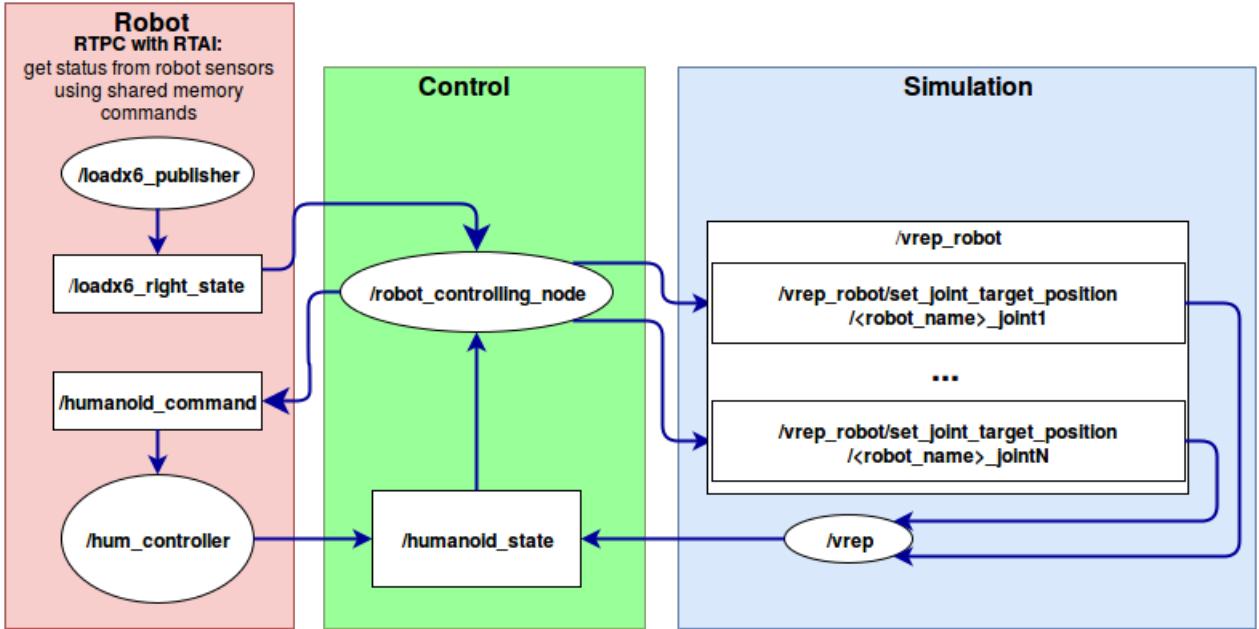


Figure 4.6: ROS computation graph integrating the Meka robot, the control node and the V-REP simulation environment.

- Nodes

- `/robot_controlling_node`: This is a C++ node which runs the kinematic controllers implemented using DQ Robotics and Eigen. It publishes the Meka robot joint commands to the `/humanoid_command` topic and subscribe to the `/humanoid_state` topic to receive the robot joints state information. To publish joint commands to the V-REP simulator it uses the `/vrep_robot` topic.
- `/vrep`: This is a node created by the V-REP simulator which has services to exchange information with the simulation environment. It publish the robot joint states to the `/humanoid_state` topic and subscribe to the `/vrep_robot` topic to receive the commanded robot joint states from the control module.
- `/hum_controller`: This is a C++ node created by the real time shared memory module of the robot which get the status of each robot joint sensor. It publish the robot joint

⁷<http://eigen.tuxfamily.org>

states to the `/humanoid_state` topic and subscribe to the `/humanoid_command` topic to receive the commanded robot joint states from the control module.

- `/loadx6_publisher`: This is a C++ node created by the real time shared memory module of the robot force sensor which get the status of the forces and torques on each axis of the force sensor. It publish the forces and torques to the `/loadx6_right_state` and `/loadx6_left_state` topic. Since the available robot only has the right arm, only the right sensor status is presented.

- Topics

- `/humanoid_command`: Contains the joints command being sent to the robot. In this work, the Meka robot.
- `/humanoid_state`: Contains the joints state being read from the Meka robot or from the robot in the V-REP simulation environment.
- `/vrep_robot`: Contains the joint command message which will be sent to the robot in the V-REP simulation environment through the node `/vrep`.
- `/loadx6_right_state`: Contains the force and torques on each axis read from the force sensor.

Given the ROS nodes connections, the main development and implementation tasks of this work were done in C++ in the robot controlling node. An overview of the control node architecture can be seen in Figure 4.7.

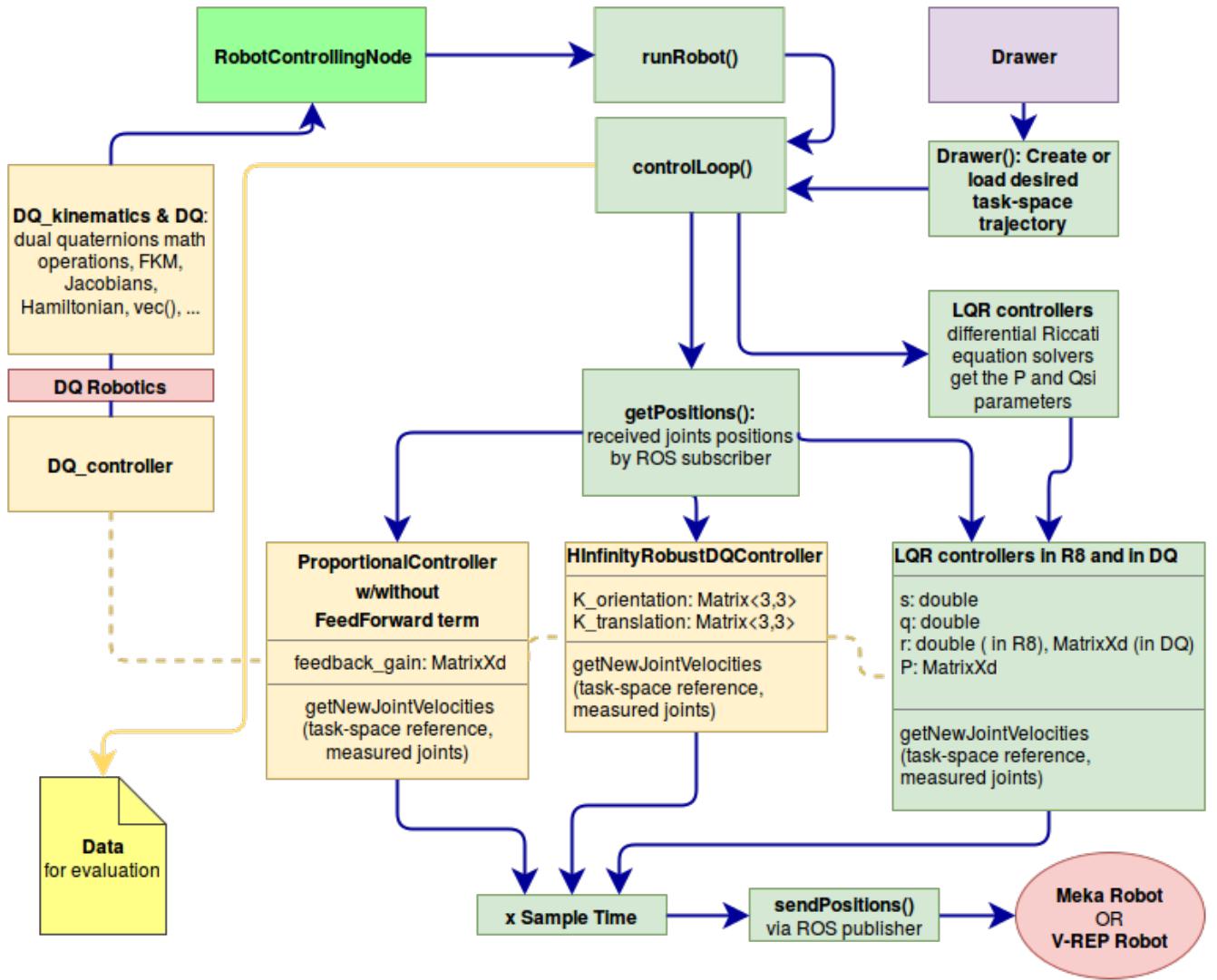


Figure 4.7: Overview of the robot controlling node, the use of the DQ Robotics library and the Drawer class.

In Figure 4.7, note that the Drawer is a class defined outside of the robot controlling node. It is used to create or load trajectories. The trajectory draw can be done in two ways. One of them is to describe the linear and circular segments of the trajectory in a .txt file following a specified syntax. Then, the Drawer load this file and generate a discrete trajectory based on the one specified. The second option is to write all the trajectory points as cartesian coordinates in a .txt file and then load it. Both methods deliver the task-space trajectory used as reference by the controllers.

To keep simplicity and make easier the future use of the controllers, they were implemented similarly to the DQ Robotics controllers class and were defined outside of the robot controlling nodes. Note that the LQR controllers were not yet adapted to the DQ controllers class. All the DQ controllers can be used for set point control. However, the LQR controllers needs to have previous knowledge of the whole trajectory in order to calculate their gain matrices. In this sense, they were implemented as methods of the robot controlling node.

4.6 How to start working with the Meka Robot

There are different ways to control the Meka robot. Section 4.2 gives an overview of the robot and Section 4.5 explains the ROS programming framework used to control the robot. In this section, the basic steps on how to start the robot system will be introduced. The following steps will start the robot basic modules and create the robot ROS nodes which publish data from sensors and receive commands for the robot actuators:

1. Turn on the Meka robot and the Meka robot computer;
2. Start ROS on a terminal by running `roscore`;
3. Open a new terminal and start the program which identifies and communicate with the available robot components and start the RTAI⁸ procedures by running the command `m3rt_server_run -m` on the terminal;
4. Open another terminal and start the shared memory program (Figure 4.6: Robot) which creates the ROS nodes which publish the robot sensors data to ROS topics and receive commands to be sent to the robot actuators. The shared memory program is started by running the command `rosrun shm_humanoid_controller shm_humanoid_controller` on the terminal;
5. At this point, data from the robot sensors can be obtained via ROS and joint commands from the robot controlling node can be sent to the robot. The ROS computation graph was presented in Figure 4.5.

This section presented the basic modules which must be started to control the robot. However, to use the Meka robot at the LARA laboratory, the detailed instructions available on the LARA GitHub Meka Wiki⁹ must be read. Further details about the robot software and hardware and instructions on how to integrate new sensors will also be added. The Meka Wiki will be constantly updated to ease the development of future projects involving the A2 Arm robot at LARA.

4.7 Conclusion

This chapter presented the robotic tools used in this work. It explained the need for a robotic simulation environment in order to check the stability of the kinematic controllers and described the V-REP simulator with the Kuka LBR iiwa 7. Then, the robot A2 Arm from Meka Robotics and its underlying controllers and software were discussed. For future works, the Meka robot virtual model may be implemented on the V-REP simulator for more precise simulation results. As a result of both V-REP and Meka being able to use ROS integration, and due to ROS availability of lots of sensors which may be added in future works, it was selected as the communication

⁸<https://www.rtai.org/>

⁹<https://github.com/lara-unb/Meka>

system across platforms. The dual quaternion and linear algebra libraries, DQ Robotics and Eigen respectively, were chosen as programming tools to implement the kinematic controllers. Lastly, the programming framework developed to test and implement the controllers in C++ was described and a short introduction on how to start working with the Meka robot is done.

Chapter 5

Evaluation and Results

Concerning the practical application of the kinematic controllers for robot manipulators implemented in this work, it is necessary to test them in simulations and in a real robot before using them in HRI applications. In Chapter 5, first, the methodology analysis is presented by defining tasks, evaluation criteria and parameter selection. Afterward, experimental results on the real robot are shown and analyzed in order to bring up the advantages and disadvantages of each controller. Then, in order to evaluate the frequency and time periods achieved by the controllers, the sampling time histograms along the trajectory are presented. Lastly, images of a simple light painting application which may be extended and used in educational contexts are presented.

The results of this work presented in this chapter contain the experimental evaluation of the proportional controllers with and without feedforward term defined in (3.8) and (3.12), respectively, the LQR controller in \mathbb{R}^8 manifold (LQR in \mathbb{R}^8) given by (3.45) and in the dual quaternion space (LQR DQ) given by (3.60), and the two robust H_∞ dual quaternion controllers described by (3.30) and (3.33) which are a setpoint (H-infinity robust - HIR) and a tracking controller (H-infinity robust tracking - HIRT), respectively. Table 3.1 presents an overview of the controllers equations. The controllers were implemented in C++ and executed on the Meka robot as described in details in Chapter 4. Figure 5.1 present the coordinate axes on the robot.

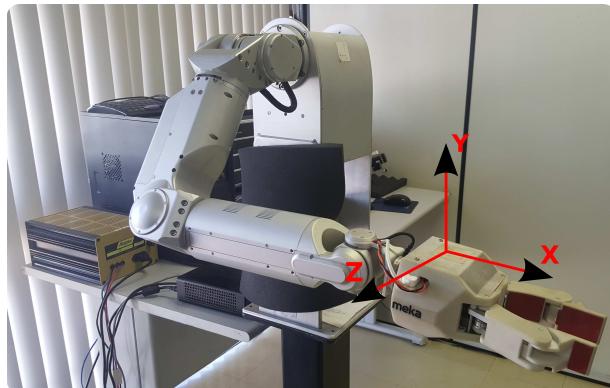


Figure 5.1: Coordinate axes on the Meka robot used on the experiments.

Four experiments were devised and their results are presented together with the controllers performance evaluation. Table 5.1 presents an overview of the four experiments. More experiments were also done in the authors previous work [29] and in Appendix C.

Table 5.1: Overview of all experiments

Experiment	Goal
LQR controllers performance analysis	Evaluate both LQR controllers performance regarding q and r influence by executing a circular trajectory 10 times for each parameter set
H_∞ controllers performance analysis	Evaluate both H_∞ controllers performance regarding noise to error amplification executing a spiral trajectory 5 times for each parameter set
Circular trajectory	Compare all controllers executing a circular trajectory 10 times for each parameter set
Spiral trajectory	Compare all controllers performance and repeatability executing a spiral trajectory 50 times for each parameter set

5.1 Tasks Definition

In order to demonstrate results of the controllers behavior under different set of parameters, different experiments were run on the A2 Arm Meka Robotics manipulator with seven degrees of freedom (DOF). The control algorithms stability was also previously checked on the V-REP simulator with the Kuka LBR iiwa 7 (Figure 4.1).

The motivation for the tasks trajectories lies on the need for robots to work alongside humans, they need to accomplish repetitive tasks. Therefore, it is investigated the ability of the robot to follow simple trajectories which would be tiring for a human to do for a long time. Due to anthropomorphic arm joint limitations it is always considered initial robot configurations θ_0 such that the trajectory lies within the end-effector reachable space. In other words, θ_0 must allow the robot to execute the whole trajectory¹.

In most of the tasks chosen, the controllers were set with an intentional initial error to evaluate the smoothness of each controller convergence. This means that the robot end effector initial position was not at the first point of the devised tasks. Therefore, all the controllers yield larger control signals θ_0 in the beginning of the trajectory in order to correct the trajectory error.

In this sense, it turns out that if a high control gain is set right at the start of the desired trajectory, it may yield high initial joint velocities which can lead the controller to an unstable behavior. For this reason, when selecting parameters for the controllers, some of them were varied

¹Avoidance techniques for kinematic singularities and joint limits were not the main goal of this work. A short review on workspace singularities was done in [21, Section 3.3] and singularity avoidance techniques using dual quaternions were presented in [14, ch. 5].

along the trajectory. According to their influence on the controller they were initially set to low or high values to deliver a smoother initial error decay. After the manipulator reaches a stable pose, the gains are tuned in order to enhance the controller performance while keeping internal stability.

5.2 Evaluation Criteria

In order to compare performance, evaluation criteria which allow to understand the benefits of each controller were adopted. Before mentioning them, some concepts are explained. It is necessary to understand the meaning of the invariant error norm, joints control effort and control signal.

The invariant error norm computation was defined in Section 2.2.3. It gives the error between the trajectory desired pose $\underline{\mathbf{x}}_d(t)$ and the robot pose $\underline{\mathbf{x}}(t)$ at each time instant t

$$Err(t) = 1 - \underline{\mathbf{x}}^*(t) \underline{\mathbf{x}}_d(t). \quad (5.1)$$

One of the main contributions of the LQR controllers derived in Section 3.4 is its ability to achieve a trade-off between control effort and invariant error norm. In this work, joint control effort is defined as the difference between two consecutive joint positions of the robot. It means how much the robot joints turned between two measured samples and it is given by

$$Eff = \vec{\theta}(k+1) - \vec{\theta}(k). \quad (5.2)$$

Similar to the control effort, the control signal is also being analyzed. The control signal is the $\dot{\vec{\theta}}$ computed for each controller in equations (3.8), (3.12), (3.30), (3.33), (3.45) and (3.60). This value is useful to demonstrate that the joints control effort in the real robot are much lower than the control signal as expected. This occurs in order to prevent damage to the robot actuators. Joint velocities commanded by the controller are much greater than the angular velocities achieved by the robot joints.

Lastly, to better understand the physical meaning of control effort and the controller parameters effect on the robot trajectory tracking ability, the end effector linear velocity (*L.Vel*) and acceleration (*L.Acc*) is also being analyzed.

All of the evaluation criteria used are graphically and numerically presented. Each graph presents the norm of the value at each time instant. Since *Eff.* is a vector, the norm of the joints control effort at each time instant t is the quadratic norm of $\vec{\theta}(k+1) - \vec{\theta}(k)$, the same goes for control signals $\dot{\vec{\theta}}$. The velocities and acceleration norms are obtained of the end effector linear velocity and linear acceleration vectors, respectively, at each time instant t . The robot pose trajectory error is already obtained as the invariant error norm.

The numerical evaluation of each criteria is presented in tables and their value is calculated as the integral of the norm of the criteria along the whole trajectory time, such as

$$\text{Integral Norm} = \int_{t_0}^{t_f} \| \text{criteria} \|_2 dt. \quad (5.3)$$

For instance, the numerical evaluation of the end effector linear velocity during a trajectory is the integral of the linear velocity norm along the trajectory time frame.

For all the proceeding analysis, the following notation is used.

Err. is the invariant error norm.

$\dot{\theta}$ [rad/s] is the norm of the control signal.

Eff. [rad] is the norm of the control effort.

L.Vel. [m/s] is the end effector velocity norm.

L.Acc. [m/s²] is the end effector acceleration norm.

The plots show the norm of the criteria along the trajectory and the tables shows the integral norm of the criteria along the trajectory.

5.3 Parameters Selection

5.3.1 Proportional Controller with and without Feedforward Term

The K and K+FF controllers described in (3.8) and (3.12), respectively, make use only of the proportional gain $k \in \mathbb{R}$ such that $\mathbf{K} = k\mathbf{I}$ and $\mathbf{I} \in \mathbb{R}^{8 \times 8}$. The K and K+FF controllers are bounded to the overall effect of the proportional gain k . The choice of k was done by trial and error elevating the k value until there would be no more improvement in the integral norm of the error without an excessive increase in the joint control effort.

5.3.2 LQR Controllers

In order to simplify both LQR controllers choice of parameters, it is defined $s, q, r \in \mathbb{R}$ such that

$$\mathbf{S} = s\mathbf{I}, \mathbf{Q} = q\mathbf{I}, \mathbf{R} = r\mathbf{I},$$

where $\mathbf{I} \in \mathbb{R}$ is an identity matrix.

The choice of parameters is closely related with the task optimization goals. For instance, it is supposed that the end-time error is not of higher importance than the error in the remainder of the trajectory by setting $s = 0$. The choice of q is done in order to keep the system stable during the whole trajectory and is closely related with r . Because the r parameter weighs the control effort in terms of end effector velocity norm, for a higher r , the joint velocities will be lower and

it is needed to compensate with a higher q , that is, a “stronger” error correction demanding faster correction, in order to obtain an acceptable minimization overall.

The optimal dual quaternion based controller in \mathbb{R}^8 manifold, or LQR in \mathbb{R}^8 , has the following gain matrices $\mathbf{S}, \mathbf{Q}(t) \geq 0$ and $\mathbf{R}(t) > 0$ with $\mathbf{S}, \mathbf{Q}, \mathbf{R} \in \mathbb{R}^{8 \times 8}$ and the optimal quadratic controller in dual quaternion space, or LQR in DQ has the matrices $\mathbf{S}, \mathbf{Q}(t) \geq 0$ and $\mathbf{R}(t) > 0$ with $\mathbf{S}, \mathbf{Q}, \mathbf{R} \in \mathbb{R}^{6 \times 6}$.

In order to compare performance, the LQR controllers are distinguished by varying the weight of the control effort parameter, that is r is increased and decreased to evaluate its influence on the trajectory. In all controllers, the choice of q allows a well-behaved initial motion.

5.3.3 Dual Quaternion H_∞ controllers

The two H_∞ controllers, that is, the H-infinity robust setpoint controller (HIR) and the H-infinity robust tracking controller (HIRT) make use of the gain matrices $\mathbf{K}_O, \mathbf{K}_T \in \mathbb{R}^{3 \times 3}$ which are respectively the orientation and translation gain matrices. Both gain matrices depends on the induced norm upper bound variable $\gamma > 0$, which can be specified as $\gamma_{O_1}, \gamma_{O_2}, \gamma_{T_1}, \gamma_{T_2}$. To ensure exponential H_∞ tracking performance, the gain matrices must be defined as

$$\begin{aligned}\mathbf{K}_O &\geq k_O \mathbf{I} = \sqrt{2} \sqrt{\gamma_{O_1}^{-2} + \gamma_{O_2}^{-2}} \mathbf{I} \text{ and} \\ \mathbf{K}_T &\geq k_T \mathbf{I} = \sqrt{\gamma_{T_1}^{-2} + \gamma_{T_2}^{-2}} \mathbf{I}, \\ k_O, k_T &\in \mathbb{R}^+.\end{aligned}\tag{5.4}$$

Considering it is difficult to decouple both sources of uncertainties on the real robot, the induced norm upper bound variable γ may be set the same, that is, $\gamma_T = \gamma_{T_1} = \gamma_{T_2}$ and $\gamma_O = \gamma_{O_1} = \gamma_{O_2}$ as mentioned in [14, ch. 3]. Applying this to (5.4), the following gain relations are obtained

$$k_O \gamma_O = 2\tag{5.5}$$

$$k_T \gamma_T = \sqrt{2}.\tag{5.6}$$

5.4 Linear Quadratic Optimal Control (LQR) performance analysis

In order to evaluate both LQR controllers performance, a similar experiment to the one presented in Appendix C was repeated. Table 5.2 presents an overview of the experiment of this section.

Table 5.2: Overview of experiment to evaluate the LQR controller performance

Trajectory	Circle on xz-plane (Figure 5.1)
Initial joint configuration	$\theta_0 = [0, \pi/5, 0, \pi/2, 0, \pi/5, 0, 0]$
Trajectory parametrization	$x = r \sin(\theta), y = 0, z = r \cos(\theta), \theta \in [0, 2\pi]$
Radius	$r = 7 \text{ cm}$
Sampling time	20 ms
Trajectory points	3000
Parameter sets	4 sets: $r \in \{0.001, 0.01, 0.1, 1\}$ and $q \in \{1, 10, 100, 1000\}$
Executions	10 times for each parameter set

To exploit the performance obtained before, a new sampling time of $T_s = 20ms$ was used in comparison with the $25ms$ sampling time of Section C. Moreover, the number of the circular trajectory points was doubled from 1500 to 3000 steps in comparison with the experiments in Appendix C.

The devised task is a circle with a radius $r = 7 \text{ cm}$ and is drawn on the XZ-plane (Figure 5.1) while maintaining Y constant, that is,

$$\begin{aligned} x &= r \sin(\theta), \theta \in [0, 2\pi], \\ y &= 0, \\ z &= r \cos(\theta), \theta \in [0, 2\pi]. \end{aligned} \quad (5.7)$$

In order to numerically compare performance with the evaluation criteria, the LQR in \mathbb{R}^8 and the LQR in DQ controllers are distinguished by choosing a increasing weight for the control effort and error parameters r and q , respectively, that is $r \in \{0.001, 0.01, 0.1, 1\}$ and $q \in \{1, 10, 100, 1000\}$. The same procedure was followed in Appendix C. For each gain combination the same trajectory was executed 10 times. The 10 experiments results can be found in Appendix Sections E.1-E.4. All the other controllers were also run and their results will be used later in this chapter.

Tables 5.3 and 5.4 shows the results for the trajectories run on the Meka robot with the LQR controller \mathbb{R}^8 based and the LQR controller DQ based, respectively.

Table 5.3: Trajectory results obtained on Meka performing the circular end effector trajectory described by (5.7) with the LQR in \mathbb{R}^8 : $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s²].

LQR controller in \mathbb{R}^8								
Q	1		10		100		1000	
R	0.001		0.01		0.1		1	
	Mean	Std. Dev.						
<i>Err.</i>	4.882	0.126	4.858	0.138	5.034	0.122	4.853	0.218
$\dot{\theta}$	307.08	6.35	304.87	6.70	313.98	5.99	306.48	10.17
<i>Eff.</i>	0.111	0.001	0.109	0.002	0.111	0.002	0.108	0.001
<i>L.Vel.</i>	0.968	0.012	0.969	0.010	0.966	0.007	0.960	0.010
<i>L.Acc.</i>	24.094	3.004	25.105	3.837	23.235	0.934	22.648	0.498

Table 5.4: Trajectory results obtained on Meka performing the circular end effector trajectory described by (5.7) with the LQR in DQ: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s²].

LQR controller in DQ								
Q	1		10		100		1000	
R	0.001		0.01		0.1		1	
	Mean	Std. Dev.						
<i>Err.</i>	5.593	0.182	5.635	0.209	5.855	0.168	5.688	0.344
$\dot{\theta}$	318.22	9.20	319.96	9.94	330.54	8.49	325.35	16.37
<i>Eff.</i>	0.114	0.003	0.116	0.003	0.118	0.004	0.116	0.003
<i>L.Vel.</i>	0.998	0.029	1.007	0.024	1.013	0.037	1.016	0.025
<i>L.Acc.</i>	24.730	1.534	25.834	1.357	26.254	2.035	26.171	1.569

In comparison to the simulation results presented in Table 3 in the Appendix Section C, when the trajectory was run on the Meka robot, the results show a larger variation. Table 5.3 showed the LQR in \mathbb{R}^8 controller performance oscillated with the increase in r and q . As expected, as long as r or q are not high enough, the error norm and the control effort will decrease with an increase in q and r . The control effort will be more influenced by the q increase than by the r increase. A lower $r = 0.001$ with a lower $q = 1$ showed an increase in control effort which happens because the $r = 0.001$ have a lower weigh on the control effort minimization. Therefore, the control signal $\dot{\theta}$ is also higher than with $r = 0.01$. For $r = 1$ and $q = 1000$ it is seen although q was increased and the error decreased, the control effort did not increase because of the effect of r . In this sense, the error decay is smoother. Similarly, Table 5.4 showed the LQR in DQ had an increase in the error norm with the increase in r until the error weigh q start to influence the error convergence over the effect of r . It is also seen r kept the control effort lower than 0.12 even though q was increased for both LQR controllers which lowers the error.

For a more in depth analysis of the LQR controller in \mathbb{R}^8 performance in the simulation environment and on the Meka robot, Appendix D show the results of a wider range of q and r values variation and their influence on the control effort, the error norm and the end effector velocity.

5.5 Dual Quaternion H_∞ Controllers Performance Analysis

The goal of this experiment is to check that the H_∞ performance is being achieved. Table 5.5 presents an overview of the experiment of this section.

Table 5.5: Overview of experiment to evaluate the H_∞ controller performance

Trajectory	Spiral trajectory along the x-axis (Figure 5.1)
Initial joint configuration	$\theta_0 = [0, \pi/3.7, 0, \pi/2, 0, \pi/3.7, 0, 0]$
Trajectory parametrization	$x = l, y = r \sin(\theta), z = r \cos(\theta), \theta \in [0, 6\pi]$
Radius	$r = 5$ cm
Height	$l = 5$ cm
Sampling time	8 ms
Trajectory points	3000
Parameter sets for $t > 0.1t_f$	5 sets: $\gamma = \{0.372, 0.465, 0.620, 0.930, 1.86\}$
Executions	5 times for each parameter set
Gain variation	Gains varied for $t \leq 0.1t_f$ as described in (5.9)
Error norm	Evaluated for $t > 0.1t_f$ while all gain values were constant

The dual quaternion H_∞ controllers performance is such that the noise to error amplification—directly related to (3.22) in Chapter 4 — should apply for a trajectory in the time interval $[t_0, t_1]$ with $t_0, t_1 \in \mathbb{R}^+$

$$\frac{\int_{t_0}^{t_1} \| \text{vec } \mathbf{e} \|_2}{\int_{t_0}^{t_1} \| \underline{v} \|_2} \leq \gamma, \quad (5.8)$$

where \underline{v} represent the disturbances and uncertainties to which the manipulator is subjected. In [14], \underline{v} was estimated in the V-REP simulation environment for the KUKA LBR IV arm, however, it is out of the scope of this work to precisely estimate \underline{v} acting on the real Meka robot.

Therefore, to evaluate the H_∞ performance given a desired $\gamma_d = \gamma_{Od} = \gamma_{Td}$ yielding gain matrices \mathbf{K}_{Td} and \mathbf{K}_{Od} , the orientation gain matrices were reduced to $0.2\mathbf{K}_{Od}$, $0.4\mathbf{K}_{Od}$, $0.6\mathbf{K}_{Od}$ and $0.8\mathbf{K}_{Od}$. The same was applied to the translation gain matrices which were also obtained as $0.2\mathbf{K}_{Td}$, $0.4\mathbf{K}_{Td}$, $0.6\mathbf{K}_{Td}$ and $0.8\mathbf{K}_{Td}$. Assuming that \underline{v} remains constant for the manipulator, then, the error norm integral is directly proportional to γ as seen in (5.8).

Given a desired γ_d , \mathbf{K}_{Od} and \mathbf{K}_{Td} are calculated using (5.5)-(5.6). Then, γ is calculated for each \mathbf{K} percentage using (5.5)-(5.6).

Considering the error in the beginning of the trajectory will be higher as mentioned in Section 5.1, the gain matrices \mathbf{K}_O and \mathbf{K}_T initial values at time instant t_0 were lowered. They were then

increased along the trajectory until they reach their final value, such that

$$\gamma_d(t_f) = \gamma_O(t_f) = \gamma_T(t_f) = 0.372$$

$$k_O(t) = \begin{cases} 2.8 & t = t_0 \\ 2.8 + \frac{t}{0.1t_f} 2.5 & t \leq 0.1t_f \\ 5.3 & t > 0.1t_f \end{cases}$$

$$k_T(t) = \begin{cases} 2.4 & t = t_0 \\ 2.4 + \frac{t}{0.1t_f} 1.4 & t \leq 0.1t_f \\ 3.8 & t > 0.1t_f \end{cases}, \quad (5.9)$$

and, then, $\mathbf{K}_O(t) = k_O(t)\mathbf{I}$ and $\mathbf{K}_T(t) = k_T(t)\mathbf{I}$.

The $\gamma_d = 0.372$ parameter was found by an heuristic method such that the controller would deliver the lowest integral error norm without an excessive increase in joints control effort or too much oscillations during the trajectory. Afterwards, with γ_d , \mathbf{K}_{Od} and \mathbf{K}_{Td} and using (5.5)-(5.6), the respective γ values were obtained. They can be seen in Tables 5.6-5.7.

For this experiment the sampling time was reduced as much as possible considering the control loop frequency. After some code review of the C++ controller implementation, the control loop period was decreased from between $20ms$ and $25ms$ to $8ms$. Therefore, the sampling time for this experiment was set to $T_s = 8ms$. The chosen task was a spiral trajectory with 3000 points, constant radius of $r = 5cm$, length $l = 5cm$ and robot initial joint configuration given by $\boldsymbol{\theta}_0 = [0, \pi/3.7, 0, \pi/2, 0, \pi/3.7, 0, 0]$ such that

$$\begin{aligned} x &= l, \\ y &= r \sin(\theta), \theta \in [0, 6\pi]. \\ z &= r \cos(\theta), \theta \in [0, 6\pi] \end{aligned} \quad (5.10)$$

The trajectory was executed 5 times for each value of γ . The five experiments full tables are in Appendix Section E.5. Tables 5.6 and 5.7 show the effect of decreasing the γ or, in other words, proportionally increasing the value of \mathbf{K}_O and \mathbf{K}_T on the H_∞ setpoint and H_∞ tracking controllers.

Table 5.6: Trajectory results obtained on Meka performing the spiral end effector trajectory described by (5.10) with the HIR controller: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s²].

HIR										
$\gamma = \gamma_O = \gamma_T$	$\mathbf{K}_{20\%}, \gamma = 1.86$		$\mathbf{K}_{40\%}, \gamma = 0.930$		$\mathbf{K}_{60\%}, \gamma = 0.620$		$\mathbf{K}_{80\%}, \gamma = 0.465$		$\mathbf{K}_{100\%}, \gamma_d = 0.372$	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
$Err.$	3.133	0.018	1.660	0.011	1.138	0.009	0.837	0.008	0.661	0.007
$\dot{\theta}$	344.05	2.879	373.86	1.274	392.40	2.013	393.60	2.206	394.64	2.659
$Eff.$	0.038	0.0004	0.050	0.001	0.055	0.0003	0.059	0.0003	0.065	0.001
$L.Vel.$	1.051	0.0128	1.251	0.005	1.346	0.004	1.427	0.014	1.613	0.020
$L.Acc.$	97.195	3.537	111.025	4.149	119.582	1.995	126.120	4.271	139.597	2.681

Table 5.7: Trajectory results obtained on Meka performing the spiral end effector trajectory described by (5.10) with the HIRT controller: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s²].

HIRT										
$\gamma = \gamma_O = \gamma_T$	$\mathbf{K}_{20\%}, \gamma = 1.86$		$\mathbf{K}_{40\%}, \gamma = 0.930$		$\mathbf{K}_{60\%}, \gamma = 0.620$		$\mathbf{K}_{80\%}, \gamma = 0.465$		$\mathbf{K}_{100\%}, \gamma_d = 0.372$	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
$Err.$	3.118	0.015	1.662	0.007	1.125	0.011	0.829	0.011	0.662	0.005
$\dot{\theta}$	342.95	2.519	374.93	1.426	389.44	2.403	390.60	2.918	395.77	2.427
$Eff.$	0.038	0.0003	0.050	0.0004	0.055	0.001	0.059	0.0004	0.065	0.001
$L.Vel.$	1.065	0.008	1.253	0.005	1.337	0.009	1.441	0.012	1.617	0.028
$L.Acc.$	95.224	0.686	109.450	1.073	117.818	2.024	126.929	3.720	138.806	2.038

As expected by (5.8), the results from 5.6 and 5.7 show that if we increase the value of γ , the error integral norm is proportionally increased. For example, $\frac{Err_{\mathbf{K}40\%}}{Err_{\mathbf{K}80\%}} \approx 2$, which means the H_∞ performance was enhanced by 2 when the gain matrices \mathbf{K}_O and \mathbf{K}_T were doubled.

Figure 5.2 depicts graphically the increase by 5 times in the error norm if the gains are 5 times lower. The improvement in the trajectory end effector translation and in the pose orientation quaternion can also be seen.

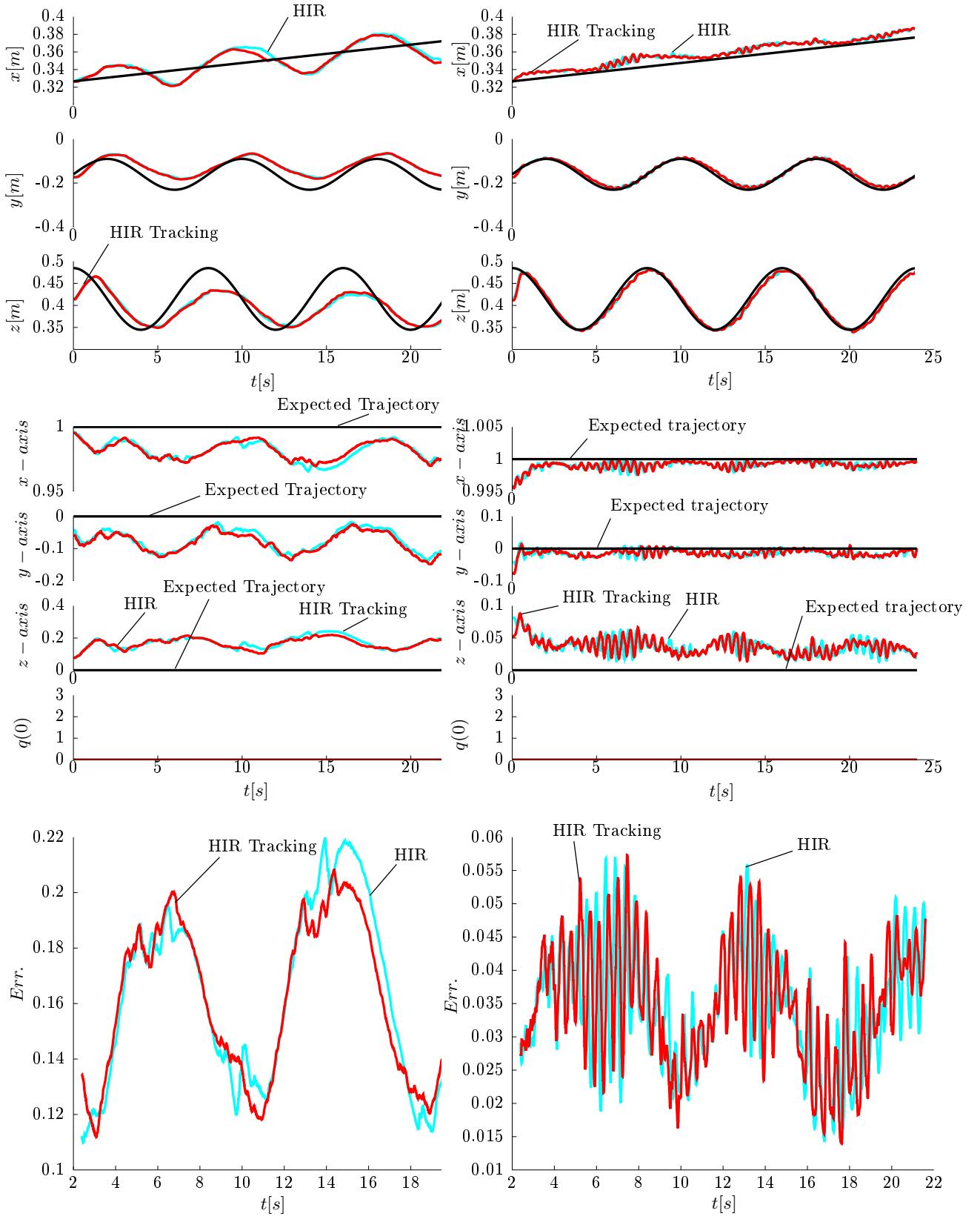


Figure 5.2: Gains adjusted to 20% (left) and 100% (right)

5.6 Circular Trajectory

The proportional controller (K controller), proportional controller with feedforward term (K+FF controller) and the LQR controller in \mathbb{R}^8 manifold (LQR in \mathbb{R}^8) were first evaluated in the authors previous work [29] with a circular and a squared trajectory where the goal was to verify the LQR in \mathbb{R}^8 performance and possible advantages on a real robot. Since then, the gains and sampling time have been tuned to get better results. In Appendix C, results with lower error norm and control effort for similar trajectories are presented.

Motivated by the results obtained in [29] and in Appendix C, more three dual quaternion based kinematic controllers designed in [14] were implemented. The dual quaternion based LQR in the dual quaternion space (LQR DQ) given by (3.60) and two robust H_∞ dual quaternion controllers described by (3.30) and (3.33) which are a setpoint (H-infinity robust - HIR) and a tracking controller (H-infinity robust tracking - HIRT), respectively.

In order to have a first comparisson of all the controllers performance together, the same experiment presented in Section 5.4 was repeated including all controllers. Table 5.8 presents an overview of the experiment of this section.

Table 5.8: Overview of the experiment to compare all controllers using a circular trajectory.

Trajectory	Circle on xz-plane (Figure 5.1)
Initial joint configuration	$\boldsymbol{\theta}_0 = [0, \pi/5, 0, \pi/2, 0, \pi/5, 0, 0]$
Trajectory parametrization	$x = r \sin(\theta), y = 0, z = r \cos(\theta), \theta \in [0, 2\pi]$
Radius	$r = 7 \text{ cm}$
Sampling time	20 ms
Trajectory points	3000
LQR controller parameters	4 sets: $r \in \{0.001, 0.01, 0.1, 1\}$ and $q \in \{1, 10, 100, 1000\}$
Proportional controllers parameters	$k = 50$
H_∞ controllers γ parameters	$\gamma_T = 0.471$ and $\gamma_O = 0.666$
H_∞ controllers gain k variation	$k_T(t) = k_O(t) = 2.5 + \frac{t}{t_f} 0.5$ for $0 < t \leq t_f$ (hence, varied along whole trajectory)
Executions	10 times for each parameter set

The sampling time of $T_s = 20ms$ was used again and the number of circular trajectory points was 3000.

As in Section 5.4, the devised task is a circle with a radius $r = 7 \text{ cm}$ and is drawn on the

XZ-plane (Figure 5.1) while maintaining Y constant, that is,

$$\begin{aligned} x &= r \sin(\theta), \theta \in [0, 2\pi], \\ y &= 0, \\ z &= r \cos(\theta), \theta \in [0, 2\pi]. \end{aligned} \quad (5.11)$$

The results obtained for the proportional controllers and the H_∞ controllers which were not shown in Section 5.4 are presented in Table 5.9. For each parameter variation in the experiment of Section 5.4, all the controllers were executed 10 times. However the mean and standard deviation presented in Table 5.9 takes into account only the experiments from Appendix Subsection E.4. The tables with results for all the 40 experiments can be found in Appendix E.1-E.4.

Table 5.9: Trajectory results obtained on Meka performing the circular end effector trajectory described by (5.7): $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s²].

K		K+FF		HIR		HIRT		
k=50		k=50		$\gamma_T = 0.471, \gamma_O = 0.666$				
Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	
<i>Err.</i>	3.16	0.086	3.16	0.114	1.50	0.024	1.52	0.010
$\dot{\theta}$	314.98	8.740	316.77	10.984	372.60	4.933	376.56	2.705
<i>Eff.</i>	0.117	0.001	0.117	0.002	0.140	0.004	0.141	0.002
<i>L.Vel.</i>	0.986	0.007	0.990	0.006	1.265	0.043	1.289	0.017
<i>L.Acc.</i>	22.945	0.367	23.690	2.428	27.378	1.563	26.941	0.586
LQR in \mathbb{R}^8								
Q	1		10		100		1000	
R	0.001		0.01		0.1		1	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
<i>Err.</i>	4.882	0.126	4.858	0.138	5.034	0.122	4.853	0.218
$\dot{\theta}$	307.08	6.349	304.87	6.701	313.98	5.990	306.48	10.171
<i>Eff.</i>	0.111	0.001	0.109	0.002	0.111	0.002	0.108	0.001
<i>L.Vel.</i>	0.968	0.012	0.969	0.010	0.966	0.007	0.960	0.010
<i>L.Acc.</i>	24.094	3.004	25.105	3.837	23.235	0.934	22.648	0.498
LQR in DQ								
Q	1		10		100		1000	
R	0.001		0.01		0.1		1	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
<i>Err.</i>	5.593	0.182	5.635	0.209	5.855	0.168	5.688	0.344
$\dot{\theta}$	318.22	9.199	319.96	9.944	330.54	8.493	325.35	16.37
<i>Eff.</i>	0.114	0.003	0.116	0.003	0.118	0.004	0.116	0.003
<i>L.Vel.</i>	0.998	0.029	1.007	0.024	1.013	0.037	1.016	0.025
<i>L.Acc.</i>	24.730	1.534	25.834	1.357	26.254	2.035	26.171	1.569

In Table 5.9, it is seen although the LQR controllers allowed to keep a lower control effort while setting the error convergence, they had a worse performance in the integral error norm in comparison to the other controllers. Lastly, as expected, the H_∞ controllers delivered the lowest error norms and only a slight increase of 0.02 in control effort. This makes sense since it was designed considering disturbances rejection and minimum control effort as stated in Theorems 3.1-3.2.

Figure 5.3 illustrates the behavior of the LQR controller for $q = 1$ and $r = 0.001$. The same analysis can be done. As presented in Table 5.9 the LQR in DQ had a worse trajectory tracking and, therefore, higher integral error norm. However, both LQR controllers had lower end effector linear velocity peaks in the beginning of the trajectory. This contribute to their use in HRI applications, since someone working near or with the robot would feel safer with a smoother sudden manipulator movement than with a fast sudden tool displacement.

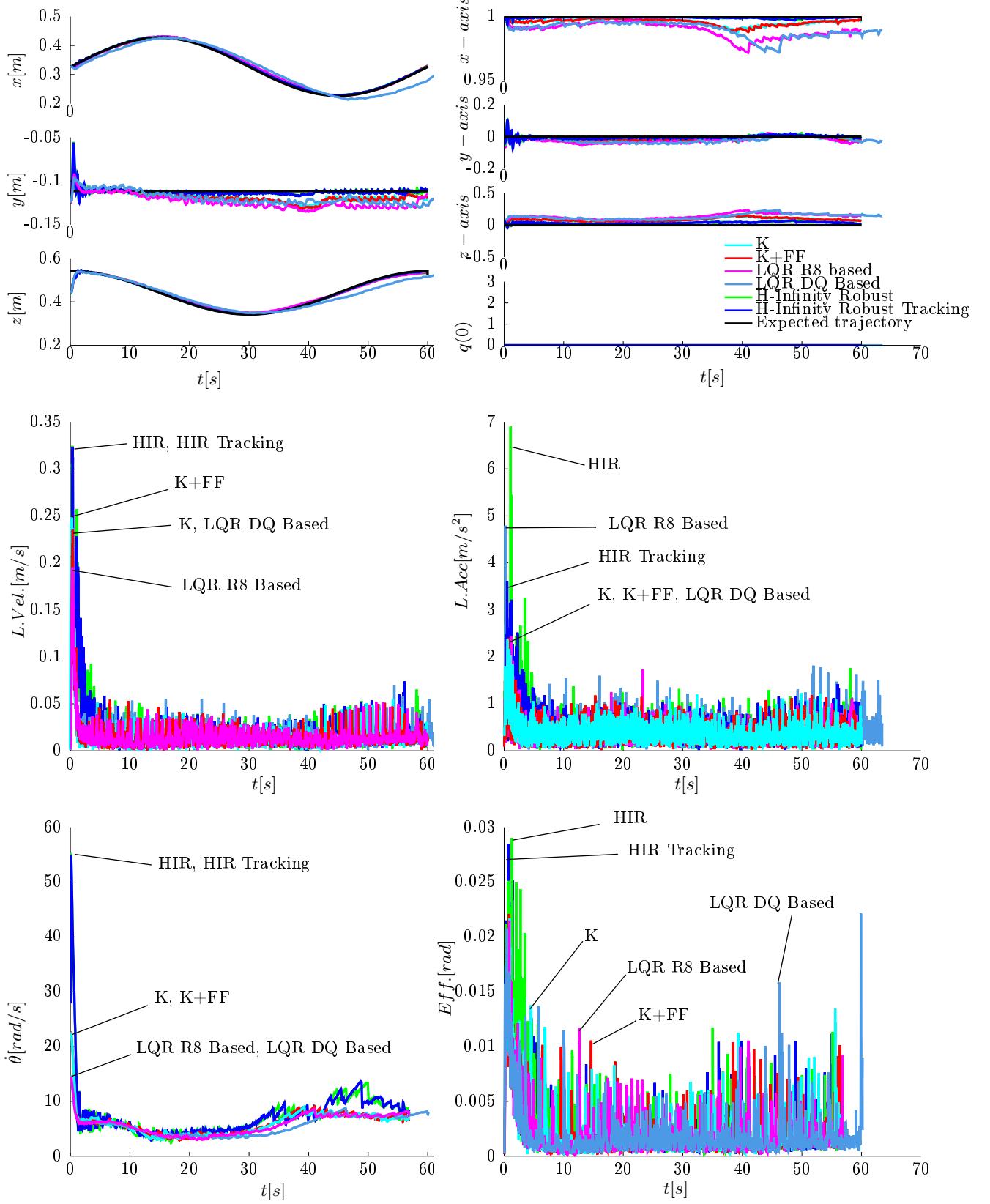


Figure 5.3: Circular trajectory with $q = 1$ and $r = 0.001$.

5.7 Spiral Trajectory

This experiment goal is to compare all controllers and their reliability on the real robot. Table 5.10 presents an overview of the experiment of this section.

Table 5.10: Overview of the experiment to compare all controllers using a spiral trajectory.

Trajectory	Spiral trajectory along the y-axis (Figure 5.1)
Initial joint configuration	$\theta_0 = [0, \pi/3.7, 0, \pi/2, 0, \pi/3.7, 0, 0]$
Trajectory parametrization	$x = r \sin(\theta), y = l, z = r \cos(\theta), \theta \in [0, 6\pi]$
Radius	$r = 7 \text{ cm}$
Height	$l = 7 \text{ cm}$
Sampling time	8 ms
Trajectory points	3000
LQR in \mathbb{R}^8 controller parameters	The q and r parameter vary as in Table 5.11
LQR in DQ controller parameters	$r \in \{0.01\}$ and $q \in \{129.722, 187.15, 270\}$
Proportional controllers parameters	$k = \{60.057, 86.643, 125\}$
H_∞ controllers γ parameters	$\gamma_T = \{0.372, 0.537, 0.775\}$ and $\gamma_O = \{0.377, 0.544, 0.786\}$
H_∞ controllers gain k	Table 5.12 shows the k_T and k_O parameter variation
Executions	50 times for each parameter set

The following trajectory was executed 50 times. The tables with results for all the experiments can be found in Appendix Section E.6. It is a spiral trajectory with constant radius of $r = 7\text{cm}$, length $l = 7\text{cm}$ and robot initial joint configuration given by $\theta_0 = [0, \pi/3.7, 0, \pi/2, 0, \pi/3.7, 0, 0]$ such that

$$\begin{aligned} x &= r \sin(\theta), \theta \in [0, 6\pi], \\ y &= l, \\ z &= r \cos(\theta), \theta \in [0, 6\pi]. \end{aligned} \tag{5.12}$$

This trajectory was chosen because it involves displacements in all directions of the coordinate frames. Furthermore, the gravity force is clearly acting on the end effector while it does the spiral circles. It is then an interesting task to test the performance of all the controllers, specifically, the

H_∞ controller due to its goal of reducing the influence of uncertainties and exogenous disturbances. Therefore, the advantages and disadvantages of each controller are seen.

To exploit the controllers results presented in Section C, the sampling time for this experiment was set to $T_s = 8ms$ after the code review done for the experiment of Section 5.5.

The controllers parameter selection was done by setting the best gains for each controller through an heuristic method. The best gains were searched until there would be no more improvement in the integral norm of the error without an excessive increase in the joints control effort. Except for the LQR controller in \mathbb{R}^8 , all other controllers follow an exponential error decay. In this sense, to check the controllers exponential error decay, the same trajectory was repeated with the best gain multiplied by $\ln 2$ and by $(\ln 2)^2$. Therefore, the error norm should increase by $\frac{1}{\ln 2}$ and $\frac{1}{(\ln 2)^2}$, respectively.

Considering the error in the beginning of the trajectory will be higher as mentioned in section 5.1, some gain variation adjusts were made for the LQR in \mathbb{R}^8 and for the H_∞ controllers. This was done to prevent high joint velocities in the beginning of the trajectory which could lead the system to become unstable. The LQR in \mathbb{R}^8 gain variation is presented in Table 5.11

Table 5.11: LQR in \mathbb{R}^8 controller gain variations

Gain	q	r
best gain	$\begin{cases} q(t) = 220 & t < 0.3t_f \\ q(t) = 270 & t \geq 0.3t_f \end{cases}$	0.01
best gain $\ln 2$	187.15	0.01
best gain $(\ln 2)^2$	129.722	0.01

The H_∞ controller gains were set, in order that the gain matrices \mathbf{K}_O and \mathbf{K}_T initial values at time instant t_0 were lowered and increased along the trajectory until they reach their final value. The gain variation can be seen in Table 5.12

Table 5.12: H_∞ gains variation.

Gain	γ_O	γ_T
k	$k_O(t) = (k_O(t_f) - 2.3) + \frac{t}{t_f}(k_O(t_f) - 2.8)$	$\begin{cases} k_T(t) = (k_T(t_f) - 1.4) + \frac{t}{0.3t_f}1.4 & t \leq 0.3t_f \\ k_T(t) = k_T(t_f) & t > 0.3t_f \end{cases}$
$k \ln 2$	$k_O(t) = k_O(t_f)/2 + \frac{t}{t_f}(\frac{k_O(t_f)}{2})$	$\begin{cases} k_T(t) = (k_T(t_f) - 0.834) + \frac{t}{0.3t_f}0.834 & t \leq 0.3t_f \\ k_T(t) = k_T(t_f) & t > 0.3t_f \end{cases}$
$k (\ln 2)^2$	$k_O(t) = k_O(t_f)/2 + \frac{t}{t_f}(\frac{k_O(t_f)}{2})$	$\begin{cases} k_T(t) = (k_T(t_f) - 1) + \frac{t}{0.3t_f}0.826 & t \leq 0.3t_f \\ k_T(t) = k_T(t_f) & t > 0.3t_f \end{cases}$

In order to evaluate the repeatability and consistency of the controllers the trajectory was executed 50 times for each controller with each gain setting. Tables 5.13-5.15 show the experiment results.

Table 5.13: Trajectory results obtained on Meka performing the spiral end effector trajectory described by (5.12) with all controllers set to $(\ln 2)^2$ of performance: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s 2].

K		K+FF		LQR in \mathbb{R}^8		LQR in DQ		
$k = 60.057$		$k = 60.057$		$q = 129.722, q = 0.01$				
Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	
$Err.$	2.43	0.052	2.43	0.047	1.41	0.032	1.58	0.043
$\dot{\theta}$	316.63	4.752	316.28	4.084	353.42	6.232	357.02	8.067
$Eff.$	0.054	0.0005	0.054	0.0005	0.060	0.001	0.061	0.001
$L.Vel.$	1.327	0.008	1.328	0.008	1.424	0.019	1.443	0.031
$L.Acc.$	120.332	9.517	121.295	9.024	114.235	4.003	110.081	5.172
HIR		HIRT						
$\gamma_T = 0.775, \gamma_O = 0.786$								
Mean	Std. Dev.	Mean	Std. Dev.					
$Err.$	1.84	0.036	1.86	0.039				
$\dot{\theta}$	335.27	4.388	336.17	4.486				
$Eff.$	0.057	0.0004	0.057	0.0004				
$L.Vel.$	1.348	0.007	1.350	0.006				
$L.Acc.$	120.058	6.784	117.563	4.596				

Table 5.14: Trajectory results obtained on Meka performing the spiral end effector trajectory described by (5.12) with all controllers set to $\ln 2$ of performance: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s 2].

K		K+FF		LQR in \mathbb{R}^8		LQR in DQ		
$k = 86.643$		$k = 86.643$		$q = 187.15, r = 0.01$				
Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	
$Err.$	1.68	0.034	1.69	0.038	1.56	0.059	1.29	0.034
$\dot{\theta}$	326.87	3.868	328.02	4.497	460.68	17.689	356.35	9.092
$Eff.$	0.058	0.001	0.058	0.001	0.059	0.001	0.064	0.002
$L.Vel.$	1.378	0.008	1.377	0.007	1.406	0.043	1.480	0.063
$L.Acc.$	125.172	7.887	123.161	6.601	112.552	4.549	110.381	6.262
HIR		HIRT						
$\gamma_T = 0.537, \gamma_O = 0.544$								
Mean	Std. Dev.	Mean	Std. Dev.					
$Err.$	1.24	0.021	1.24	0.024				
$\dot{\theta}$	337.31	2.943	336.14	3.669				
$Eff.$	0.060	0.001	0.060	0.001				
$L.Vel.$	1.388	0.007	1.388	0.006				
$L.Acc.$	122.898	4.731	119.003	2.426				

Table 5.15: Trajectory results obtained on Meka performing the spiral end effector trajectory described by (5.12) with all controllers set to the best performance defined: $Err.$, $\dot{\theta}$ [rad/s], $Eff.$ [rad], $L.Vel$ [m/s], $L.Acc$ [m/s²].

K		K+FF		LQR in \mathbb{R}^8		LQR in DQ		
$k = 125$		$k = 125$		$q = 270, r = 0.01$				
Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	
$Err.$	1.232	0.036	1.232	0.032	1.394	0.039	1.071	0.025
$\dot{\theta}$	342.94	6.154	343.11	5.618	468.73	7.39	353.96	7.170
$Eff.$	0.060	0.001	0.060	0.001	0.061	0.001	0.063	0.002
$L.Vel.$	1.409	0.024	1.404	0.008	1.426	0.025	1.470	0.035
$L.Acc.$	126.561	7.881	125.703	8.762	114.867	3.235	112.349	4.981
HIR		HIRT						
$\gamma_T = 0.372, \gamma_O = 0.377$								
Mean	Std. Dev.	Mean	Std. Dev.					
$Err.$	0.876	0.021	0.876	0.023				
$\dot{\theta}$	346.06	5.630	346.17	6.096				
$Eff.$	0.063	0.001	0.063	0.001				
$L.Vel.$	1.449	0.010	1.443	0.008				
$L.Acc.$	125.601	4.338	123.524	2.816				

As expected, from the point of view of the integral error norm, the performance of the H_∞ controller is the best one for all of the gain sets both for the mean value and the standard deviation. Moreover, it also does this without an excessive increase in the end effector velocity and joints control effort. Therefore, the disturbance rejection with minimum control effort mentioned in Theorems 3.1-3.2 surely contributed to this results.

The LQR controller achieved better results for this parameter selection and smaller sampling time $T_s = 8ms$ in comparison to the experiments of Sections 5.4 and Appendix D. It is important to recall that the LQR in \mathbb{R}^8 controller does not follow an exponential error decay, only the LQR in DQ controller. In this sense, it is more difficult to compare its error with the other controllers. Comparing both LQR controllers, it is seen the LQR in DQ is more predictable and easier to tune gains. The LQR in DQ did not require to vary the q gain along the trajectory and, as expected, its error norm decreased with the increase of q . Lastly, the error norm of the LQR in DQ controller was 0.161 lower than the proportional controllers with an increase of only 0.003 in control effort and of 0.066 in end effector velocities. This shows it was possible to optimize the error norm without an excessive increase in control effort.

The proportional controllers with and without feedforward term behavior was predictable because of their simplicity. As k is increased, the error norm decreases with an increase in the end effector velocity and, consequently, in the joints control effort. Their main advantage is their easy gain setting and their disadvantage is that they do not allow to optimize an specific evaluation attribute. However, they had a worse repeatability and, hence, reliability. This can be seen in

their standard deviation values which are in most cases greater than for the LQR controllers or H_∞ controllers. Regarding the integral error norm, they had higher error norms, hence, lower precision. Nonetheless, it is not possible to generalize that they will always have a worse performance. Depending on the application, they may be the best controller option to choose.

Figure 5.4 shows the graphical results of the best gain selection. The same evaluation for each controller can be done. It is possible to see that both LQR controllers have a longer computational time. Therefore, their trajectory takes longer to complete. Nonetheless, their numerical performance presented along this chapter was done based on their own duration. In this sense, the longer time it takes to complete does not affect the error norm which is calculated online.

Lastly, the error norm increased by a factor of $\frac{1}{\ln 2}$ as the gain was multiplied by $\ln 2$. Therefore the exponential error decay behavior was checked.

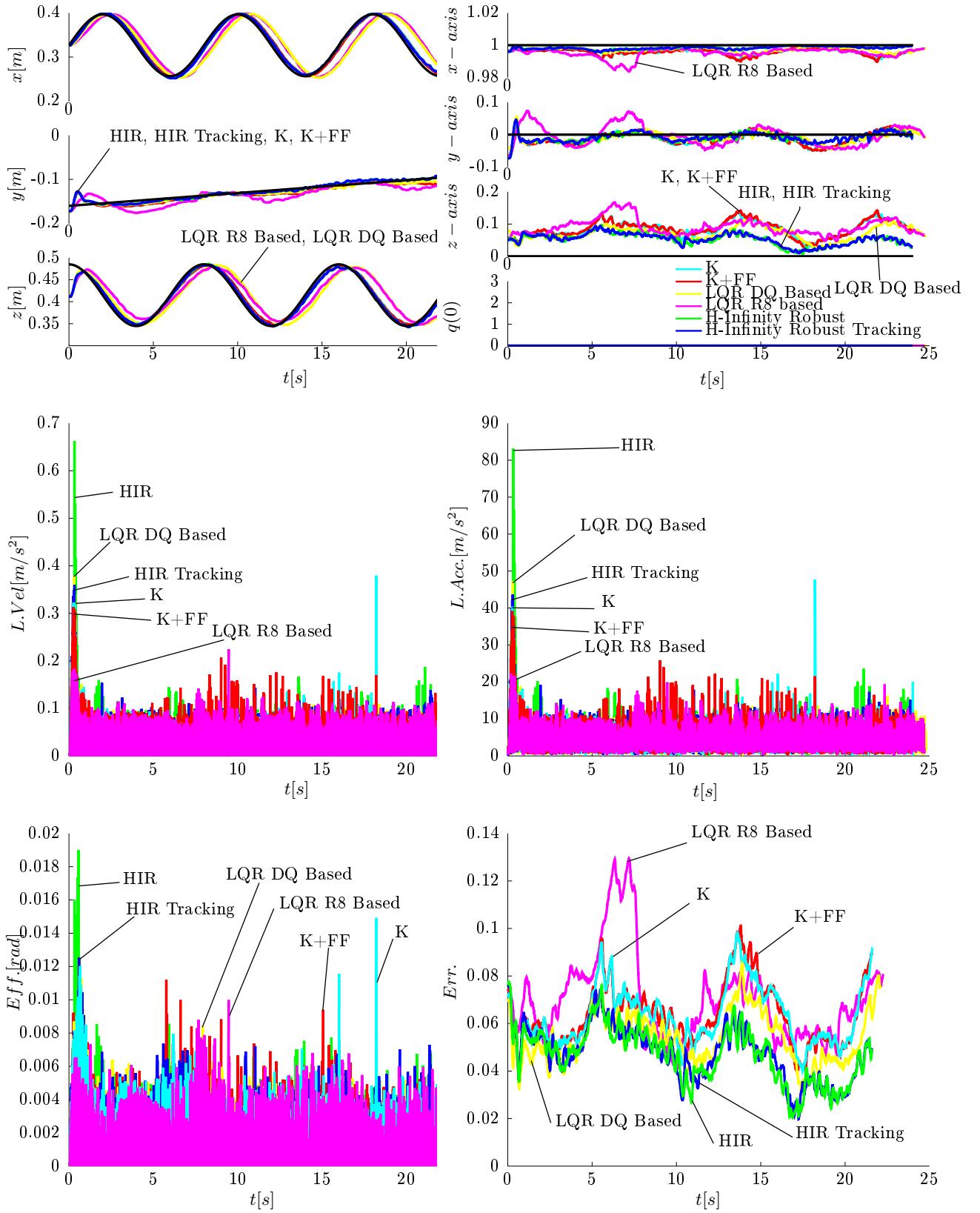


Figure 5.4: Best gain selection: $k = 125$, $q = 270$, $r = 0.01$, $\gamma_T = 0.372$, $\gamma_O = 0.377$

5.8 Sampling Time Histogram

In the four experiments executed to evaluate and compare the dual quaternion based kinematic controllers, two sampling times were chosen. The experiments of Sections 5.4 and 5.6 used a sampling time of 20 ms and the experiments of Sections 5.5 and 5.7 used a sampling time of 8 ms. Table 5.16 gives an overview of the sampling times.

Table 5.16: Overview of experiments sampling time

Experiment	Sampling Time
LQR controllers (Section 5.4)	20 ms
H_∞ controllers (Section 5.5)	8 ms
Circular trajectory (Section 5.6)	20 ms
Spiral trajectory (Section 5.7)	8 ms

The controllers were implemented in C++ as explained in Chapter 4. The sampling time was chosen based on the lowest sampling time which the controllers could execute keeping the system stability. On experiments of Sections 5.5 and 5.7, the LQR and H_∞ controllers could execute in around 20 ms. However, in order to improve the performance, the controllers implementation was reviewed and a sampling time of 8 ms was achieved.

To set the sampling time of the control loop in the robot controlling ROS node (Figure 4.6), the ROS class of the type `ros::Rate`² was used. To measure the trajectory time duration and sampling time achieved, the function `ros::Time::now().toSec()`³ was used.

Although the C++ programs, more specifically, the control loop rate in the robot controlling ROS node (Figure 4.6) can be set, it does not guarantee that the frequency will be kept. The `ros::Rate` is a class which makes a best effort at maintaining a particular rate, or period, for a loop. This effort to keep a stable rate can be seen in Figures 5.5 and 5.6 which shows the sampling time histogram of trajectories with rate periods set to 20 ms and 8 ms, respectively. Nonetheless, the communication rate of the low level modules of the Meka robot which receive and send joint commands to the electronic circuit boards (EC boards) may not be running with such sampling time.

The low level software modules of the Meka Robot use the Real Time Application Interface (RTAI⁴). The C++ program called `shm_humanoid_controller.cpp` given by Meka Robotics is used to get status from the Meka robot sensors using shared memory commands and use RTAI to enable more strict timing constraints. Afterward, they create ROS nodes which publish the robot sensors data to ROS topics. This was explained in Figure 4.6.

In this sense, the results obtained with the controllers on the Meka robot can be further enhanced by exploring the low level software modules. The sampling time achieved on the robot

²http://wiki.ros.org/roscpp/Overview/Time#Sleeping_and_Rates

³<http://wiki.ros.org/roscpp/Overview/Time>

⁴<https://www.rtai.org/>

(Figure 4.6) can be investigated and the choice of the `ros::Rate` used in the robot controlling node can be optimized.

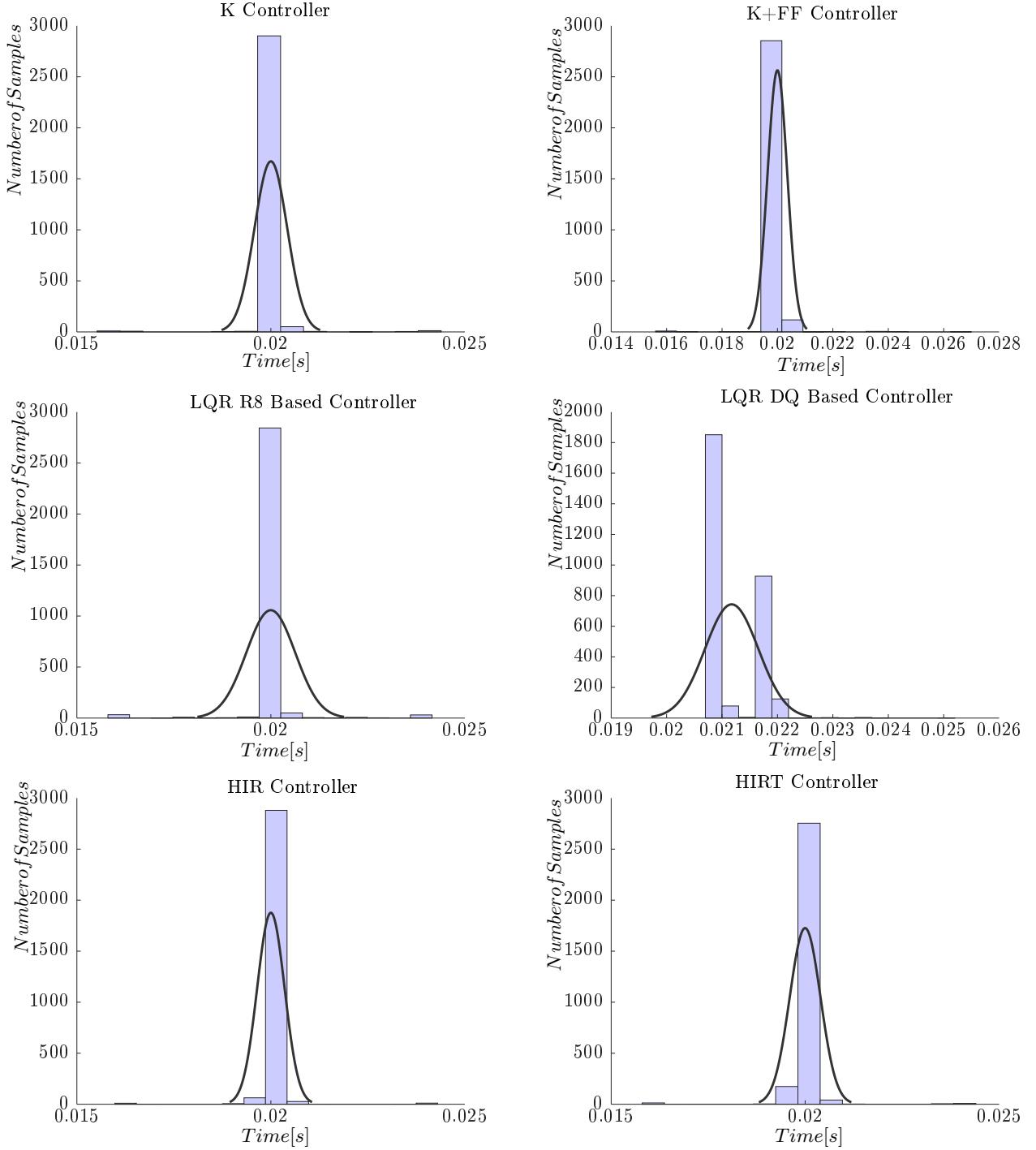


Figure 5.5: Histogram of the time difference between each trajectory step for a sampling time of 20 ms. Mean values (left to right): 0.020, 0.020, 0.020, 0.021, 0.020, 0.020. Std. Dev.: 0.001 (See Appendix E.7 Table 61).

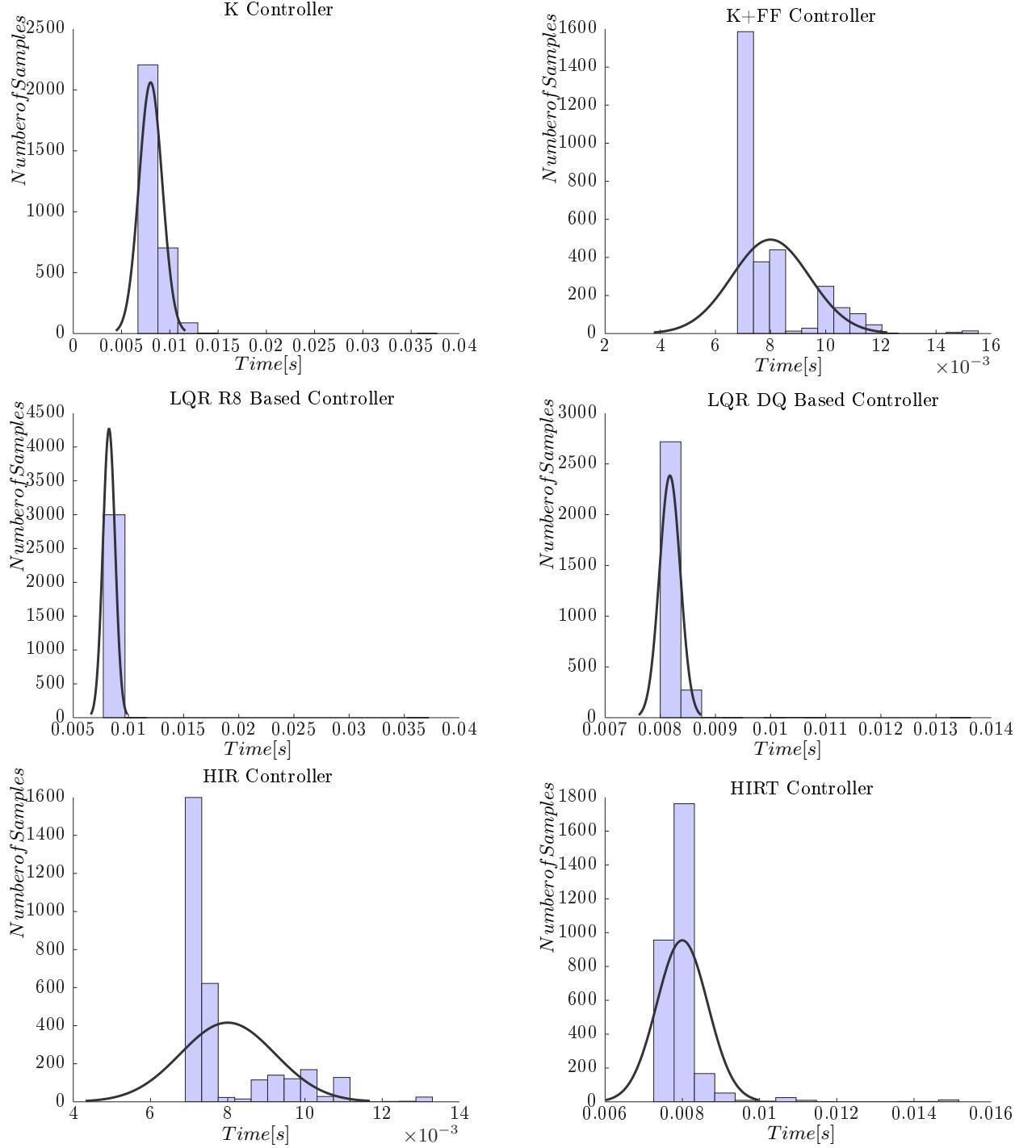


Figure 5.6: Histogram of time difference between each trajectory step for a sampling time of 8 ms.
 Mean values: 0.008, Std. Dev. values: 0.001 (See Appendix E.7 Table 61)

5.9 Trajectory Control Simple Application: Robot Light Painting

The main goal of kinematic control for robot manipulators is to make possible for a robot arm to execute a given task specified in the task-space. As it was mentioned in Section 1.2, one of the contributions of this work should be to deliver a method for controlling a compliant robot

manipulator, more specifically, the Meka Robot which is available at the Robotics and Automation Laboratory (LARA). The dual quaternion based implemented algorithms were applied to draw light paintings using linear and circular trajectories. The results can be seen in Figures 5.7(a)-(f).



Figure 5.7: Kinematic controllers applied to light painting tasks

5.10 Conclusion

First, the procedure for the task definition, the evaluation criteria and the parameter selection was presented. Then, experiments were run to evaluate the implemented dual quaternion based kinematic controllers for trajectory control.

The LQR controllers performance was analyzed. It was shown that the LQR controller had a worse performance in error norm in comparison to the other controllers. Nonetheless, the LQR goal of reaching a trade-off between error norm and control effort varying q and r was achieved. It was also mentioned the earlier experiments with the LQR in \mathbb{R}^8 controller of section D for a similar circular trajectory with less trajectory points. There it was shown that the LQR controller has the potential to achieve the safety-performance trade-off needed for HRI tasks while the K and K+FF controllers do not allow to tune error and joints velocity simultaneously. Even though the LQR controller has a higher trajectory error, for HRI tasks safety is as important as performance. The lower joints control effort obtained with the LQR generated lower torques and forces which score good points in an HRI environment.

In the next experiment, the H_∞ controllers performance was evaluated by investigating the noise to error amplification results when varying the gain parameters. First, the controller parameters were investigated until there would be no more improvement in the integral norm of the error without an excessive increase in the joints control effort. Then the gain matrices parameters were reduced to 80%, 60%, 40% and 20%. As expected, the integral error norm is improved when the gain matrices are increased. In this sense, the dual quaternion based H_∞ controllers proved to be behave well when applied to kinematic control of a compliant robot manipulator.

Afterward, the reliability of all controllers and the exponential error decay were evaluated by running 50 times the same trajectory with three different sets of parameters related by a reason of $\ln 2$.

The H_∞ controllers showed themselves to deliver the best integral error norm performance together with lower joints control effort. This was expected since they are designed with the goal of disturbance rejection and reducing uncertainties with minimum control effort mentioned as stated in Theorems 3.1-3.2. Their disadvantage is the more complicated parameters selection procedure.

The LQR controllers proved to deliver acceptable error norm results without excessive joints control effort and the choice of gain parameters was easier than in the H_∞ controllers. They enable to keep lower control effort and still optimize the error norm performance.

Although the proportional controllers did not deliver the best error norm, the gain adjustment is simple and the error norm improves as k is increased. However, they had a worse repeatability when their standard deviation of each evaluation criteria after 50 repetitions was compared with the results of the LQR and H_∞ controllers.

As the main result of this chapter, the controllers evaluation provided useful information on when to use each of them with the Meka robot. If the robot needs to run lots of distinct trajectories

without adjusting gain parameters and with no need of high precision, the proportional controllers are good choices. When the task trajectory does not change much and error norm must be optimized keeping lower end effector velocity, the LQR controllers should be used. The H_∞ robust controllers are for trajectories with high performance needs. Table 5.17 shows an overview of all controllers with their performance advantages and disadvantages.

Table 5.17: Overview of controllers evaluation.

Controller	Invariant Err. Norm	Joints control effort and end effector velocities	Parameter selection
Proportional with/without Feedforward term	Good for different non-high precision tasks. Lower repeatability in comparisson with the other controllers.	Do not take them into account. For lower error, deliver high velocities.	Easy. Changes only with changes in sample time.
LQR	Improved error. Good for similar tasks without variation.	No excessive increase in control effort and velocities. Enable to adjust both with q and r .	Parameters may change for each trajectory.
H_∞	Best error performance.	Disturbances rejection with minimum control effort. Deliver high velocities which are feasible.	Parameter may change for different trajectory and vary along each trajectory.

The controllers were evaluated considering the Meka robot low level controllers work as expected. However, the minimum sampling time achieved by the robot controlling node and the low level controllers may differ. In this sense, the sampling time histograms of the robot controlling node were presented. From the point of view of the control module, the rate was kept around the desired value. However, the performance of the dual quaternion based kinematic controllers can be enhanced if the performance of the low level modules of the Meka robot are exploited.

Lastly, the LQR controller in DQ was used to execute task trajectories of drawings. The results were presented using light painting. Long exposure photos were taken in order to visualize the robot running the trajectories.

Chapter 6

Conclusion and Future Works

This work evaluated six different recently developed task-space dual quaternion based kinematic controllers on the compliant anthropomorphic manipulator A2 Arm from Meka Robotics with seven degrees of freedom. This project was motivated by the challenges posed in the field of human-robot interaction (HRI) related to control a compliant robotic arm which is safe to work alongside humans. A review of the HRI field was done to better understand the safety issues involved. The mathematical background on unit dual quaternion representation of rigid body motion and the dual quaternion algebra applied on the design of kinematic controllers was also presented.

To evaluate the controllers, several trajectory tasks and evaluation criteria were devised. Trajectories which lies within the end-effector reachable space were chosen as to avoid joint limits and boundary singularities — which are out of the scope of the present manuscript. Considering the HRI safety requirements, the end effector linear velocities and acceleration, the joints control effort and the control signals were analyzed. Taking into account the dual quaternion spatial difference, the invariant error norm was used to check the controller's error performance.

It was obtained that simple proportional kinematic controllers with and without feedforward term showed acceptable error norm results. Furthermore, the parameter selection is straightforward. As the proportional gain is increased, the error norm decreases. However, it does not take end effector velocities and acceleration into account. Hence, if the gain is too high, the controller may command velocities which are not feasible by the robot joints and lead to chattering ad local instability performance. Moreover, in the HRI context, high-velocity peaks are not desired for the end effector because they cause discomfort on the humans working near him.

The high-velocity peaks motivated the use of two dual quaternion based linear quadratic optimal controllers (LQR controllers). Their parameters allow weighting the control effort in terms of the end effector velocity and the error cost along the trajectory. As a consequence, results were obtained in which the error norm improved in comparison to the proportional controllers without an excessive increase in the end effector velocity and in the joints control effort. However, exogenous disturbances and uncertainties influenced the LQR controller. Even though the parameters were well tuned, the performance varied too much for distinct trajectories with a different number of points and sampling times. It was perceived a constant need to tune parameters.

In order to reduce the robot sensitivity to disturbances and uncertainties, two dual quaternions based H_∞ robust controllers were tested. Experiments checked their noise to error amplification properties worked on the real robot. Then, experiments showed they had the best improvement in the error norm in comparison to the other controllers. Moreover, they behaved well under different trajectories and sampling times. On the other hand, to keep their performance with lower control frequencies, they deliver high end effector velocities and cause oscillations in the manipulator. To solve this problem, their gains were set lower in the beginning of the trajectory and linearly increased with time. Hence, their disadvantage is the need for gain tuning along the trajectory.

Considering the Meka robot system use ROS, the kinematic controller module was implemented in a C++ ROS node and used the DQ Robotics library. Later, this will ease the integration of force and vision sensors with ROS support. Stability problems in the controllers due to numerical conditioning, numerical drifting, and unexpected issues were checked with the V-REP simulator to prevent damages on the real robot. The Kuka LBR iiwa 7 R800 model was used since the fully working Meka robot model was not yet available. Therefore, the performance of the controllers in the simulation and on the real robot could not be compared one-to-one.

Hence, the contribution of this work is the sum of different aspects. An environment to implement and test dual quaternion based kinematic controllers on a compliant robot manipulator was developed. It allows it to execute end effector trajectories defined in the task-space. Moreover, each controller's evaluation provided useful information on when to use each of them with the Meka robot. If the robot needs to run lots of distinct trajectories without adjusting gain parameters and with no need of high precision, the proportional controllers are good choices. When the task trajectory does not change much and error norm must be optimized keeping lower end effector velocity, the LQR controllers should be used. The H_∞ robust controllers are for trajectories with high-performance needs. Lastly, in order to use the robot with the new controllers in a practical application, simple trajectory drawings were generated and the Meka robot drew them as light paintings captured by long exposure photos.

In future works, the Meka robot force sensor information could be used to implement force control in order to get a better gravity compensation. Due to the compliant characteristics of the Meka robot, the gravity, and other exogenous forces strongly influence the manipulator behavior although the low level controllers provided by Meka Robotics already do the gravity compensation. Moreover, the Meka low level controllers achievable sampling times may be investigated and the choice of the task-space controllers sampling time may be optimized. Solving this issues may enable the robot to execute assembling and writing tasks and also give force feedback to haptic interface devices. Regarding the kinematic controllers implementation, most of the controllers were implemented as a DQ Robotics controller class which must be able to do both setpoint and trajectory tracking control. However, the LQR controllers were designed for trajectory tracking and not for setpoint control. This issue was explained at the end of Chapter 4. In future works, the LQR controllers could be adapted for setpoint control and implemented as a DQ Robotics controller.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MILLER, D. P.; NOURBAKHSH, I. R.; SIEGWART, R. Robots for Education. In: *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 1283–1301. Disponível em: <http://link.springer.com/10.1007/978-3-540-30301-5_56>.
- [2] BICCHI, A.; PESHKIN, M. A.; COLGATE, J. E. Safety for Physical Human-Robot Interaction. In: *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 1335–1348. Disponível em: <http://link.springer.com/10.1007/978-3-540-30301-5_58>.
- [3] HÄGELE, M.; NILSSON, K.; PIRES, J. N. Industrial Robotics. In: *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 963–986. Disponível em: <http://link.springer.com/10.1007/978-3-540-30301-5_43>.
- [4] ZELDMAN, M. I. *What every engineer should know about robots*. [S.l.]: M. Dekker, 1984. 197 p. ISBN 9780824771232.
- [5] NOF, S. Y. *Handbook of industrial robotics*. [S.l.]: J. Wiley, 1985. 1358 p. ISBN 0471896845.
- [6] DHILLON, B. S.; FASHANDI, A. R. M.; LIU, K. L. Robot systems reliability and safety: a review. *Journal of Quality in Maintenance Engineering*, v. 8, n. 3, p. 170–212, 2002. Disponível em: <<http://dx.doi.org/10.1108/13552510210439784>>.
- [7] AHMED, M. R. *Compliance Control of Robot Manipulator for Safe Physical Human Robot Interaction*. 122 p. Tese (Doutorado) — Örebro University, School of Science and Technology, 2011.
- [8] BROGLIATO, B.; LOZANO, R. Correction to Adaptive control of robot manipulators with flexible joint. *IEEE Transactions on Automatic Control*, IEEE, v. 41, n. 6, p. 920–922, jun 1996. ISSN 0018-9286. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=506251>>.
- [9] LUCA, A. D. Feedforward/feedback laws for the control of flexible robots. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*. IEEE, 2000. v. 1, p. 233–240. ISBN 0-7803-5886-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=844064>>.
- [10] PRATT, G.; WILLIAMSON, M. Series elastic actuators. In: *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Coop-*

rative Robots. IEEE Comput. Soc. Press, 1995. v. 1, p. 399–406. ISBN 0-8186-7108-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=525827>>.

- [11] SHARON, A.; HOGAN, N.; HARDT, D. High bandwidth force regulation and inertia reduction using a macro/micro manipulator system. In: *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*. IEEE Comput. Soc. Press, 1988. p. 126 — 132 vol.1. ISBN 0818608528 (ISBN). Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=12036>>.
- [12] MORITA, T.; SUGANO, S. Design and development of a new robot joint using a mechanical impedance adjuster. In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation*. IEEE, 1995. v. 3, p. 2469–2475. ISBN 0-7803-1965-6. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=525630>>.
- [13] BICCHI, A.; TONIETTI, G. Fast and Soft-Arm Tactics. *IEEE Robotics & Automation Magazine*, IEEE, v. 11, n. 2, p. 22–33, jun 2004. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1310939>>.
- [14] FIGUEREDO, L. F. C. *Kinematic control based on dual quaternion algebra and its application to robot manipulators*. Tese (Doutorado) — Universidade de Brasília, 2016.
- [15] ADORNO, B. V. *Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration*. Tese (Doutorado) — Université Montpellier 2, 2011. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.393.2333>>.
- [16] SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. *Robot Modeling and Control*. [s.n.], 2005. Disponível em: <<http://www.wiley.com/WileyCDA/WileyTitle/productCd-EHEP000518.html>>.
- [17] SICILIANO, B. et al. *Robotics - Modelling, Planning and Control / Bruno Siciliano / Springer*. [s.n.], 2009. Disponível em: <<http://www.springer.com/br/book/9781846286414>>.
- [18] MARINHO, M. M.; FIGUEREDO, L. F. C.; ADORNO, B. V. A dual quaternion linear-quadratic optimal controller for trajectory tracking. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015. p. 4047–4052. ISBN 978-1-4799-9994-1. Disponível em: <<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=7353948>>.
- [19] LARA, R. A. L. *Meka Robot Wiki at LARA*. 2015. <https://github.com/lara-unb/Meka/wiki>. [acessado em 15/Jan/2016].
- [20] KUIPERS, J. B. *Quaternions and rotation sequences : a primer with applications to orbits, aerospace, and virtual reality*. [S.l.]: Princeton University Press, 1999. 371 p. ISBN 9780691102986.
- [21] MARINHO, M. M. Robot-aided Endoscope Control Under Laparoscopic Surgery Constraints Using Dual Quaternions. 2014.

- [22] ADORNO, B. V.; BÓ, A. P. L.; FRAISSE, P. Kinematic modeling and control for human-robot cooperation considering different interaction roles. *Robotica*, Cambridge University Press, v. 33, n. 02, p. 314–331, feb 2014. ISSN 0263-5747. Disponível em: <http://journals.cambridge.org/abstract_S0263574714000356>.
- [23] WHITNEY, D. Resolved Motion Rate Control of Manipulators and Human Prostheses. *IEEE Transactions on Man Machine Systems*, IEEE, v. 10, n. 2, p. 47–53, jun 1969. ISSN 0536-1540. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4081862>>.
- [24] WAMPLER, C. Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods. *IEEE Transactions on Systems, Man, and Cybernetics*, IEEE, v. 16, n. 1, p. 93–101, jan 1986. ISSN 0018-9472. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4075580>>.
- [25] DOMBRE, E.; KHALIL, W. *Wiley: Robot Manipulators: Modeling, Performance Analysis and Control - Etienne Dombre, Wisama Khalil*. [s.n.], 2007. Disponível em: <<http://www.wiley.com/WileyCDA/WileyTitle/productCd-190520910X.html>>.
- [26] PHAM, H.-L. et al. Position and orientation control of robot manipulators using dual quaternion feedback. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010. p. 658–663. ISBN 978-1-4244-6674-0. ISSN 2153-0858. Disponível em: <<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5651097>>.
- [27] FIGUEREDO, L. F. C. et al. Robust kinematic control of manipulator robots using dual quaternion representation. In: *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013. p. 1949–1955. ISBN 978-1-4673-5643-5. ISSN 1050-4729. Disponível em: <<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6630836>>.
- [28] MACIEJEWSKI, A. A.; KLEIN, C. A. Numerical filtering for the operation of robotic manipulators through kinematically singular configurations. *Journal of Robotic Systems*, v. 5, n. 6, p. 527–552, dec 1988. ISSN 07412223. Disponível em: <<http://doi.wiley.com/10.1002/rob.4620050603>>.
- [29] PEREIRA, M. S.; FIGUEREDO, L. F. C.; BERNARDES, M. C. Evaluation of Dual Quaternion Based Controllers on Anthropomorphic Compliant Manipulators. In: *XXI Congresso Brasileiro de Automatica*. Vitória/ES: [s.n.], 2016. Disponível em: <http://www.swge.inf.br/PDF/CBA2016-0861_048126.PDF>.
- [30] SCHERER, C. *The Riccati Inequality and State-Space H-Infinity -Optimal Control*. Tese (Doutorado) — University of Würzburg, Germany, 1990.
- [31] LOCATELLI, A. *Optimal Control - An Introduction*. [s.n.], 2001. Disponível em: <<http://www.springer.com/us/book/9783764364083>>.
- [32] GARBER, V. Optimum intercept laws for accelerating targets. *AIAA Journal*, v. 6, n. 11, p. 2196–2198, nov 1968. ISSN 0001-1452. Disponível em: <<http://arc.aiaa.org/doi/abs/10.2514/3.4962>>.

- [33] SACHKOV, Y. L. Control Theory on Lie Groups. *Journal of Mathematical Sciences*, v. 156, n. 3, p. 381–439, 2009.
- [34] ROHMER, E.; SINGH, S. P. N.; FREESE, M. V-REP: A versatile and scalable robot simulation framework. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013. p. 1321–1326. ISBN 978-1-4673-6358-7. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6696520>>.
- [35] KANEHIRO, F. et al. Open architecture humanoid robotics platform. In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*. IEEE, 2002. v. 1, p. 24–30. ISBN 0-7803-7272-7. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1013334>>.
- [36] KOENIG, N.; HOWARD, A. *Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator*.
- [37] MICHEL, O. Webots TM : Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems*, v. 1, p. 39–42, 2004.
- [38] MEKA ROBOTICS. *Overview*. 2013. mekabot.com/wiki/doku.php?id=user_guides:arm_a2r4. [acessado em 16/Oct/2013].
- [39] MEKA ROBOTICS. *Understanding M3 Controllers*. 2013. mekabot.com/wiki/doku.php?id=m3:sidebar:understanding_m3_controllers. [acessado em 16/Oct/2013].
- [40] QUIGLEY, M. et al. ROS: an open-source Robot Operating System. *ICRA workshop on open source system*, v. 3, n. 3.2, p. 5, 2009. Disponível em: <<http://www.willowgarage.com/papers/ros-open-source-robot-operating-system>>.

ANEXOS

A Rigid Motions and Homogeneous Transformations

A.1 Representing Positions

The Euclidean space can be used to represent points and vectors in multiple coordinate frames. Positions may be geometrically described using the synthetic approach which works directly with geometric entities (e.g., points or lines). For example, one can say that x_0 and y_0 are perpendicular and that the cross product $\mathbf{v}_1 \times \mathbf{v}_2$ defines a vector perpendicular to the plane containing \mathbf{v}_1 and \mathbf{v}_2 (Figure 1).

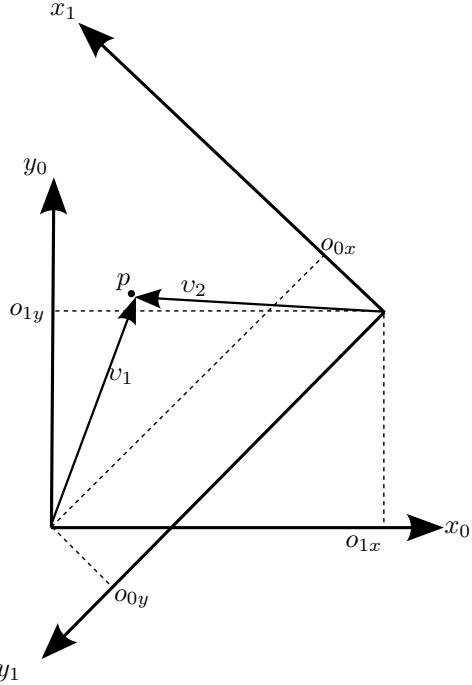


Figure 1: Two coordinate frames, a point p and vectors \mathbf{v}_1 and \mathbf{v}_2

The second approach is analytic and represents positions using coordinates and equations. Robot manipulator tasks are often defined using Cartesian coordinates. It is possible to describe the trajectory of the end effector of a robot in relation to a fixed coordinate system, called the world (w) or base frame to which all objects including the manipulator are referenced (Figure 2). Therefore, the analytic reasoning is usually chosen [16].

From a geometrical point of view, a point p corresponds to a specific location in space. In the analytic approach, its position is represented by a coordinate vector \mathbf{p}^0 with respect to a coordinate frame $o_0x_0y_0$ or in the case of a robot, the world or base frame. To clearly denote the reference frame, the superscript notation will be adopted.

In robotics, it is often needed to represent the positions of objects in space with respect to different coordinate systems. The coordinate system origin is just a point in space, then the position of the origin of one coordinate system with respect to another may be assigned. In Figure 1, we have $\mathbf{o}_1^0 = [o_{1x} \; o_{1y}]^T$ and $\mathbf{o}_0^1 = [o_{0x} \; o_{0y}]^T$.

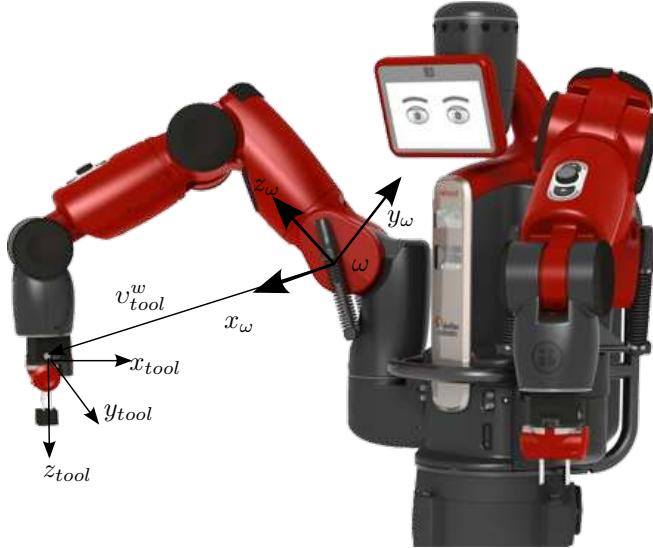


Figure 2: Baxter robot with world coordinate frame w (source: Rethink Robotics).

In comparison to a point, a vector specifies a direction and a magnitude. They can be used to represent displacements or forces. The displacement of the origin o_0 to the point p is given by the vector \mathbf{v}_1 . Along the text, the term vector will be used to refer to what are also called free vectors which are not constrained to be located at a particular point in space. Under this convention, it is clear that points and vectors are not equivalent because points refer to a specific location in space and a vector can be moved to any location in space. Then, two vectors are equal if they have the same direction and magnitude [16]. The assignment of coordinates to vectors use the same notation convention of points. They depend directly on the choice of reference coordinate frame and to perform algebraic manipulations all vectors must be defined with respect to the same coordinate frame. In Figure 3, they would be $\mathbf{v}_1^0 = [v_{1x}^0 \ v_{1y}^0]^T$, $\mathbf{v}_1^1 = [v_{1x}^1 \ v_{1y}^1]^T$, $\mathbf{v}_2^0 = [v_{2x}^0 \ v_{2y}^0]^T$, $\mathbf{v}_2^1 = [v_{2x}^1 \ v_{2y}^1]^T$.

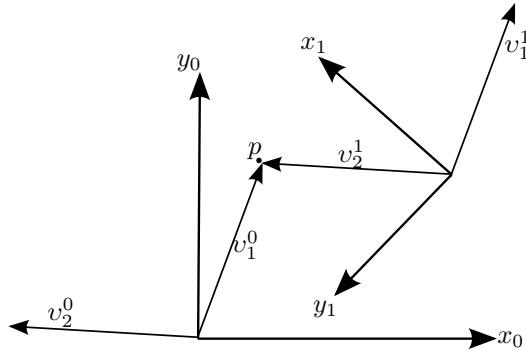
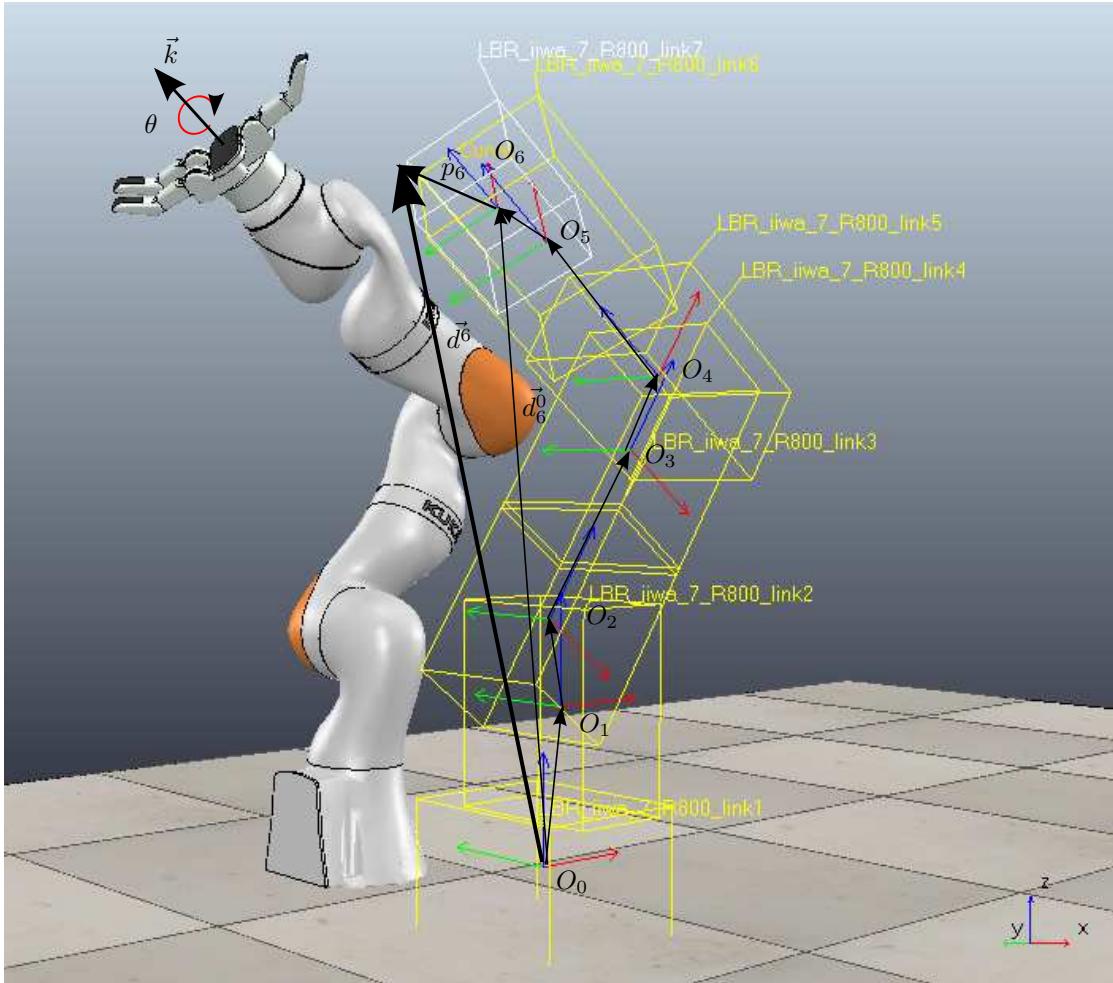


Figure 3: Two coordinate frames, a point p and vectors $\mathbf{v}_1^0, \mathbf{v}_2^0, \mathbf{v}_1^1, \mathbf{v}_2^1$.

Therefore, a representation system needs to allow points to be represented with respect to various coordinate systems and to allow to transform the coordinates of points between coordinate systems.

Points and vectors are useful to describe the position of the origin of one frame with respect to another. However, it is also needed to represent the orientation of one coordinate frame relative to another frame. This problem appears, for example, when we have the robot manipulator end effector rotated with respect to the world or base coordinate frame (Figure 2). Another situation occurs when we describe the robot joints coordinate frames which are usually rotated or translated. In Figure 4, it is possible to see that each joint of a robot has its own coordinate frame. They are translated and rotated in relation to each other. The Figure 4 will be used to help visualize the concepts of translations, rotations and rigid body motions.



V-REP Simulator¹

Figure 4: Kuka LBR iiwa 7 on simulation environment.

In Figure 4, each joint has its own coordinate frame. The vectors in black represent the translation between \mathcal{O}_0 , \mathcal{O}_1 , ..., \mathcal{O}_6 . The red, green and blue vectors represents the x , y and z axes, respectively. The vector \mathbf{k} represents an arbitrary axis in space. In order to have a clearer figure, all the coordinate frames and vectors were not shown directly on the robot, except \mathbf{k} , which will be used to represent the robot gripper orientation.

¹The V-REP simulator is introduced in the Chapter 4. This figure was adapted from the V-REP EDUCATIONAL version: <http://www.coppeliarobotics.com/>

A.2 Representing Rotations

It is possible to specify the coordinate vectors for the axes of frame $o_1x_1y_1$ with respect to coordinate frame $o_0x_0y_0$ as

$$R_1^0 = [x_1^0 | y_1^0]$$

where \mathbf{x}_1^0 and \mathbf{y}_1^0 are the coordinates in frame $o_0x_0y_0$ of unit vectors \mathbf{x}_1 and \mathbf{y}_1 , respectively. \mathbf{R}_1^0 is the rotation matrix. The notation convention of allowing the superscript to denote the reference frame is also used. The rotation matrix column vectors are the coordinates of the (unit vectors along the) axes of one frame expressed relative to the frame of reference expressed by the superscript.

It is possible to build the rotation matrix by projecting the axes of the one frame onto the coordinate axes of the reference frame of the superscript. This is also scalable to the three dimensional case. Considering that the dot product of two unit vectors gives the projection of one onto the other, we have for the three dimensional case

$$R_1^0 = \begin{bmatrix} \mathbf{x}_1 \cdot \mathbf{x}_0 & \mathbf{y}_1 \cdot \mathbf{x}_0 & \mathbf{z}_1 \cdot \mathbf{x}_0 \\ \mathbf{x}_1 \cdot \mathbf{y}_0 & \mathbf{y}_1 \cdot \mathbf{y}_0 & \mathbf{z}_1 \cdot \mathbf{y}_0 \\ \mathbf{x}_1 \cdot \mathbf{z}_0 & \mathbf{y}_1 \cdot \mathbf{z}_0 & \mathbf{z}_1 \cdot \mathbf{z}_0 \end{bmatrix} \quad (\text{A.1})$$

and

$$R_0^1 = \begin{bmatrix} \mathbf{x}_0 \cdot \mathbf{x}_1 & \mathbf{y}_0 \cdot \mathbf{x}_1 & \mathbf{z}_0 \cdot \mathbf{x}_1 \\ \mathbf{x}_0 \cdot \mathbf{y}_1 & \mathbf{y}_0 \cdot \mathbf{y}_1 & \mathbf{z}_0 \cdot \mathbf{y}_1 \\ \mathbf{x}_0 \cdot \mathbf{z}_1 & \mathbf{y}_0 \cdot \mathbf{z}_1 & \mathbf{z}_0 \cdot \mathbf{z}_1 \end{bmatrix}.$$

The inner product is commutative, (i.e. $\mathbf{x}_i \cdot \mathbf{y}_j = \mathbf{y}_j \cdot \mathbf{x}_i$), then $R_0^1 = (R_0^1)^T$. And remembering that coordinate axes are always mutually orthogonal, it is seen that $(R_1^0)^T = (R_1^0)^{-1}$. Moreover, the column vectors of R_1^0 are of unit length and mutually orthogonal which means the rotation matrix is orthogonal. It is possible to show that $\det R_1^0 = \pm 1$ and restricting to right-handed coordinate systems, then $\det R_1^0 = +1$. The set of all 3×3 matrices which have this properties are parts of the Special Orthogonal group of order 3 (SO(3)) [16, 17].

The basic rotation matrices which describe the rotation of a frame through an angle θ about the x , y and z -axis are derived from (A.1) and represented as

$$R_{z,\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A.2})$$

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (\text{A.3})$$

$$R_{y,\theta} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}. \quad (\text{A.4})$$

A rotation matrix $R \in SO(3)$ has three equivalent geometrical meanings [17, 16]:

1. It represents a coordinate transformation relating the coordinates of a point \mathbf{p} in two different frames:

$$\mathbf{p}_b^0 = R_1^0 \mathbf{p}_b^1. \quad (\text{A.5})$$

2. It gives the coordinate transformation between the coordinates of a point expressed in two different frames:

$$R_1^0.$$

3. It is an operator which allows the rotation of a vector in the same coordinate frame:

$$\mathbf{v}_1^0 = R_1^0 \mathbf{v}^1.$$

The (A.5) describes a rotational transformation between two frames. It is possible to do more than one rotation and add new coordinate frames. A point \mathbf{p} can then be represented by coordinates specified with respect to any of the frames. The order in which a sequence of rotations and the matrices multiplications are done affects the result. While positions are a vector quantity, rotational transformation do not commute.

Composition of rotational transformations have the following rule. Consider a fixed frame $o_0x_0y_0z_0$, a current frame $o_1x_1y_1z_1$ and a third frame $o_2x_2y_2z_2$, the rotation matrix R_1^0 , that is, the relation between 1 and 2, is given by rotation $R = R_2^1$ done to the current frame. To obtain the rotation matrix between 0 and 2, then post-multiply R_1^0 by R

$$R_2^0 = R_1^0 R_2^1. \quad (\text{A.6})$$

For the rotation done relative to the fixed frame, pre-multiply R_1^0 by R to obtain

$$R_2^0 = R R_1^0. \quad (\text{A.7})$$

In Figure 4, it is possible to obtain the rotation matrix between each joint coordinate frame: $R_1^0, R_2^1, \dots, R_6^5$. Each of them is easily obtained using robot forward kinematics [16, p. 68]. Then, if, for instance, it is desired to know the rotation between the base coordinate frame and the end effector:

$$R_6^0 = R_1^0 R_2^1 R_3^2 R_4^3 R_5^4 R_6^5 = R_6^0. \quad (\text{A.8})$$

Regarding the parametrizations of rotations, a general rotational transformation has nine elements r_{ij} which are not independent quantities, but related by six constraints due to the orthogonality conditions. A rigid body has three rotational degrees-of-freedom (DOF). Therefore, three quantities are needed to describe its orientation and this representation is called minimal representation [17]. This can be seen by inspecting the properties of the $SO(3)$ matrices:

$$\sum_i r_{ij}^2 = 1, j \in \{1, 2, 3\} \quad (\text{A.9})$$

$$r_{1i}r_{1j} + r_{2i}r_{2j} + r_{3i}r_{3j} = 0, i \neq j. \quad (\text{A.10})$$

Equations (A.9) and (A.10) are based on the fact that the columns of a rotation matrix are unit vectors and are also mutually orthogonal.

A.3 Parametrization of Rotations

One arbitrary rotation can be parametrized using only three independent quantities. A minimal representation of orientation can then be obtained by using a set of three angles (ϕ, θ, ψ) . A generic rotation matrix can be built by doing three elementary rotations and guaranteeing that two successive rotations are not made about parallel axes. Therefore, 12 different combinations of the angles are allowed of all 27 possible sets. Each set is known as triplet of Euler Angles. The more common Euler angles used are the ZYZ angles and the ZYX (or roll-pitch-yaw) angles [17] which are both minimal representations.

The ZYZ angles yields the following rotation matrix and considering the notation adopted¹

$$R_{ZYZ} = R_{z,\phi} R_{y,\theta} R_{z,\psi} = \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\phi & s_\theta s_\psi & c_\theta \end{bmatrix} = R \quad (\text{A.11})$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \quad (\text{A.12})$$

The more difficult problem to tackle is to determine a set of Euler angles ϕ, θ and ψ in order that $R = R_{zyz}$. It is shown there are infinitely many solutions [16, 17]. Suppose that not both of r_{13} and r_{23} are zero. Hence from equation A.11, $s_\theta \neq 0$, and thus not both r_{31}, r_{32} are zero. If not both r_{13} and r_{23} are zero, follows that $r_{33} \neq \pm 1$, and $c_\theta = r_{33}$, $s_\theta = \pm\sqrt{1 - r_{33}^2}$. Hence,

$$\theta = \text{atan2}(r_{33}, \pm\sqrt{1 - r_{33}^2}). \quad (\text{A.13})$$

If the positive square root in (A.13) is chosen, then $s_\theta > 0$ and

$$\phi = \text{atan2}(r_{13}, r_{23}) \quad (\text{A.14})$$

$$\psi = \text{atan2}(-r_{31}, r_{32}). \quad (\text{A.15})$$

If the choice is the negative square root in (A.13), then $s_\theta < 0$ and

$$\phi = \text{atan2}(-r_{13}, -r_{23}) \quad (\text{A.16})$$

$$\psi = \text{atan2}(r_{31}, -r_{32}). \quad (\text{A.17})$$

Assuming that R has the form

$$R = \begin{bmatrix} r_{11} & r_{12} & 0 \\ r_{21} & r_{22} & 0 \\ 0 & 0 & \pm 1 \end{bmatrix}, \quad (\text{A.18})$$

if $r_{33} = 1$, from (A.11), it is seen that

$$\phi + \psi = \text{atan2}(r_{11}, r_{21}) = \text{atan2}(r_{11}, -r_{12}). \quad (\text{A.19})$$

¹The notations c_x and s_x means $c_x = \cos(x)$ and $s_x = \sin(x)$ respectively.

The sum $\phi + \theta$ is the only result which can be determined in this case, hence, there are infinitely many solutions. It is assumed $\phi = 0$ by convention. If $r_{33} = -1$, then it is yielded by (A.11) that

$$\phi + \psi = \text{atan2}(-r_{11}, -r_{12}) \quad (\text{A.20})$$

which also have infinitely many solutions. A similar conclusion is obtained for the roll-pitch-yaw angles [17, 16].

In robotic manipulators applications it is, however, interesting to use representations which can solve the problem of trajectory planning for the manipulator's end effector orientation. Furthermore, rotations are not always performed about the principal coordinate axis. Sometimes it is desired to rotate about an arbitrary axis in space. For example, in Figure 4, the robot end effector orientation is given by \vec{k} .

In order to simplify calculation, assume that \mathcal{O}_6 in Figure 4, is now \mathcal{O}_0 . A nonminimal representation can be obtained using four parameters indicating a rotation of a given angle about an axis in space. Assume $\mathbf{k} = (k_x, k_y, k_z)^T$, expressed in the frame $o_0x_0y_0z_0$, is a unit vector defining an axis. It is of interest to obtain the rotation matrix $R_{\mathbf{k},\theta}$ representing a rotation of θ about \mathbf{k} . $R_{\mathbf{k},\theta}$ is derived after some calculation [16, 17] as

$$R_{\mathbf{k},\theta} = \begin{bmatrix} k_x^2 v_\theta + c_\theta & k_x k_y v_\theta - k_z s_\theta & k_x k_z v_\theta + k_y s_\theta \\ k_x k_y v_\theta + k_z s_\theta & k_y^2 v_\theta + c_\theta & k_y k_z v_\theta - k_x s_\theta \\ k_x k_z v_\theta - k_y s_\theta & k_y k_z v_\theta + k_x s_\theta & k_z^2 v_\theta + c_\theta \end{bmatrix} = R, \quad (\text{A.21})$$

where $v_\theta = \text{vers } \theta = 1 - c_\theta$. With $R_{\mathbf{k},\theta}$ and considering that it was obtained considering \mathcal{O}_6 as \mathcal{O}_0 , it is possible to obtain the end effector rotation in relation to each other joint coordinate frame. Therefore, the set of rotations matrix grows to: $R_1^0, R_1^2, \dots, R_5^6$ and $R_6^{\mathbf{k}}$. Now, it is feasible to describe any rotation between each combination of the robot frames as in (A.8), but also including the robot end effector which may be rotated about a arbitrary vector \vec{k} .

Any rotation matrix $R \in SO(3)$ can be described by a single rotation about an axis in space by an angle θ , $R = R_{\mathbf{k},\theta}$, and the problem of determining θ and \mathbf{k} , given a desired $R_{\mathbf{k},\theta}$ remains. Thus, the same problem as the one with the Euler angles appear. The solution for this problem is discussed in Chapter 2. From (A.1), (A.21) and direct manipulations of the entries of (A.21), the solution as in [16, 17] is

$$\theta = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{3}\right) \quad (\text{A.22})$$

$$\mathbf{k} = \frac{1}{2 \sin \theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}. \quad (\text{A.23})$$

The nonuniqueness of the axis/angle representation occurs because a rotation of $-\theta$ about $-\mathbf{k}$ is the same as a rotation of θ about \mathbf{k} ,

$$R_{\mathbf{k},\theta} = R_{-\mathbf{k},-\theta}. \quad (\text{A.24})$$

If $\theta = 0$, R is an identity matrix and the axis of rotation is undefined.

A.4 Homogeneous Transformation Matrix

Combining together the concepts to represent positions and orientations of rigid body presented before, the definition of a rigid motion is an ordered pair (d, R) where $d \in \mathbb{R}^3$ and $R \in SO(3)$. The group of all rigid motions is the Special Euclidean Group denoted by $SE(3)$ and $SE(3) = \mathbb{R}^3 \times SO(3)$ [16, 17]. A rigid motion is a pure translation together with a pure rotation

$$\mathbf{p}^0 = R_1^0 \mathbf{p}^1 + \mathbf{d}_1^0, \quad (\text{A.25})$$

and in the case of Figure 4

$$\mathbf{d}^6 = R_6^0 \mathbf{p}^6 + \mathbf{d}_6^0. \quad (\text{A.26})$$

Rotations were represented by rotation matrices and translations by Cartesian positions. The process to obtain (A.26) using (A.25) will require a lot of vectors and matrices operations and is not practical to implement. However, grouping the rotation matrix and the Cartesian positions vector as in (A.25) leads to homogeneous transformation matrices (HTM). They enable to describe a rigid motion in a more compact form

$$H = \begin{bmatrix} R_{3 \times 3} & \mathbf{d}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1_{1 \times 1} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}; R \in SO(3), d \in \mathbb{R}^3. \quad (\text{A.27})$$

The composition and ordering rules for homogeneous transformations are the same as for rotations described in Section A.2. The HTM to describe the end effector orientation and position in Figure 4 is, then, given by

$$H_6^0 = \begin{bmatrix} R_6^0 & \mathbf{d}_6^0 \\ \mathbf{0} & 1 \end{bmatrix}. \quad (\text{A.28})$$

B Rotation matrix of a quaternion

It is possible to obtain the rotation matrix corresponding to a given quaternion [17]

$$R_{\eta, \epsilon} = \begin{bmatrix} 2(\eta^2 + \epsilon_x^2) - 1 & 2(\epsilon_x \epsilon_y - \eta \epsilon_z) & 2(\epsilon_x \epsilon_z + \eta \epsilon_y) \\ 2(\epsilon_x \epsilon_y + \eta \epsilon_z) & 2(\eta^2 + \epsilon_y^2) - 1 & 2(\epsilon_y \epsilon_z - \eta \epsilon_x) \\ 2(\epsilon_x \epsilon_z - \eta \epsilon_y) & 2(\epsilon_y \epsilon_z + \eta \epsilon_x) & 2(\eta^2 + \epsilon_z^2) - 1 \end{bmatrix} = R \quad (\text{B.1})$$

and the inverse problem to compute the quaternion corresponding to a given rotation matrix R from (A.12) and (B.1) yields

$$\eta = \frac{1}{2} \sqrt{r_{11} + r_{22} + r_{33} + 1} \quad (\text{B.2})$$

$$\epsilon = \frac{1}{2} \begin{bmatrix} sgn(r_{32} - r_{23}) \sqrt{r_{11} - r_{22} - r_{33} + 1} \\ sgn(r_{13} - r_{31}) \sqrt{r_{22} - r_{33} - r_{11} + 1} \\ sgn(r_{21} - r_{12}) \sqrt{r_{33} - r_{11} - r_{22} + 1} \end{bmatrix} \quad (\text{B.3})$$

It is assumed that $sgn(x) = 1$ for $x \geq 0$ and $sgn(x) = -1$ for $x < 0$. From (B.2) $\eta \geq 0$, then angle $\theta \in [-\pi, \pi]$, and any rotation can be described.

C Circular and Squared Trajectory

The first three controllers to be implemented were the proportional controller (K controller) and proportional controller with feedforward term (K+FF controller) and the LQR controller in \mathbb{R}^8 manifold (LQR in \mathbb{R}^8). They were first tested by running a circular and a squared trajectory both on the V-REP simulator and on the Meka robot. The trajectories are independent and were executed separately. The robot initial joint vector configuration adopted for both tasks is given by $\theta_0 = [0, \pi/5, 0, \pi/2, 0, \pi/5, 0, 0]$.

The circular task is a circle with 7 cm radius and is drawn on the XZ-plane while maintaining Y constant. It contains 1500 points equally distant from each other. The sampling time used was $T_s = 25ms$ defined based on the lowest controller frequency or on the highest controller period. Table 1 gives an overview of this experiment.

Table 1: Overview of the circular trajectory experiment to compare the proportional controllers with the LQR in \mathbb{R}^8 controller.

Trajectory	Circle on xz-plane (Figure 5.1)
Initial joint configuration	$\theta_0 = [0, \pi/5, 0, \pi/2, 0, \pi/5, 0, 0]$
Trajectory parametrization	$x = r \sin(\theta), y = 0, z = r \cos(\theta), \theta \in [0, 2\pi]$
Radius	$r = 7$ cm
Sampling time	25 ms
Trajectory points	1500
LQR controller parameters	4 sets: $r \in \{0.001, 0.01, 0.1, 1\}$ and $q \in \{1, 10, 100, 1000\}$
Proportional controllers parameters	$k = 50$

The squared task is a square with 10 cm side, also on the XZ-plane while maintaining Y constant. It contain 1600 points equally distant from each other, but the first and third sides of the square have 500 points while the other two sides have 300 points. Table 2 gives an overview of this trajectory.

Table 2: Overview of the square trajectory experiment to compare the proportional controllers with the LQR in \mathbb{R}^8 controller.

Trajectory	Square on xz-plane (Figure 5.1)
Initial joint configuration	$\theta_0 = [0, \pi/5, 0, \pi/2, 0, \pi/5, 0, 0]$
Square size	$l = 0.1\text{m}$
Sampling time	25 ms
Trajectory points	Side 1: 500, side 2: 300, side 3: 500, side 4: 500
LQR controller parameters	4 sets: $r \in \{0.001, 0.01, 0.1, 1\}$ and $q \in \{1, 10, 100, 1000\}$
Proportional controllers parameters	$k = 50$

In order to numerically compare performance with the evaluation criteria, the LQR in \mathbb{R}^8 controller is distinguished by choosing a increasing weight for the control effort and error parameters r and q , respectively, that is $r \in \{0.001, 0.01, 0.1, 1\}$ and $q \in \{1, 10, 100, 1000\}$.

Tables 3 and 4 show the results for the trajectories run on the V-REP simulator and on the Meka robot respectively. $Err.$ is the integral norm of the error, $\dot{\theta}$ is the integral norm of the control signal, $Eff.$ is the integral norm of the control effort, and $L.Vel.$ and $L.Acc.$ are the end effector velocity and acceleration integral norms, respectively. This same notation is used on the data plots which also appear in this section.

Table 3: Simulation results obtained in V-REP performing two different end effector trajectories on the Kuka LBR iiwa 7.

Q	Circular trajectory				Squared trajectory				k=50	k=50		
	LQR				K	K+FF	LQR					
	1	10	100	1000	k=50	k=50	1	10	100	1000	k=50	k=50
R	0.001	0.01	0.1	1			0.001	0.01	0.1	1		
Err.	0.0251	0.0251	0.0413	0.2183	0.0197	0.0149	0.0192	0.0196	0.0191	0.0199	0.0131	0.0095
$\dot{\theta}$	0.1778	0.1774	0.1828	0.1797	0.0186	0.0189	0.1311	0.1335	0.1308	0.1348	0.1307	0.1328
Eff.	0.0605	0.0607	0.0609	0.0605	0.0625	0.0635	0.0431	0.0431	0.0431	0.0432	0.0433	0.0432
L.Vel.	0.5145	0.5295	0.5321	0.5302	0.5569	0.5568	0.4036	0.4029	0.4022	0.4032	0.4042	0.4035
L.Acc.	19.36	22.07	20.00	21.19	20.63	21.64	11.81	10.94	11.86	13.77	12.62	11.42

In Table 3, it is seen that the overall quantitative performance of the LQR controller was worse than the K or K+FF controller. An increase in r caused an slightly increase in the trajectory error norm. The control effort barely changed. However the end effector linear velocity and acceleration were, in most cases, increased with the increase in r . This can be associated with the increase in q which tries to lower the error by increasing velocity to compensate the higher r which tends to reduce joints control effort. Note that the end effector velocity of the LQR was, in most cases,

lower than both other controllers. Nevertheless, if q was high enough in comparison to r , than r could not slow the overall trajectory.

Table 4: Trajectory results obtained on Meka performing two different end effector trajectories.

Q	Circular trajectory					Squared trajectory					k=50	
	LQR				K	K+FF	LQR					
	1	10	100	1000	k=50	k=50	1	10	100	1000	k=50	
R	0.001	0.01	0.1	1			0.001	0.01	0.1	1		
Err.	3.332	3.340	3.265	3.216	2.220	2.217	3.648	3.628	3.549	3.600	2.293	2.247
$\dot{\theta}$	4.878	4.896	4.800	4.746	5.198	5.200	5.279	5.291	5.183	5.276	5.353	5.269
Eff.	0.0874	0.0878	0.0868	0.0868	0.0960	0.0981	0.0841	0.0843	0.0834	0.0820	0.0869	0.0879
L.Vel.	0.6322	0.6440	0.6461	0.6360	0.6657	0.6680	0.5692	0.5563	0.5619	0.5688	0.5780	0.5758
L.Acc.	14.51	13.24	13.14	14.50	15.03	14.61	13.23	12.99	12.10	13.10	13.40	13.56

In comparison to the simulation results, when the trajectory was run on the Meka robot, the results show a larger variation. Table 4 allows to see that an increase in r caused a decrease on the joints control effort as expected. Moreover, the increase in q showed a better result on the error norm which has decreased. Therefore, it is seen again, that an increase in q has a better effect on diminishing the trajectory error than a decrease in r . Once more, the minimization chosen in the cost function of (3.39) showed the desired effect in the LQR in \mathbb{R}^8 controller trajectory tracking capability. Note that the joints control effort and also end effector velocity of the LQR controller were lower than the K and K+FF controllers.

A similar analysis in a qualitative manner may be done with Figures 5 and 6 which represent the results of both trajectories run on the Meka robot.

Proceeding with the analysis of Figures 5 and 6, the linear trajectory of the end effector and the unit dual quaternion representing orientation shows that the LQR in \mathbb{R}^8 controller do not follow the expected trajectory as precise as the K and K+FF controller. However, the LQR in \mathbb{R}^8 controller has the lowest end effector velocity (*L.Vel.*) and acceleration (*L.Acc.*) initial peak for both trajectories. Furthermore, it also has lower velocity and acceleration peaks along the whole motion. This can, consequently, also be seen in the joints control effort (*Eff.*) and control signal ($\dot{\theta}$) plots. In this sense, the robot joints suffer less abrupt torques and forces and, hence, the LQR controller may contribute to the robot life span.

A small addition may also be made. The initial joints control effort peaks of the circular trajectory seems to be higher on the LQR controller. However, in less than five seconds the K and K+FF controller peaks increase more than the LQR controller. The fast decrease in the LQR controller *Eff.* peaks also goes for the squared trajectory.

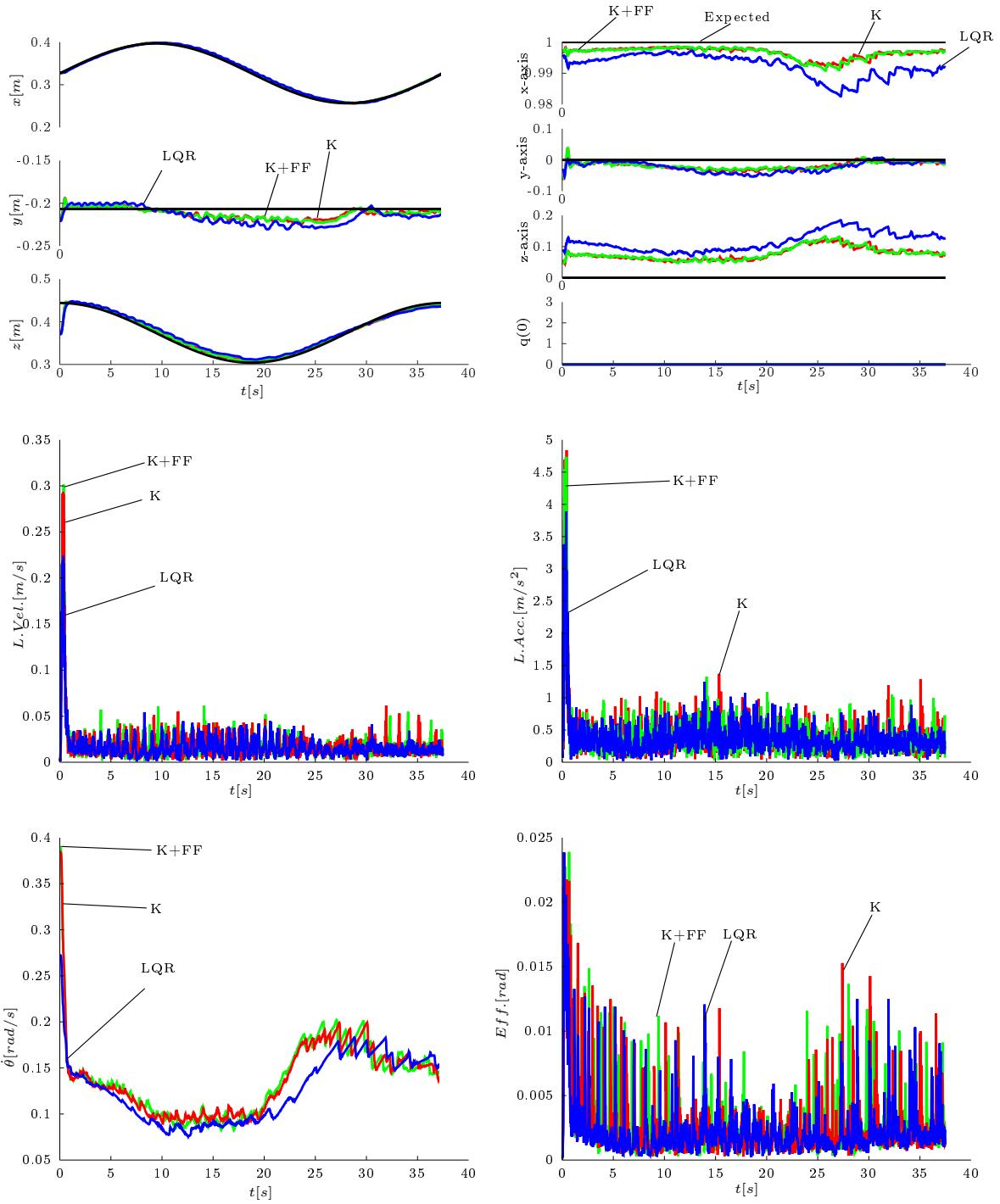


Figure 5: Circular trajectory with $r = 1$ and $q = 1000$

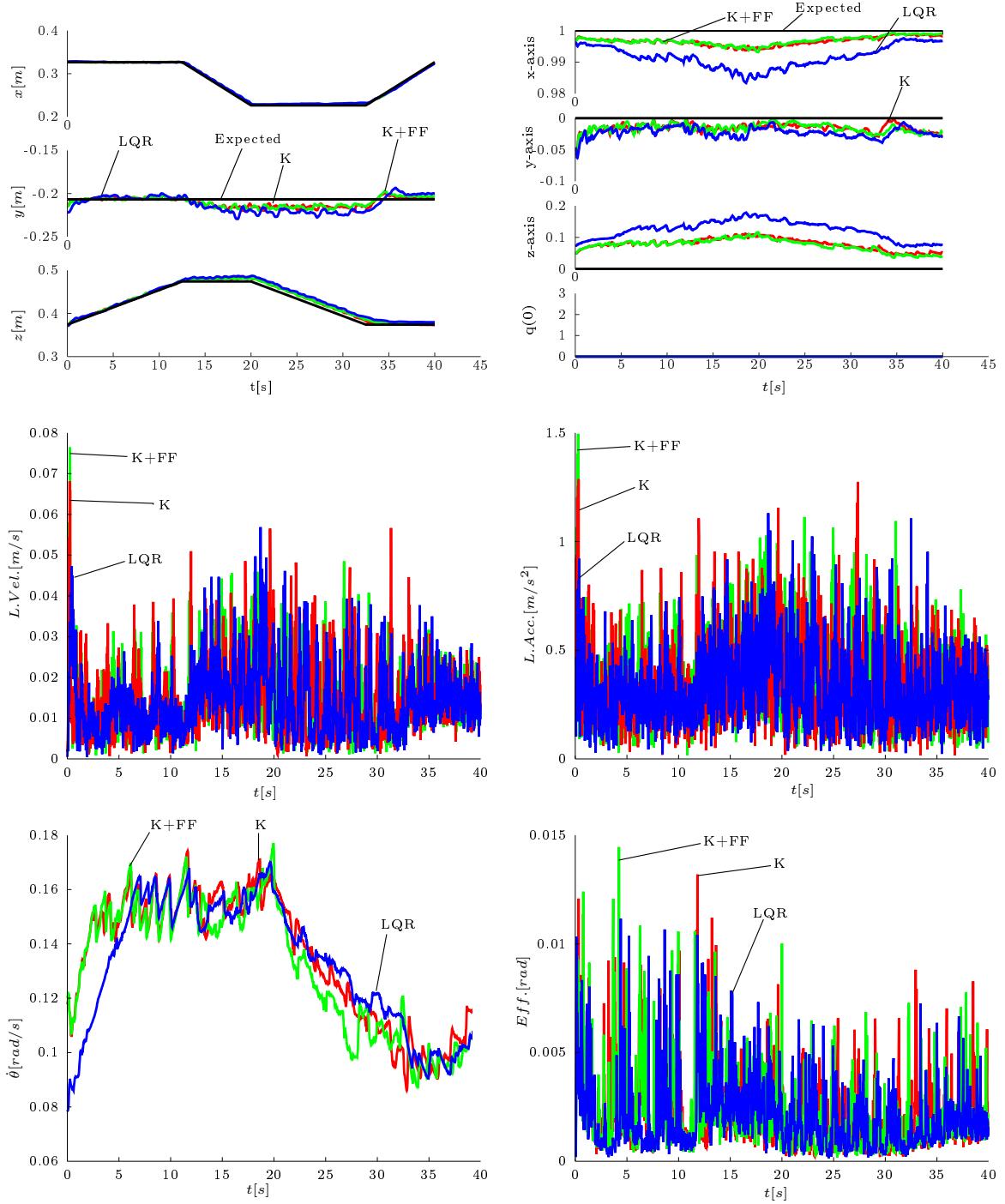


Figure 6: Square trajectory with $r = 1$ and $q = 1000$

D LQR in \mathbb{R}^8 in Depth Analysis Performance

Concerning the performance of the LQR controller and the Q and R parameters influence on the joints control effort, the end effector velocity norm and the invariant error norm the circular trajectory was run again. An increasing weight for the control effort r and the error parameter q was chosen. The results were analyzed both in the V-REP simulator with the Kuka LBR iiwa 7 R800 industrial robot and with the Meka robot.

Table 5: Overview of the circular trajectory experiment to evaluate the LQR in \mathbb{R}^8 controller behavior on the Meka robot.

Trajectory	Circle on xz-plane (Figure 5.1)
Initial joint configuration	$\theta_0 = [0, \pi/5, 0, \pi/2, 0, \pi/5, 0, 0]$
Trajectory parametrization	$x = r \sin(\theta), y = 0, z = r \cos(\theta), \theta \in [0, 2\pi]$
Radius	$r = 7$ cm
Sampling time	25 ms
Trajectory points	1500
LQR controller parameters	10 sets: $r \in \{0.05, 0.1, 0.5, 1, 5, 10, 50, 100\}$ and $q \in \{1, 10, 20, 30, 40, 50, 60, 70\}$

For the simulation, $r \in \{0.05, 0.1, 0.5, 1, 5, 10, 50, 100\}$ while $q \in \{1, 10, 20, 30, 40, 50, 60, 70\}$. On the Meka robot, $r \in \{0.05, 0.1, 1, 10, 100\}$ and $q \in \{1, 20, 40, 60, 80\}$.

In the interest of better visualizing the influence of the LQR parameters, the results were plotted to form a surface. Three surfaces are presented where their height represents the integral value along the whole trajectory of the joints control effort norm, the end effector velocity norm and the invariant error norm. Note that the increase in q was not so strong as in Tables 3 and 4. The q gain was kept lower in order to obtain stability for the whole trajectory considering the r variation and still be able to get enough data to build a surface. For this reason, some results are not exactly as expected.

The simulation results in Figure 7 show, as expected, that an increase in r delivers a lower control effort as long as q is not dramatically increased. The end effector velocity norm will be higher when r is decreased and q is increased accordingly. It is expected that an increase in q and a lower r will result in lower errors, however, the error norm surfaces obtained show that this will not always happen. If q is not high enough while r is too low or too high, the velocities achieved may deliver a slight increase in error. The controller may be not be capable of compensating the error for such velocities. A solution for this problem is seen both in Tables 3 and 4, a high increase in q reduced the trajectory error norm.

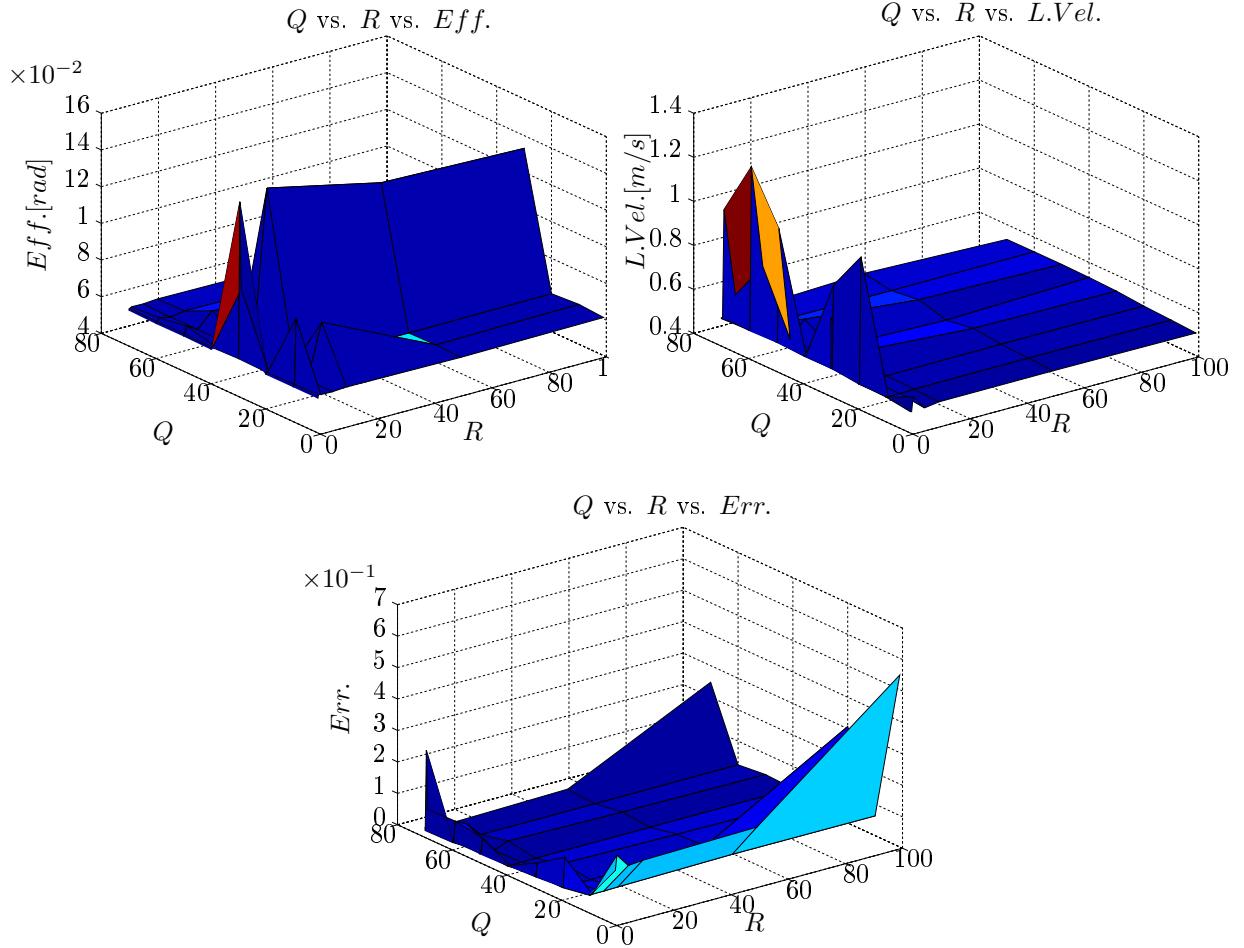


Figure 7: LQR parameters surface on Kuka LBR iiwa 7 on V-REP

A similar analysis can be done for the results on the Meka Robot using Figure 8. The V-REP simulator incorporates the dynamical properties of the robot and the environment, but, in comparison to the real robot, they vary much less. Furthermore, the kinematic control used in this work does not include dynamics, forces and torques. Therefore, the results on the Meka robot show some disparities. The joints control effort norm do not decrease with the increase in r , if q is not properly increased. During the trajectory execution it was seen that the controller could not rightly follow the desired trajectory if r was too high and q not so drastically increased. This could explain the divergences obtained in the error in which the error was lower for a smaller q and lower r . Despite of the divergence of the error, the end effector velocity behaved itself as expected, for a higher r and lower q , the end effector moves slower as far as the trajectory is correctly being followed with a lower error.

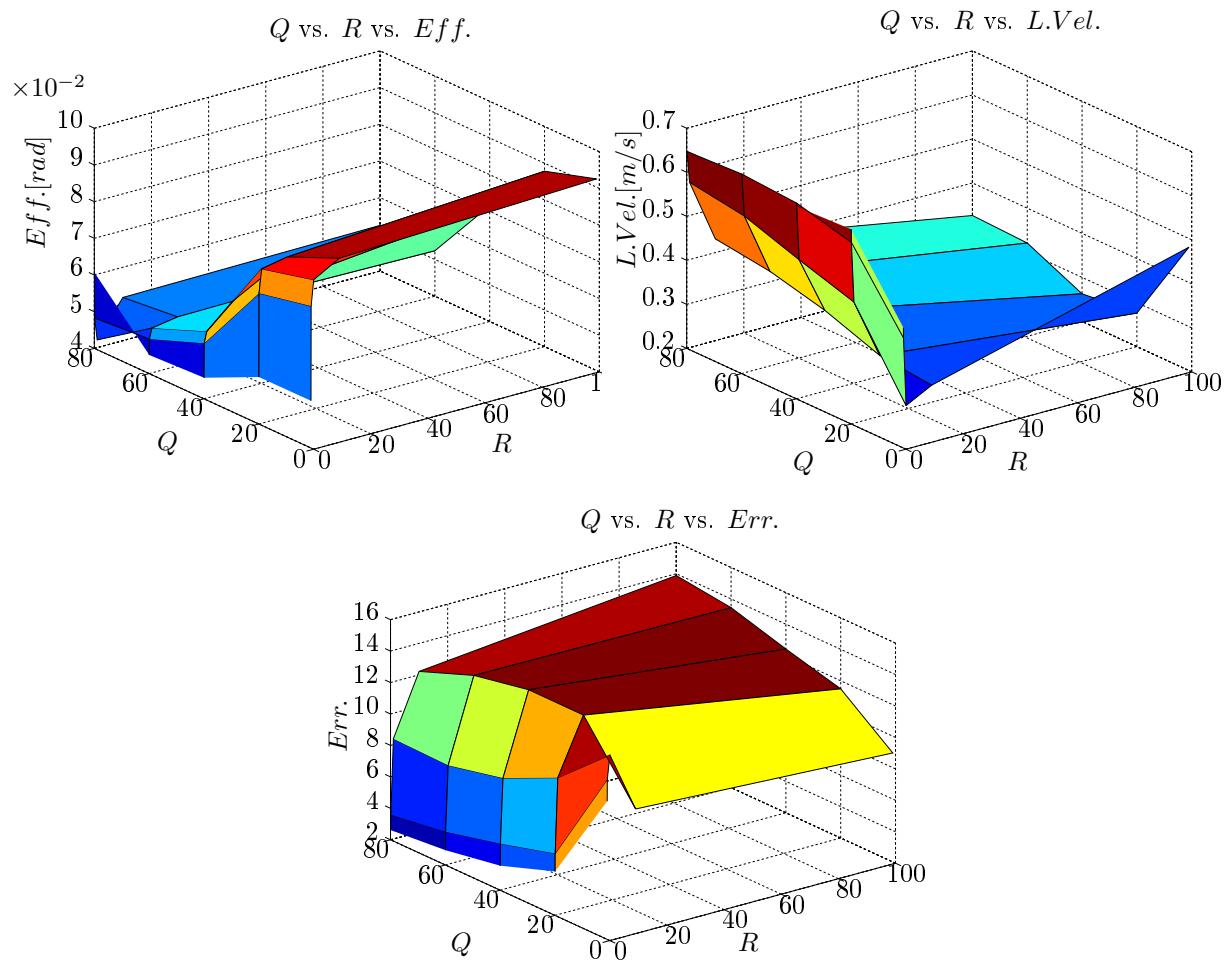


Figure 8: LQR parameters surface on Meka robot

E Tables of Results

E.1 Tables from Circular Trajectory with $r = 0.001, q = 1$ —Sections 5.4 and 5.6

Table 6: LQR controller with $r = 0.001, q = 1$: Integral norm of the end-effector invariant error.

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.001, q = 1$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	3.128367	3.111993	4.773411	5.599988	1.481183	1.513701
2	3.143041	3.005740	4.734508	5.462487	1.528755	1.513384
3	3.110902	3.232741	4.746095	5.466275	1.442317	1.506571
4	3.172984	3.065525	4.734550	5.837564	1.521314	1.509682
5	3.103292	3.183072	4.908637	5.482014	1.499819	1.501389
6	3.011214	3.124857	4.957241	5.555529	1.504782	1.526967
7	3.128492	3.013988	4.960959	5.309714	1.488762	1.516047
8	3.236896	3.300024	5.079988	5.786483	1.510137	1.528165
9	3.310722	3.304715	5.015199	5.567993	1.492056	1.508575
10	3.246760	3.270729	4.904716	5.861354	1.489426	1.528850

Table 7: LQR controller with $r = 0.001, q = 1$: Control output $\dot{\theta}$.

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.001, q = 1$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	311.037168	310.402087	301.868174	319.269808	368.972470	375.168101
2	312.629730	303.216900	300.842179	310.316613	379.924969	375.159911
3	310.035336	321.300264	300.598134	312.587799	361.589695	374.534059
4	314.659601	308.911881	300.275683	331.176162	375.709128	375.794078
5	309.610052	316.801997	307.561037	314.074326	371.341757	372.234980
6	301.380326	312.300715	307.803059	315.615006	373.549999	379.056367
7	310.965094	303.151137	309.790510	304.235086	371.470844	376.535829
8	324.135357	330.244751	319.406773	327.093552	376.389168	379.863511
9	329.218077	332.379851	313.797516	315.684269	374.379229	376.061720
10	326.177986	328.962727	308.873003	332.140058	372.660163	381.162825

Table 8: LQR controller with $r = 0.001, q = 1$: Joints Control Effort Norm (Eff.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.001, q = 1$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	0.117964	0.117395	0.111321	0.112979	0.138753	0.141051
2	0.119017	0.117910	0.110463	0.110235	0.143544	0.140937
3	0.118709	0.118047	0.108275	0.112962	0.141903	0.142055
4	0.118379	0.117590	0.110718	0.118669	0.142404	0.140961
5	0.117933	0.118318	0.109275	0.116037	0.141599	0.145783
6	0.117305	0.118949	0.112137	0.116899	0.143023	0.141588
7	0.118983	0.119208	0.110455	0.110883	0.142887	0.143033
8	0.115465	0.115388	0.111045	0.109965	0.133489	0.138636
9	0.115455	0.114737	0.111985	0.113115	0.136385	0.138377
10	0.115336	0.113511	0.109558	0.115313	0.135808	0.137756

Table 9: LQR controller with $r = 0.001, q = 1$: End effector linear velocity (L. Vel.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.001, q = 1$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	0.989696	0.982646	0.957519	1.016560	1.255906	1.306051
2	0.973547	0.984773	0.987741	0.971089	1.264021	1.271009
3	0.986715	0.989741	0.962186	0.998487	1.222288	1.305377
4	0.983419	0.980696	0.982293	1.038601	1.296339	1.270802
5	0.985127	0.993482	0.961477	1.029174	1.301607	1.312345
6	0.979948	0.995573	0.974686	1.003628	1.302144	1.264491
7	0.991657	1.000678	0.968137	0.978986	1.296664	1.291550
8	0.984080	0.993064	0.974552	0.962870	1.176083	1.285176
9	0.997709	0.986662	0.966618	0.961197	1.302639	1.282935
10	0.986021	0.991169	0.945484	1.014948	1.234499	1.296365

Table 10: LQR controller with $r = 0.001$, $q = 1$: End effector linear acceleration (L. Acc.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.001, q = 1$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	22.916654	30.482965	22.530082	24.727423	26.696148	26.900505
2	22.815918	22.607981	23.178839	22.940774	26.587019	27.051911
3	23.272522	23.715574	22.565928	25.046173	26.226317	27.187062
4	23.077284	22.831542	23.607238	27.041424	27.335831	26.204175
5	23.208757	23.190118	23.092996	26.329517	27.732785	27.382558
6	22.701270	22.789812	23.361433	25.213418	31.222485	26.864843
7	23.139270	23.452011	23.244298	23.857070	27.036465	28.187437
8	22.313654	23.065582	32.582470	22.867501	25.459152	26.365433
9	23.504121	22.588686	23.498653	22.978083	28.290839	26.321159
10	22.504249	22.176451	23.273069	26.296261	27.190694	26.946540

E.2 Tables from Circular Trajectory with $r = 0.01$, $q = 10$ —Sections 5.4 and 5.6

Table 11: LQR controller with $r = 0.01, q = 10$: Integral norm of the end-effector invariant error.

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.01, q = 10$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	3.279806	3.291509	5.013098	5.760778	1.514283	1.528292
2	3.109048	3.048276	4.745299	5.739577	1.479808	1.477260
3	3.088182	3.049559	4.842139	5.829555	1.501052	1.487824
4	3.128897	3.079712	4.723980	5.321395	1.446299	1.509183
5	3.183707	3.136173	4.781896	5.728371	1.445686	1.506773
6	3.111173	3.090586	4.850429	5.334785	1.515239	1.502841
7	3.175278	3.147619	4.868070	5.525922	1.517182	1.500992
8	3.137127	3.149314	4.739240	5.479084	1.514157	1.547204
9	3.208774	3.141802	4.847400	5.695411	1.540820	1.532840
10	3.341611	3.318165	5.169726	5.934421	1.611344	1.633253

 Table 12: LQR controller with $r = 0.01, q = 10$: Control output $\dot{\theta}$.

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.01, q = 10$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	327.644890	334.294935	313.487339	324.083413	379.158698	379.115081
2	311.866427	306.924405	301.694955	326.561690	369.117726	370.904534
3	310.274646	306.757543	304.852437	330.169273	375.073979	371.815811
4	314.353218	314.137501	300.285466	307.306949	364.114549	376.604005
5	317.808741	315.909959	299.535172	324.770096	360.635230	375.775251
6	307.938199	311.592830	302.752087	303.533350	375.785028	372.838083
7	316.867318	315.303378	304.336644	315.014406	376.338514	373.113336
8	314.803843	315.116687	299.519550	311.636725	375.375340	383.029638
9	320.368530	314.328679	302.166628	323.715943	380.987872	379.497896
10	331.624149	328.755956	320.021225	332.774660	398.671178	403.851860

Table 13: LQR controller with $r = 0.01, q = 10$: Joints Control Effort Norm (Eff.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.01, q = 10$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	0.114312	0.122055	0.108220	0.113674	0.134955	0.135518
2	0.115339	0.115779	0.107419	0.116540	0.135759	0.135539
3	0.115058	0.114404	0.107448	0.120605	0.139093	0.139514
4	0.115651	0.116503	0.108574	0.111791	0.137878	0.140643
5	0.117730	0.116771	0.108943	0.119985	0.139579	0.141269
6	0.116528	0.115989	0.112196	0.114250	0.138498	0.140274
7	0.115121	0.116183	0.109087	0.114079	0.136166	0.142696
8	0.114924	0.116215	0.111211	0.112228	0.140185	0.139110
9	0.115063	0.114888	0.108134	0.117143	0.139187	0.140797
10	0.119312	0.119249	0.111256	0.120197	0.145972	0.150701

Table 14: LQR controller with $r = 0.01, q = 10$: End effector linear velocity (L. Vel.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.01, q = 10$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	0.985045	1.026824	0.950634	0.996898	1.238909	1.274431
2	0.999239	0.987223	0.974203	1.033382	1.238542	1.214609
3	0.992187	0.993154	0.970346	1.034153	1.346329	1.261576
4	0.993333	1.002342	0.972692	0.963097	1.281394	1.349574
5	0.999381	0.989022	0.974634	1.028729	1.284474	1.297131
6	0.997462	0.993287	0.969356	0.986082	1.257085	1.290371
7	0.987895	0.983929	0.971139	0.986995	1.199455	1.314910
8	0.976529	0.985394	0.952404	1.004500	1.278610	1.277611
9	0.993576	0.996461	0.969554	1.019314	1.255219	1.267206
10	0.986456	0.987717	0.981368	1.018534	1.325124	1.354712

Table 15: LQR controller with $r = 0.01$, $q = 10$: End effector linear acceleration (L. Acc.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.01, q = 10$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	22.408657	22.662297	23.409343	25.108651	25.552879	25.984867
2	22.121794	22.054782	23.274637	26.298487	25.866053	25.819997
3	29.983563	22.515183	27.788416	27.189799	27.220253	26.592841
4	23.055553	23.168932	22.619764	23.622339	27.172178	27.377939
5	23.472819	23.255432	23.198917	27.309489	27.037386	26.974257
6	22.954181	23.715481	23.685605	23.845141	26.695643	27.006101
7	23.244493	25.159734	34.999155	25.301207	26.743042	27.558097
8	22.809555	22.849953	22.649749	25.701431	29.826394	26.828696
9	23.271223	26.245662	23.480860	27.095652	26.786403	26.936612
10	23.763140	23.268542	25.939655	26.871093	30.678912	28.357721

E.3 Tables from Circular Trajectory with $r = 0.1$, $q = 100$ –Sections 5.4 and 5.6

Table 16: LQR controller with $r = 0.1, q = 100$: Integral norm of the end-effector invariant error.

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.1, q = 100$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	3.224737	3.308168	5.074394	5.761797	1.602424	1.581452
2	3.353828	3.353356	5.164810	6.026074	1.587569	1.608253
3	3.301526	3.331762	5.141589	5.653705	1.582725	1.581338
4	3.412422	3.399187	5.201922	5.849904	1.571771	1.583000
5	3.337518	3.412450	4.811209	6.087800	1.610040	1.573865
6	3.389497	3.253588	5.041014	5.968023	1.595498	1.595126
7	3.412940	3.317498	5.017600	6.058584	1.591707	1.589469
8	3.315143	3.372180	4.915169	5.644469	1.594419	1.572659
9	3.241784	3.280417	4.929389	5.771981	1.511240	1.513258
10	3.335754	3.219715	5.043219	5.731621	1.534127	1.522916

Table 17: LQR controller with $r = 0.1, q = 100$: Control output $\dot{\theta}$.

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.1, q = 100$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	321.934581	330.244394	315.040121	324.542926	397.858381	394.721239
2	330.964909	333.348820	319.793289	338.537658	392.403486	398.618072
3	328.834676	330.136903	318.850378	319.944637	394.359122	392.874071
4	337.423309	336.571613	322.062408	329.509344	391.378691	393.568228
5	331.962562	338.410155	301.854695	343.028151	399.270217	391.896988
6	335.488553	326.953207	313.655097	337.148628	396.393992	395.377834
7	336.910957	328.825790	312.934202	339.792619	395.565895	395.376528
8	328.752459	336.018045	307.732171	319.576367	396.538751	391.956003
9	324.675187	328.410575	311.510249	327.238694	378.973141	380.703815
10	332.806793	322.945504	316.338718	326.115334	384.354753	381.407409

Table 18: LQR controller with $r = 0.1, q = 100$: Joints Control Effort Norm (Eff.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.1, q = 100$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	0.120105	0.118990	0.112582	0.116444	0.145734	0.147949
2	0.120918	0.119081	0.114737	0.124235	0.145274	0.147683
3	0.118934	0.120498	0.113528	0.115675	0.146153	0.148944
4	0.120904	0.119891	0.111059	0.117815	0.145662	0.147966
5	0.119840	0.120045	0.112583	0.123855	0.148917	0.146711
6	0.119273	0.118883	0.112599	0.120089	0.146873	0.148143
7	0.119436	0.120417	0.112002	0.122046	0.143755	0.146882
8	0.120029	0.118846	0.108672	0.114765	0.146033	0.149541
9	0.111848	0.114319	0.108205	0.111517	0.133151	0.135614
10	0.113686	0.114007	0.107176	0.115806	0.134775	0.135279

Table 19: LQR controller with $r = 0.1, q = 100$: End effector linear velocity (L. Vel.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.1, q = 100$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	0.988990	0.985132	0.969637	1.011656	1.267878	1.304507
2	0.995297	0.991134	0.968109	1.079345	1.300797	1.313210
3	0.999152	1.017452	0.962429	0.974584	1.286807	1.335026
4	1.004261	0.995093	0.974065	0.987071	1.308882	1.257524
5	1.005842	0.984460	0.958917	1.054453	1.324277	1.279808
6	0.992483	1.001436	0.961113	1.014161	1.300913	1.313290
7	0.993654	1.005098	0.976861	1.042165	1.232417	1.295692
8	1.004595	1.008007	0.960594	0.965452	1.321664	1.343360
9	0.982786	0.993645	0.966248	0.985445	1.244177	1.275816
10	0.990627	0.976424	0.957850	1.015194	1.283581	1.259794

Table 20: LQR controller with $r = 0.1$, $q = 100$: End effector linear acceleration (L. Acc.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 0.1, q = 100$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	23.444362	24.692603	23.278344	25.384412	28.759955	27.528697
2	24.012440	23.111159	25.216067	29.189016	27.892882	27.740110
3	23.946524	24.160492	23.576793	24.904633	27.904901	27.962885
4	24.284863	24.403648	23.577492	25.088770	27.751314	27.595485
5	24.062990	25.631242	22.998993	29.580290	28.309808	27.622409
6	23.953182	23.929086	23.378479	26.626735	27.717127	27.542349
7	23.906289	23.559295	23.316952	28.105313	26.945552	27.741331
8	23.704642	24.186455	23.198393	23.647196	28.402500	27.585589
9	22.406002	22.864331	21.719535	24.701596	25.517425	25.995551
10	22.419965	22.101138	22.090469	25.309240	26.591739	25.610895

E.4 Tables from Circular Trajectory with $r = 1$, $q = 1000$ —Sections 5.4 and 5.6

Table 21: LQR controller with $r = 1, q = 1000$: Integral norm of the end-effector invariant error.

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 1, q = 1000$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	3.265432	3.209739	5.019396	5.865678	1.539791	1.540600
2	3.224401	3.333474	5.016777	6.059302	1.529502	1.538543
3	3.297897	3.266339	4.956768	5.972665	1.543325	1.545404
4	3.283132	3.372171	4.992939	5.984452	1.537390	1.538964
5	3.277682	3.326582	5.065056	5.814905	1.536915	1.540114
6	3.191356	3.328606	4.931744	5.913453	1.539181	1.518566
7	2.992234	3.007952	4.628419	5.367810	1.472481	1.483363
8	3.091167	3.184765	4.695239	5.066122	1.488619	1.478364
9	2.959533	2.948301	4.386905	5.519525	1.493555	1.511719
10	3.094144	3.027774	4.833367	5.311782	1.458942	1.514873

Table 22: LQR controller with $r = 1, q = 1000$: Control output $\dot{\theta}$.

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 1, q = 1000$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	324.815487	324.282839	315.335426	332.772841	384.109247	385.713681
2	324.520663	333.805590	315.349581	342.715077	383.279766	385.078193
3	328.732263	326.865811	310.314767	337.802762	386.179844	385.500458
4	327.334760	335.045952	312.003783	338.885927	383.572075	384.114433
5	325.933380	331.188545	317.465060	330.998746	383.629720	385.653382
6	319.747406	330.338874	308.759656	338.061261	383.446787	379.807165
7	301.843978	304.438854	295.289583	309.581214	364.976295	369.326651
8	309.911887	319.126440	299.285492	294.070526	370.275133	369.636448
9	301.872737	300.395705	285.884171	319.336303	371.817146	375.224422
10	313.027698	309.462147	305.161424	309.250032	363.183240	377.434490

Table 23: LQR controller with $r = 1$, $q = 1000$: Joints Control Effort Norm (Eff.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 1, q = 1000$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	0.113099	0.115107	0.108437	0.114733	0.135654	0.138184
2	0.115301	0.115276	0.107171	0.118218	0.137117	0.138861
3	0.114927	0.115730	0.106979	0.120137	0.137476	0.136693
4	0.113584	0.115417	0.111129	0.117362	0.137659	0.135081
5	0.116148	0.114296	0.109674	0.113927	0.137779	0.138631
6	0.113797	0.114098	0.108127	0.118553	0.138040	0.139248
7	0.115566	0.116961	0.106988	0.112658	0.138568	0.138673
8	0.115730	0.116499	0.107182	0.109905	0.140450	0.138309
9	0.117840	0.118670	0.108174	0.117620	0.142617	0.143737
10	0.119125	0.117344	0.109750	0.113857	0.142240	0.143876

Table 24: LQR controller with $r = 1$, $q = 1000$: End effector linear velocity (L. Vel.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 1, q = 1000$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	0.971134	0.988900	0.938537	1.008629	1.285433	1.277422
2	0.989915	0.985503	0.960147	1.037774	1.307272	1.274731
3	0.973104	0.993790	0.959436	1.047060	1.333427	1.257763
4	0.993189	0.980965	0.950813	1.033642	1.273288	1.216551
5	0.976480	0.990089	0.956889	0.997794	1.313807	1.283705
6	0.966204	0.996507	0.967900	1.049008	1.317216	1.270714
7	0.986870	0.984383	0.965687	1.002390	1.262150	1.241089
8	0.975551	0.997825	0.964017	0.976350	1.246471	1.239499
9	0.976134	0.978169	0.967988	1.018999	1.271200	1.245869
10	1.004319	0.999232	0.970647	0.992014	1.232170	1.267514

Table 25: LQR controller with $r = 1$, $q = 1000$: End effector linear acceleration (L. Acc.).

Experiment	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 50$	$k = 50$	$r = 1, q = 1000$		$\gamma_T = 0.471, \gamma_O = 0.666$	
1	21.917141	23.073463	22.404420	26.045646	26.641037	26.301895
2	23.166972	22.505385	23.187434	27.709829	26.588317	39.678952
3	22.486977	23.133859	22.512780	27.882955	27.128130	26.828229
4	23.108805	22.586362	22.854689	27.654520	26.742009	26.546402
5	23.242579	22.890033	22.655668	25.310052	26.674973	26.447638
6	22.932756	23.238637	22.821124	27.946382	26.909805	26.524620
7	22.821866	22.760494	21.800936	24.788215	26.855014	26.433429
8	22.666487	22.712969	22.107479	23.434383	26.674017	26.258426
9	23.107193	23.089608	22.607370	25.893763	26.792061	26.784458
10	22.749556	23.058656	23.529205	25.043114	26.657251	27.409088

E.5 Tables from Dual Quaternion H_∞ Controllers Performance Analysis–Section 5.5

Table 26: H_∞ controller behavior: Integral norm of the end-effector invariant error.

Exp.	HIR		HIRT		HIR		HIRT		HIR		HIRT	
	$\gamma_T = \gamma_O = 1.86$		$\gamma_T = \gamma_O = 0.930$		$\gamma_T = \gamma_O = 0.620$		$\gamma_T = \gamma_O = 0.465$		$\gamma_T = \gamma_O = 0.372$			
1	3.105020	3.130416	1.661440	1.655479	1.129358	1.114526	0.836497	0.821427	0.660390	0.665032		
2	3.141411	3.121509	1.669011	1.668547	1.135748	1.143185	0.842166	0.829010	0.654989	0.654544		
3	3.149936	3.130758	1.647908	1.664758	1.139455	1.126871	0.844095	0.836463	0.665998	0.666279		
4	3.142550	3.112167	1.649165	1.653133	1.152585	1.121572	0.823881	0.843082	0.653196	0.664149		
5	3.125664	3.096059	1.672417	1.666831	1.132126	1.120526	0.836549	0.815955	0.668965	0.659987		

Table 27: H_∞ controller behavior: Control output $\dot{\theta}$.

Exp.	HIR		HIRT		HIR		HIRT		HIR		HIRT	
	$\gamma_T = \gamma_O = 1.86$		$\gamma_T = \gamma_O = 0.930$		$\gamma_T = \gamma_O = 0.620$		$\gamma_T = \gamma_O = 0.465$		$\gamma_T = \gamma_O = 0.372$			
1	339.48781	342.81488	372.05950	373.46463	391.06247	386.64510	393.94487	388.58918	394.87609	395.15948		
2	344.71173	343.16210	374.73962	376.71878	391.68426	393.15636	396.05764	390.41028	392.42381	392.35311		
3	346.86515	346.77980	373.10258	375.34005	392.41530	389.64128	394.37641	392.62521	396.33665	398.63703		
4	345.90233	342.22322	375.20577	373.47260	395.84942	389.50017	390.03651	394.29313	391.58787	397.49209		
5	343.30657	339.76798	374.16866	375.63088	390.98457	388.24603	393.56817	387.10033	397.97581	395.20277		

Table 28: H_∞ controller behavior: Joints Control Effort Norm (Eff.).

Exp.	HIR		HIRT		HIR		HIRT		HIR		HIRT	
	$\gamma_T = \gamma_O = 1.86$		$\gamma_T = \gamma_O = 0.930$		$\gamma_T = \gamma_O = 0.620$		$\gamma_T = \gamma_O = 0.465$		$\gamma_T = \gamma_O = 0.372$			
1	0.038356	0.038931	0.049315	0.050697	0.054863	0.055767	0.058665	0.059208	0.064671	0.065262		
2	0.037764	0.038363	0.050286	0.050779	0.055225	0.055470	0.058245	0.058622	0.064757	0.062812		
3	0.037182	0.037967	0.050505	0.050048	0.055884	0.054174	0.059229	0.059090	0.065315	0.064421		
4	0.038072	0.038209	0.050661	0.049855	0.055017	0.055089	0.059033	0.059295	0.065598	0.065597		
5	0.038153	0.038424	0.050447	0.050667	0.055233	0.054927	0.058851	0.058298	0.063132	0.065776		

Table 29: H_∞ controller behavior: End effector linear velocity (L. Vel.).

Exp.	HIR		HIRT		HIR		HIRT		HIR		HIRT	
	$\gamma_T = \gamma_O = 1.86$		$\gamma_T = \gamma_O = 0.930$		$\gamma_T = \gamma_O = 0.620$		$\gamma_T = \gamma_O = 0.465$		$\gamma_T = \gamma_O = 0.372$			
1	1.053734	1.076614	1.245332	1.259511	1.343930	1.341329	1.431315	1.438452	1.608259	1.618468		
2	1.068187	1.069328	1.249841	1.249246	1.346105	1.349712	1.407349	1.431988	1.617426	1.579939		
3	1.033594	1.059773	1.253032	1.253475	1.353144	1.328162	1.445617	1.451929	1.624849	1.601669		
4	1.044420	1.061176	1.248319	1.247093	1.345231	1.331409	1.424587	1.456308	1.634280	1.634385		
5	1.054070	1.058839	1.259157	1.257261	1.343191	1.332821	1.427155	1.427501	1.580866	1.651597		

Table 30: H_∞ controller behavior: End effector linear acceleration (L. Acc.).

Exp.	HIR		HIRT		HIR		HIRT		HIR		HIRT	
	$\gamma_T = \gamma_O = 1.86$		$\gamma_T = \gamma_O = 0.930$		$\gamma_T = \gamma_O = 0.620$		$\gamma_T = \gamma_O = 0.465$		$\gamma_T = \gamma_O = 0.372$			
1	94.49797	95.80462	111.8452	110.8942	122.3788	121.2549	126.4543	130.9134	140.4273	138.9856		
2	102.1279	95.72132	108.1474	110.1813	118.5676	116.2885	126.6149	123.4663	136.3896	137.8304		
3	94.54717	95.25620	109.2199	108.7551	118.1414	117.7075	132.6424	124.9104	143.6406	136.3956		
4	99.78237	95.25322	107.9975	108.3886	117.8383	117.4721	123.6863	131.0172	138.9012	138.8806		
5	95.01788	94.08591	117.9161	108.8290	120.9819	116.3667	121.2046	124.3395	138.6248	141.9394		

E.6 Tables from Spiral Trajectory—Section 5.7

E.6.1 $(\ln 2)^2$ of the best performance

Table 31: Integral norm of the end-effector invariant error: spiral trajectory with $(\ln 2)^2$ of the best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 60.057$	$k = 60.057$	$q = 129.722, q = 0.01$		$\gamma_T = 0.775, \gamma_O = 0.786$	
1	2.268931	2.276568	1.323556	1.490234	1.747520	1.737543
2	2.229215	2.265468	1.344884	1.470916	1.752185	1.713109
3	2.275881	2.331420	1.353992	1.561978	1.736945	1.765156
4	2.439919	2.432450	1.394379	1.570548	1.876606	1.893139
5	2.483150	2.466904	1.455097	1.641296	1.919997	1.911240
6	2.441242	2.430732	1.459092	1.651180	1.873756	1.904767
7	2.485043	2.474700	1.476800	1.663814	1.864008	1.919127
8	2.512113	2.496303	1.437371	1.587033	1.872599	1.900597
9	2.473074	2.535148	1.471238	1.578637	1.904214	1.917390
10	2.494314	2.458293	1.466971	1.588187	1.854626	1.890730
11	2.514821	2.475940	1.460671	1.636987	1.899454	1.907780
12	2.434070	2.395845	1.405276	1.526140	1.816358	1.860744
13	2.436220	2.423770	1.422719	1.560727	1.850162	1.852220
14	2.426624	2.423094	1.418287	1.618067	1.856332	1.875963
15	2.449063	2.422059	1.391765	1.625861	1.860494	1.868759
16	2.464254	2.484265	1.390019	1.556193	1.864220	1.885607
17	2.449078	2.458137	1.388233	1.578789	1.853750	1.876439
18	2.440889	2.460179	1.437709	1.552346	1.865745	1.862778
19	2.475485	2.490195	1.449897	1.649099	1.843762	1.853482
20	2.430912	2.433455	1.431069	1.557230	1.866149	1.876663
21	2.398399	2.424250	1.432139	1.641966	1.876667	1.881869
22	2.437336	2.456539	1.438684	1.565529	1.862539	1.859809
23	2.427957	2.473488	1.423501	1.546684	1.874663	1.852118
24	2.442517	2.423821	1.410946	1.528640	1.859646	1.870732
25	2.455300	2.445113	1.375088	1.526087	1.887840	1.840770
26	2.454429	2.448432	1.417771	1.623552	1.849200	1.850271
27	2.450052	2.425915	1.405210	1.542290	1.840962	1.824602
28	2.438835	2.456820	1.398059	1.527039	1.829434	1.834108
29	2.409778	2.428810	1.395896	1.531834	1.831790	1.824838
30	2.450441	2.450334	1.416950	1.575659	1.830605	1.862087
31	2.420430	2.406738	1.369727	1.558557	1.806434	1.852238
32	2.433244	2.431419	1.392136	1.537414	1.802488	1.840154
33	2.387488	2.400458	1.412697	1.600479	1.828577	1.852504
34	2.426813	2.415202	1.407113	1.602991	1.818940	1.841981
35	2.403027	2.413592	1.411558	1.567384	1.813315	1.837323

36	2.408958	2.388213	1.418329	1.628627	1.823923	1.855356
37	2.416086	2.446995	1.381755	1.579058	1.811664	1.839731
38	2.419935	2.450039	1.359364	1.528949	1.842990	1.818611
39	2.406924	2.420356	1.406839	1.612392	1.820673	1.862236
40	2.411082	2.439522	1.386057	1.588262	1.806479	1.848654
41	2.445971	2.432676	1.414754	1.599544	1.852943	1.840840
42	2.456986	2.400088	1.383864	1.546704	1.838615	1.858656
43	2.424771	2.458291	1.428424	1.627727	1.836889	1.867144
44	2.447868	2.386386	1.416142	1.550924	1.802053	1.843765
45	2.440500	2.406939	1.419125	1.588304	1.810358	1.855754
46	2.458035	2.402882	1.418004	1.632737	1.830199	1.841494
47	2.453044	2.400010	1.386885	1.553212	1.839634	1.879314
48	2.457722	2.407752	1.389539	1.566726	1.848899	1.867521
49	2.471141	2.435757	1.423197	1.551818	1.865658	1.875485
50	2.439184	2.415828	1.423496	1.618629	1.847202	1.879861

Table 33: Control output $\dot{\theta}$: spiral trajectory with $(\ln 2)^2$ of the best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 60.057$	$k = 60.057$	$q = 129.722, q = 0.01$		$\gamma_T = 0.775, \gamma_O = 0.786$	
1	301.831309	303.888108	340.038152	344.579206	325.695692	325.631568
2	299.164906	303.611655	344.364852	341.350026	325.724756	322.306218
3	303.346257	308.341957	344.061729	357.814094	322.689472	326.644372
4	317.485567	316.200278	351.189683	357.540699	340.017436	341.928506
5	321.947317	320.655881	364.364225	370.272867	344.982355	343.192869
6	316.947215	317.048749	363.825415	371.675872	340.301521	342.911280
7	321.846266	321.297965	365.443377	374.058422	338.311795	345.230788
8	324.553389	322.589626	358.642535	358.562718	339.732892	342.320637
9	320.513020	326.284558	366.211586	357.260440	343.413174	343.025712
10	322.638653	319.753172	364.866678	357.404840	336.239070	340.345389
11	324.618920	321.326668	364.135756	369.424426	343.358851	342.279150
12	319.934235	315.091731	355.535123	349.588076	335.702520	338.411136
13	318.175993	316.063919	356.987159	355.505297	338.227871	336.567911
14	316.562878	315.949761	356.510170	365.231224	337.149632	339.107579
15	318.354478	317.272534	350.417694	366.487516	338.508300	340.126947
16	319.286666	321.688764	348.885694	353.510913	338.761394	339.642087
17	318.833534	319.760516	348.158235	355.185714	337.469849	338.380851
18	317.804359	320.508053	359.159074	351.746044	338.466337	337.850004
19	320.825644	321.214202	362.126532	370.886061	335.691965	336.559893
20	315.907598	316.520399	356.425062	351.443180	336.683807	338.005560
21	312.966016	315.844011	356.369226	367.337786	339.662038	338.976347
22	315.892842	319.014557	357.071512	352.680882	337.140689	336.937425
23	317.602050	318.782164	354.641309	349.810987	340.333889	335.303331
24	317.699903	315.402352	352.789599	346.755260	336.564870	336.783117
25	317.409533	317.278373	344.764826	346.764288	336.675951	332.874910
26	318.379852	318.198660	354.600888	367.008139	337.714665	336.421631
27	318.238802	315.176802	351.666129	349.042630	335.879842	331.664057
28	317.007203	318.280546	349.923661	347.375357	333.624415	333.947942
29	314.666710	315.866072	350.240779	347.776568	332.702165	331.496844
30	317.371665	317.782700	354.932377	356.604282	332.855813	336.183651
31	315.089653	313.101180	343.675814	353.024920	330.518331	335.070488
32	316.266978	315.227034	348.420089	349.036212	330.099563	333.141196
33	312.033803	313.526256	351.953149	359.052884	334.255113	335.426339
34	314.899367	314.881186	351.603838	361.127873	331.934471	332.769722
35	313.895029	314.293592	352.566331	354.630556	331.320963	331.999119
36	314.050317	311.620555	353.218890	365.108637	332.492716	334.412294

37	314.482693	317.962587	347.115012	357.178915	332.610304	333.354494
38	315.095527	317.465602	342.432102	347.249484	333.334135	329.641857
39	313.796943	315.140044	351.332682	362.714448	333.020914	335.963554
40	314.336977	317.058243	347.809695	358.472480	331.233161	334.825085
41	317.428153	316.627209	353.413922	360.436268	335.613934	334.237908
42	317.781367	313.006775	348.825805	349.425636	335.346774	335.031977
43	315.931657	317.899112	356.726768	365.831164	333.604885	336.054190
44	317.970948	312.588073	354.058957	350.027054	330.479592	334.083113
45	316.696922	313.785046	355.558079	358.600993	330.938914	336.006018
46	319.786335	314.271595	355.517274	367.407655	333.071533	333.867210
47	318.275240	313.074622	348.359290	351.419722	334.435423	338.519215
48	318.473263	314.573481	349.789231	353.940566	336.282911	337.398048
49	320.641948	316.352559	355.839741	351.017992	337.359786	337.855598
50	316.506784	314.963178	354.660531	364.387835	335.218110	337.651079

Table 35: Joints Control Effort Norm (Eff.): spiral trajectory with $(\ln 2)^2$ of the best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 60.057$	$k = 60.057$	$q = 129.722, q = 0.01$		$\gamma_T = 0.775, \gamma_O = 0.786$	
1	0.054672	0.054604	0.059250	0.061244	0.056990	0.056747
2	0.053104	0.054645	0.060387	0.059912	0.056537	0.056216
3	0.054004	0.053771	0.060581	0.063406	0.057656	0.057422
4	0.053713	0.054072	0.058643	0.059181	0.056455	0.056847
5	0.053620	0.053838	0.061034	0.063138	0.057234	0.056806
6	0.053858	0.054266	0.060767	0.062781	0.057056	0.056800
7	0.054291	0.053995	0.060061	0.062694	0.057089	0.056939
8	0.053541	0.054010	0.059058	0.059462	0.056882	0.056700
9	0.053981	0.053585	0.061855	0.059836	0.056488	0.057037
10	0.053831	0.053792	0.061453	0.059596	0.056871	0.057062
11	0.053702	0.054412	0.061356	0.062987	0.056692	0.057361
12	0.053989	0.054379	0.059526	0.059047	0.056586	0.055970
13	0.053599	0.054029	0.060379	0.060061	0.056158	0.056507
14	0.053605	0.053927	0.059582	0.062203	0.056577	0.056200
15	0.054173	0.054631	0.059037	0.062129	0.056957	0.056472
16	0.053100	0.053830	0.058724	0.059379	0.056091	0.056594
17	0.054629	0.054413	0.058689	0.059611	0.057397	0.056623
18	0.054224	0.053854	0.061206	0.059961	0.056634	0.057106
19	0.054497	0.054089	0.061125	0.063030	0.056690	0.057164
20	0.053860	0.054642	0.060207	0.059809	0.057197	0.056628
21	0.054911	0.054736	0.060619	0.063201	0.057168	0.057275
22	0.054904	0.053786	0.061125	0.059798	0.056613	0.057034
23	0.054877	0.055013	0.060054	0.059474	0.057048	0.057046
24	0.054424	0.054759	0.060480	0.059460	0.056375	0.056825
25	0.054353	0.054262	0.058914	0.059478	0.057592	0.057313
26	0.054704	0.055155	0.060926	0.062573	0.056988	0.056935
27	0.054171	0.055057	0.059922	0.059576	0.057515	0.057304
28	0.053797	0.054238	0.060081	0.059361	0.057072	0.057342
29	0.053897	0.054073	0.059609	0.059558	0.057208	0.056823
30	0.053988	0.054596	0.061233	0.060764	0.057205	0.056949
31	0.054535	0.053395	0.059019	0.060848	0.057683	0.057106
32	0.054095	0.053674	0.059885	0.059706	0.056450	0.057846
33	0.053754	0.053922	0.060945	0.061889	0.056859	0.056947
34	0.053505	0.053601	0.060573	0.061234	0.056534	0.057464
35	0.054016	0.053880	0.060314	0.059994	0.057048	0.057702

36	0.053680	0.053534	0.060273	0.062626	0.058307	0.057368
37	0.054136	0.053397	0.059049	0.060751	0.056825	0.056441
38	0.054244	0.054667	0.058684	0.059371	0.056692	0.056929
39	0.054316	0.053447	0.060123	0.062026	0.056667	0.056736
40	0.053410	0.053812	0.059185	0.060756	0.056186	0.057010
41	0.054499	0.053414	0.060694	0.060274	0.057370	0.057017
42	0.053785	0.053752	0.059006	0.059764	0.056592	0.057347
43	0.054034	0.054238	0.060378	0.061802	0.057267	0.056770
44	0.053972	0.054518	0.060095	0.059174	0.057064	0.056823
45	0.053830	0.054687	0.059667	0.060676	0.056767	0.056874
46	0.055125	0.054541	0.060570	0.062666	0.056469	0.056629
47	0.053886	0.054462	0.059454	0.059439	0.057180	0.056924
48	0.053731	0.054580	0.059281	0.059969	0.057030	0.057325
49	0.054579	0.054437	0.060096	0.058670	0.057376	0.056771
50	0.053592	0.054470	0.060190	0.062089	0.056592	0.056473

Table 37: End effector linear velocity (L. Vel.): spiral trajectory with $(\ln 2)^2$ of the best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 60.057$	$k = 60.057$	$q = 129.722, q = 0.01$		$\gamma_T = 0.775, \gamma_O = 0.786$	
1	1.341699	1.305637	1.393111	1.431237	1.331473	1.342722
2	1.318170	1.331367	1.414202	1.403089	1.336744	1.338513
3	1.331697	1.320703	1.421194	1.483432	1.358710	1.351949
4	1.323423	1.327943	1.394555	1.413735	1.341484	1.348060
5	1.326998	1.323756	1.439126	1.492063	1.355989	1.346906
6	1.319980	1.328213	1.438134	1.485218	1.349170	1.346821
7	1.337307	1.328946	1.424902	1.476490	1.347358	1.345541
8	1.326241	1.332013	1.392962	1.412110	1.352786	1.349665
9	1.331180	1.326715	1.458090	1.422088	1.339999	1.352622
10	1.327177	1.329151	1.440944	1.412146	1.348557	1.349172
11	1.327112	1.323543	1.447618	1.488122	1.340893	1.349786
12	1.326738	1.336160	1.426626	1.409338	1.348745	1.342584
13	1.340489	1.333488	1.445301	1.436166	1.341514	1.351869
14	1.323052	1.327467	1.422659	1.481465	1.348167	1.348647
15	1.331968	1.340644	1.408928	1.483639	1.348031	1.344689
16	1.313664	1.321497	1.391523	1.416179	1.334904	1.350807
17	1.335430	1.338266	1.386394	1.415924	1.354140	1.351755
18	1.328722	1.317792	1.446326	1.420482	1.349315	1.348753
19	1.338490	1.326979	1.452791	1.495375	1.346067	1.354469
20	1.317522	1.330567	1.433501	1.421566	1.356068	1.345705
21	1.339802	1.333225	1.437543	1.492925	1.352505	1.353903
22	1.337875	1.317809	1.454353	1.420564	1.347050	1.354696
23	1.317542	1.339109	1.433064	1.417365	1.348711	1.360155
24	1.331897	1.340363	1.435376	1.417803	1.339077	1.346364
25	1.320800	1.319862	1.398552	1.419604	1.354490	1.352973
26	1.333976	1.343486	1.438040	1.483227	1.344148	1.348367
27	1.333795	1.337332	1.422442	1.408675	1.353100	1.354603
28	1.326486	1.331972	1.424067	1.411920	1.350755	1.353973
29	1.319791	1.333281	1.413130	1.416165	1.349333	1.344527
30	1.329408	1.335748	1.451629	1.447665	1.351382	1.350084
31	1.336738	1.314978	1.391948	1.451911	1.350769	1.351615
32	1.321199	1.320395	1.419715	1.417744	1.342781	1.361199
33	1.323475	1.319837	1.446053	1.472725	1.344801	1.347260
34	1.313568	1.317588	1.431805	1.464305	1.337598	1.355004
35	1.322919	1.324223	1.431796	1.430285	1.351581	1.363299

36	1.314975	1.315662	1.428241	1.481358	1.353841	1.362679
37	1.327988	1.316722	1.395935	1.453971	1.352557	1.344309
38	1.315531	1.336797	1.397014	1.413426	1.345880	1.346670
39	1.336838	1.320597	1.425171	1.477243	1.350012	1.351808
40	1.314226	1.322120	1.410427	1.450214	1.347757	1.353205
41	1.330004	1.315769	1.435631	1.444544	1.358679	1.345551
42	1.320950	1.324724	1.397110	1.426854	1.343067	1.360327
43	1.333914	1.338835	1.439611	1.473846	1.349997	1.341336
44	1.326295	1.328601	1.427500	1.406588	1.350005	1.348322
45	1.332858	1.340893	1.425044	1.447993	1.351168	1.349184
46	1.347449	1.338720	1.435882	1.482118	1.346209	1.349008
47	1.320865	1.334210	1.413863	1.413454	1.352649	1.350150
48	1.320893	1.326705	1.409293	1.419621	1.359158	1.353770
49	1.336743	1.331088	1.420851	1.404891	1.367163	1.346995
50	1.315774	1.325998	1.422668	1.476849	1.340023	1.337617

Table 39: End effector linear acceleration (L. Acc.): spiral trajectory with $(\ln 2)^2$ of the best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 60.057$	$k = 60.057$	$q = 129.722, q = 0.01$		$\gamma_T = 0.775, \gamma_O = 0.786$	
1	136.722495	112.915369	116.472430	111.792719	113.361807	121.829930
2	112.529986	123.097703	112.058561	113.836398	112.484560	114.401182
3	139.556628	113.016511	113.663961	103.968905	123.351983	126.528919
4	118.550489	118.586052	119.575571	114.279821	114.676808	115.744390
5	120.109350	116.424587	113.873308	102.453630	121.505215	118.158380
6	112.765424	114.258047	112.099857	103.813183	122.832160	114.806881
7	126.460798	128.165798	113.817964	103.431745	122.505782	116.061916
8	116.057200	131.099061	120.007168	116.125489	122.603777	117.855555
9	119.335525	118.196782	108.672626	117.742628	124.737281	121.556094
10	118.607951	116.311993	111.133852	116.546949	121.440276	118.443926
11	119.168811	123.886083	109.686915	104.256713	114.729915	116.041003
12	119.870170	127.922900	111.965487	113.337593	112.458744	114.597270
13	148.599693	143.155313	108.623100	109.402633	113.673479	113.630397
14	113.734416	125.238477	114.278602	102.105724	125.408016	113.542051
15	125.832933	143.940072	115.353891	103.507195	112.276907	114.318399
16	113.695453	113.626069	122.540311	114.133635	120.818769	120.483499
17	112.358637	117.256303	119.922369	115.746450	112.514133	114.849416
18	111.830539	112.909032	109.162092	114.010036	142.199671	113.198019
19	129.371428	114.082149	108.151796	102.612090	116.936876	118.300567
20	113.696797	132.945409	112.535812	114.992538	117.695998	114.983364
21	112.423690	121.119794	112.043557	102.758563	128.191478	113.567773
22	124.478655	113.040801	110.643562	113.820594	112.519960	127.690075
23	114.233806	114.969642	111.495069	114.425465	113.105000	116.018857
24	113.747766	120.727757	113.605209	113.304493	113.873567	114.529877
25	113.872359	116.009073	117.474887	114.287897	114.028708	130.760021
26	140.867457	136.609958	112.109584	101.997690	118.560641	113.582895
27	114.974742	119.322002	113.345366	114.747811	130.247633	115.371494
28	121.975595	115.354576	115.196112	114.594122	138.027968	117.073514
29	114.272641	137.725975	114.394413	115.177786	117.476375	114.135328
30	124.829006	114.246945	108.942249	108.319871	128.193120	114.034314
31	132.726231	112.695402	120.300522	107.730931	125.104080	117.950560
32	118.556282	111.566580	117.970877	113.552320	122.370059	113.625540
33	112.715786	112.656996	113.480466	105.871957	114.387462	123.757254
34	113.214256	111.992092	111.407354	106.402854	113.046168	115.584432
35	114.252194	124.356944	113.223708	112.146548	116.995647	119.223372

36	112.441094	111.724809	112.047826	103.881360	123.190047	118.739643
37	122.232731	117.334775	123.352819	107.253805	113.710055	118.534489
38	112.595789	114.980168	121.816379	115.171647	116.930040	117.338779
39	128.026434	114.188056	112.168014	104.157435	114.325728	116.152058
40	112.975241	122.503847	120.174439	108.118756	129.662988	118.373673
41	123.375529	112.798243	110.690184	110.972723	116.183225	114.093219
42	112.618016	127.395961	121.319221	117.389459	119.188913	122.715195
43	112.786071	130.439900	110.137912	103.185686	125.142529	113.780290
44	117.867908	117.937120	113.264110	114.167396	118.152474	115.974706
45	130.444770	132.773474	113.843882	109.713448	114.403982	121.613270
46	149.691062	143.484997	110.040549	103.848163	117.841308	115.437246
47	117.471902	128.765650	115.588464	115.091256	128.605383	115.031782
48	114.127998	118.116707	119.353615	113.014374	127.733213	134.518389
49	120.121548	125.776122	114.053687	115.773673	124.374475	114.784843
50	113.844858	117.094961	114.659910	105.076670	119.126476	114.810904

E.6.2 $(\ln 2)$ of the best performance

Table 41: Integral norm of the end-effector invariant error: spiral trajectory with $(\ln 2)$ of the best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 86.643$	$k = 86.643$	$q = 187.15, r = 0.01$		$\gamma_T = 0.537, \gamma_O = 0.544$	
1	1.642340	1.624950	1.592690	1.242188	1.225588	1.234155
2	1.658781	1.654525	1.624520	1.299189	1.242725	1.220993
3	1.655785	1.652415	1.500473	1.230629	1.235760	1.209952
4	1.639596	1.638225	1.565242	1.289772	1.229953	1.224240
5	1.645112	1.640259	1.614849	1.239090	1.230657	1.247814
6	1.663297	1.656902	1.549416	1.305277	1.239888	1.241181
7	1.647812	1.682431	1.586609	1.267538	1.233262	1.241091
8	1.634806	1.651401	1.568970	1.305862	1.214803	1.224710
9	1.661692	1.677400	1.527333	1.271085	1.234054	1.217828
10	1.662121	1.654896	1.601174	1.311825	1.233444	1.214569
11	1.662385	1.664383	1.494742	1.278014	1.214390	1.210791
12	1.650396	1.637147	1.466418	1.321474	1.233505	1.232203
13	1.641074	1.663595	1.637975	1.231670	1.232178	1.232529
14	1.667053	1.681249	1.567856	1.315631	1.222219	1.221977
15	1.657144	1.662774	1.478920	1.312048	1.244512	1.212514
16	1.633083	1.661042	1.667855	1.250214	1.226317	1.222995
17	1.664593	1.672101	1.483760	1.321851	1.226869	1.197773
18	1.655154	1.650095	1.526778	1.250697	1.222528	1.217682
19	1.660464	1.646678	1.600142	1.256320	1.215143	1.213629
20	1.635076	1.633025	1.544805	1.275697	1.210002	1.230358
21	1.658400	1.697624	1.622549	1.249671	1.221014	1.213877
22	1.635585	1.647796	1.492008	1.262698	1.217713	1.210993
23	1.634249	1.667149	1.535721	1.301478	1.215229	1.189586
24	1.630181	1.648881	1.587741	1.269677	1.232490	1.221631
25	1.659811	1.652169	1.618211	1.332885	1.234206	1.239158
26	1.668167	1.668272	1.554910	1.327571	1.248335	1.234199
27	1.694915	1.693814	1.527497	1.347641	1.232509	1.246358
28	1.674534	1.672045	1.514748	1.264452	1.253518	1.245941
29	1.707578	1.686485	1.620558	1.245554	1.275310	1.271968
30	1.708723	1.704679	1.581515	1.333845	1.252633	1.258441
31	1.683925	1.708281	1.553604	1.264979	1.254799	1.254404
32	1.709160	1.718176	1.547648	1.280782	1.293072	1.275432
33	1.731421	1.686914	1.535927	1.294704	1.260626	1.281891
34	1.700857	1.695341	1.602126	1.309251	1.253044	1.244139
35	1.710104	1.710514	1.488326	1.324105	1.235051	1.242807

36	1.692439	1.726115	1.576584	1.333599	1.255120	1.240404
37	1.708452	1.727059	1.617139	1.264244	1.259438	1.260450
38	1.712671	1.714158	1.551895	1.270974	1.256861	1.257652
39	1.718356	1.696913	1.361212	1.253407	1.243989	1.268518
40	1.704213	1.732264	1.569839	1.313940	1.265622	1.266963
41	1.730273	1.735843	1.510208	1.306047	1.290407	1.283073
42	1.715046	1.756487	1.662381	1.253039	1.264555	1.274731
43	1.719967	1.755595	1.578794	1.326214	1.280311	1.275405
44	1.742992	1.753077	1.527813	1.250092	1.266309	1.268828
45	1.728029	1.746995	1.573755	1.332084	1.280945	1.262776
46	1.742656	1.737488	1.647273	1.243685	1.271860	1.273593
47	1.738552	1.737942	1.627506	1.271189	1.260094	1.268215
48	1.713719	1.742821	1.567420	1.330253	1.259129	1.266383
49	1.717275	1.746780	1.588891	1.320317	1.266504	1.257098
50	1.694722	1.711057	1.657662	1.338221	1.265828	1.270645

Table 43: Control output $\dot{\theta}$: spiral trajectory with $(\ln 2)$ of the best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 86.643$	$k = 86.643$	$q = 187.15, r = 0.01$		$\gamma_T = 0.537, \gamma_O = 0.544$	
1	324.125712	321.174490	471.081478	347.307077	336.004809	338.277347
2	325.633080	325.270997	479.326230	359.191082	340.645068	334.844807
3	324.636168	325.585353	449.468107	344.575965	337.829612	334.398537
4	322.998019	321.142248	463.147029	358.749408	334.873903	335.956501
5	325.114103	322.775840	479.041296	345.607474	335.786529	338.136245
6	325.062607	324.389306	462.733884	359.679140	338.503527	337.046082
7	325.090857	329.487739	468.627397	352.171751	335.876268	337.061077
8	321.039016	324.501430	466.995774	360.941357	332.289588	334.910075
9	326.121476	328.371441	456.164743	355.893057	336.323063	331.933581
10	326.500411	324.901300	472.591671	363.059805	337.282969	331.191227
11	324.664500	324.880018	447.055921	353.300178	332.182224	331.397384
12	323.134249	321.063627	433.852865	363.331115	336.662248	334.760130
13	322.009388	324.700559	480.283767	342.983997	336.264458	335.152456
14	325.857835	328.045087	466.712531	362.661532	332.043944	332.284791
15	323.515602	326.224592	443.412144	361.385343	337.434189	330.209816
16	321.429996	325.589510	475.706147	346.799231	336.395317	333.678270
17	325.487663	327.033588	449.708569	363.886594	334.101219	327.878886
18	323.524159	323.723116	453.053852	345.290359	334.592437	331.785032
19	323.620861	321.696494	471.647953	347.019457	333.248979	330.855173
20	320.585883	321.069771	459.004224	354.250448	329.652349	333.445238
21	323.638978	329.939171	469.337877	345.921895	334.566897	332.821799
22	320.743189	322.261612	443.041805	349.306465	333.938249	331.287880
23	321.100287	325.094919	455.268083	359.019825	333.298662	326.772961
24	320.847415	323.397823	465.782046	351.803290	337.476848	333.453639
25	325.211259	323.720876	456.959628	365.431438	337.123753	336.684524
26	326.647246	325.640345	458.970622	364.675408	339.089047	335.142242
27	327.559357	328.393918	450.758370	370.732175	336.260735	337.091939
28	325.144410	325.150263	448.305723	349.006307	337.939519	336.413183
29	330.335249	327.312786	472.914974	344.349292	341.586344	339.986394
30	329.935830	330.573550	466.117691	363.970682	337.930877	337.862543
31	326.324052	328.737842	459.116265	349.299231	337.352794	336.611459
32	329.024805	330.575738	461.170350	354.722689	342.335482	338.679321
33	330.337534	325.707450	452.049668	356.759459	336.439079	340.169218
34	330.769991	330.035912	473.547983	361.083487	340.259850	338.486770
35	331.125286	331.589876	443.383558	362.722621	336.797360	337.682121
36	329.181004	333.693380	464.005561	365.468135	339.396728	335.991432

37	330.569338	333.427680	473.197863	348.999184	339.774762	340.369601
38	331.119625	331.723854	461.514715	353.358728	339.345070	338.552240
39	330.655008	329.141349	368.623440	347.256763	336.706683	340.697699
40	329.720389	333.935672	462.959590	362.923989	340.659182	339.443970
41	332.271827	333.055565	448.998749	361.038386	344.442081	343.323160
42	329.325540	335.834690	485.365415	347.684525	339.421476	341.184706
43	330.913511	336.429017	462.855549	364.029574	341.647099	341.156301
44	334.479341	335.115064	451.916026	349.020110	339.136719	339.060102
45	331.943794	334.320800	462.286331	364.808964	340.521134	339.170108
46	333.989392	333.337343	480.024904	345.299584	340.571565	339.982034
47	334.098987	333.723410	476.506140	351.954655	338.195210	339.314363
48	330.218327	334.022285	462.759821	365.251958	337.331033	337.641014
49	329.673002	334.298955	465.134990	361.958717	339.037293	337.116699
50	326.497691	329.300617	481.369168	391.291790	338.904837	339.796764

Table 45: Joints Control Effort Norm (Eff.): spiral trajectory with ($\ln 2$) of the best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 86.643$	$k = 86.643$	$q = 187.15, r = 0.01$		$\gamma_T = 0.537, \gamma_O = 0.544$	
1	0.057666	0.057302	0.057838	0.062215	0.061487	0.060673
2	0.058087	0.057691	0.059321	0.064642	0.060554	0.060054
3	0.058062	0.058546	0.059980	0.062172	0.061186	0.060795
4	0.058665	0.058097	0.057942	0.065132	0.060754	0.060616
5	0.059198	0.058685	0.059764	0.061660	0.061095	0.060610
6	0.059196	0.058466	0.058366	0.064325	0.061131	0.061066
7	0.058528	0.058932	0.058377	0.063155	0.060722	0.061141
8	0.058416	0.057898	0.059610	0.065298	0.060943	0.061034
9	0.058266	0.059403	0.059780	0.069077	0.061436	0.060370
10	0.058220	0.058167	0.059200	0.064776	0.060656	0.060803
11	0.058416	0.058017	0.059035	0.063093	0.060919	0.060872
12	0.058597	0.059263	0.060574	0.064010	0.061698	0.060603
13	0.058184	0.058321	0.059786	0.061394	0.061330	0.061074
14	0.058322	0.058818	0.058609	0.065055	0.060791	0.060719
15	0.058770	0.058709	0.060405	0.064658	0.061076	0.060733
16	0.058887	0.058429	0.065549	0.060784	0.061637	0.061071
17	0.057986	0.058297	0.059111	0.064965	0.060940	0.060706
18	0.058434	0.058372	0.057925	0.061028	0.061406	0.060791
19	0.058536	0.058603	0.058826	0.061350	0.061275	0.060254
20	0.058891	0.059061	0.058132	0.062654	0.060385	0.060794
21	0.057797	0.059065	0.059702	0.061508	0.061115	0.060917
22	0.058128	0.058523	0.058477	0.061699	0.061087	0.060232
23	0.058683	0.058472	0.058467	0.064443	0.060770	0.060012
24	0.057192	0.057327	0.060624	0.062080	0.060256	0.059132
25	0.057455	0.057112	0.064626	0.065091	0.059851	0.060119
26	0.057487	0.057617	0.059577	0.065111	0.060005	0.059385
27	0.057501	0.057449	0.057531	0.065784	0.059939	0.059560
28	0.058057	0.058048	0.057788	0.062030	0.059326	0.059536
29	0.058068	0.057842	0.057593	0.061367	0.060288	0.059976
30	0.057878	0.058275	0.058437	0.065057	0.059860	0.060088
31	0.057957	0.058422	0.058860	0.061762	0.060376	0.059766
32	0.057744	0.058076	0.058252	0.062690	0.060316	0.060344
33	0.057489	0.058110	0.060286	0.063329	0.060231	0.060258
34	0.057434	0.057484	0.059294	0.064670	0.058963	0.058855
35	0.056652	0.056904	0.058609	0.064257	0.059191	0.059132

36	0.058082	0.057092	0.058604	0.064668	0.059434	0.059068
37	0.056896	0.057247	0.058633	0.061651	0.058827	0.059298
38	0.057117	0.056867	0.059212	0.062457	0.060036	0.059221
39	0.057844	0.057982	0.060906	0.060488	0.059108	0.059579
40	0.057938	0.057565	0.058827	0.064676	0.059485	0.059833
41	0.057031	0.057582	0.058559	0.064882	0.060604	0.060829
42	0.057199	0.058421	0.058773	0.061789	0.059173	0.059208
43	0.057397	0.057798	0.059950	0.064956	0.059850	0.059531
44	0.057775	0.058266	0.060116	0.062385	0.060369	0.059547
45	0.058412	0.057461	0.059554	0.064347	0.059175	0.059599
46	0.058346	0.057927	0.058994	0.060897	0.060543	0.059199
47	0.058638	0.057380	0.058766	0.061991	0.059571	0.059439
48	0.058130	0.058363	0.058378	0.065101	0.059856	0.059504
49	0.057187	0.057792	0.058585	0.063922	0.059348	0.059542
50	0.057763	0.058295	0.058255	0.074778	0.059861	0.059568

Table 47: End effector linear velocity (L. Vel.): spiral trajectory with ($\ln 2$) of the best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 86.643$	$k = 86.643$	$q = 187.15, r = 0.01$		$\gamma_T = 0.537, \gamma_O = 0.544$	
1	1.372280	1.366343	1.382241	1.448676	1.394162	1.388639
2	1.369587	1.367986	1.394241	1.498064	1.386156	1.378776
3	1.379410	1.380213	1.402642	1.429456	1.387862	1.391873
4	1.377900	1.374399	1.374273	1.505276	1.388959	1.391262
5	1.387642	1.382898	1.408181	1.438174	1.391368	1.384837
6	1.397145	1.378569	1.382704	1.493141	1.388556	1.394383
7	1.381069	1.386052	1.384162	1.465446	1.383113	1.394314
8	1.381808	1.369456	1.416185	1.509851	1.393141	1.390036
9	1.372325	1.386281	1.399754	1.661216	1.390251	1.389671
10	1.374213	1.373223	1.407191	1.504687	1.383605	1.387912
11	1.375809	1.370845	1.398357	1.468729	1.393291	1.389300
12	1.385421	1.384221	1.414836	1.499941	1.395093	1.384582
13	1.374327	1.377565	1.408867	1.432004	1.389428	1.388630
14	1.372712	1.387383	1.396441	1.507554	1.391335	1.389610
15	1.383001	1.387448	1.415554	1.502334	1.392932	1.394005
16	1.388504	1.371693	1.610323	1.420066	1.387872	1.384536
17	1.368124	1.372199	1.390603	1.509068	1.392611	1.392360
18	1.372545	1.368942	1.381228	1.422099	1.388033	1.389351
19	1.381358	1.383493	1.402736	1.421302	1.390150	1.387093
20	1.390359	1.371443	1.378470	1.457898	1.386941	1.386658
21	1.367278	1.386795	1.406968	1.426394	1.391484	1.394614
22	1.371892	1.379682	1.395829	1.427664	1.394680	1.382102
23	1.382916	1.376151	1.388629	1.497360	1.383974	1.387902
24	1.371901	1.370223	1.428034	1.449467	1.384719	1.384832
25	1.369159	1.372187	1.594276	1.509649	1.381914	1.390101
26	1.373142	1.373027	1.421438	1.515321	1.381772	1.388212
27	1.369168	1.373448	1.356462	1.515726	1.381960	1.390911
28	1.374577	1.373912	1.372921	1.437299	1.380081	1.379397
29	1.380430	1.376261	1.376267	1.421127	1.382238	1.388781
30	1.381337	1.380453	1.387303	1.495836	1.385729	1.389081
31	1.372892	1.382455	1.392777	1.430710	1.386276	1.384849
32	1.377268	1.376218	1.388397	1.455828	1.392469	1.385805
33	1.371059	1.374531	1.411706	1.466454	1.386318	1.386049
34	1.379058	1.375304	1.413198	1.501672	1.370187	1.374902
35	1.363436	1.370364	1.389284	1.494290	1.384488	1.384480

36	1.389344	1.374435	1.404050	1.506909	1.383422	1.382341
37	1.369642	1.374999	1.396950	1.430149	1.371444	1.391830
38	1.369675	1.366915	1.405233	1.455761	1.394064	1.385493
39	1.389585	1.383567	1.442179	1.425608	1.380428	1.390746
40	1.387665	1.369793	1.394071	1.505522	1.385651	1.387067
41	1.364272	1.367581	1.393647	1.505259	1.406457	1.413632
42	1.367597	1.391144	1.398988	1.428475	1.380042	1.379017
43	1.373965	1.368551	1.410593	1.507842	1.385793	1.378091
44	1.373225	1.382430	1.409439	1.455227	1.393772	1.387332
45	1.391915	1.367641	1.401581	1.502445	1.379447	1.384018
46	1.392969	1.385955	1.401865	1.412585	1.406453	1.383016
47	1.391626	1.366529	1.388467	1.444239	1.389028	1.397153
48	1.385858	1.385082	1.384012	1.503955	1.397230	1.379417
49	1.368389	1.376652	1.387596	1.479872	1.382322	1.384712
50	1.381956	1.386610	1.394446	1.793818	1.393278	1.391449

Table 49: End effector linear acceleration (L. Acc.): spiral trajectory with $(\ln 2)$ of the best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 86.643$	$k = 86.643$	$q = 187.15, r = 0.01$		$\gamma_T = 0.537, \gamma_O = 0.544$	
1	120.358257	116.495185	116.315053	112.236567	129.070419	119.952939
2	122.209094	115.608521	109.268361	104.096984	123.078630	116.293444
3	126.826828	129.744602	109.255588	115.241832	124.086056	118.330744
4	126.111283	123.114353	117.404811	103.454329	123.583873	123.630447
5	127.996869	126.611870	113.489761	118.107188	124.774646	117.867978
6	152.162435	125.581312	113.004954	105.518988	124.622552	122.434243
7	126.068157	126.191897	112.708435	108.804851	124.455597	122.618680
8	126.765343	120.150710	113.536241	103.294817	128.470193	120.184766
9	120.363838	119.532056	109.901418	131.270655	127.099878	118.051165
10	121.621622	121.536845	109.813281	104.784080	117.838910	119.684975
11	122.284823	118.345066	112.830229	110.530872	124.360640	118.906585
12	130.424284	131.628487	113.432057	105.669049	122.647886	118.536014
13	116.743626	119.778886	108.251698	116.650945	125.357329	121.306081
14	125.718668	131.080090	109.148826	104.338528	130.836735	122.338846
15	124.308932	127.985378	108.615141	104.495388	123.411067	119.250898
16	134.934135	127.376210	134.096194	116.262664	122.087650	118.716371
17	119.601046	118.646692	112.579648	103.672927	118.388135	120.601361
18	117.826128	115.686544	112.618597	113.933470	128.196819	118.954400
19	128.998472	136.236262	109.039223	116.072711	126.179986	119.975160
20	129.230756	122.783188	115.206378	112.155605	121.382132	119.535304
21	118.896609	143.296356	110.250720	115.435205	123.197102	117.995937
22	119.442001	119.526650	111.909381	115.713674	122.938195	117.898712
23	130.056088	120.965935	115.485773	104.843386	121.456365	117.609181
24	118.363567	115.682016	108.451065	117.385987	125.974330	120.894995
25	122.922963	119.809768	127.531282	103.784991	117.334067	120.175405
26	118.164088	123.657181	109.600975	101.991056	120.687671	118.045316
27	121.058613	118.800473	117.183208	104.001518	117.351552	117.537494
28	118.529964	116.318166	113.357005	114.700041	121.431316	119.077055
29	127.033114	123.536554	112.491031	116.337243	121.762345	126.576807
30	131.283001	123.676064	113.259813	105.847710	119.157611	121.563685
31	125.807494	124.969010	111.330354	115.067071	122.816124	122.118329
32	116.940818	116.046998	116.136685	110.317997	119.529835	117.556507
33	116.798056	116.569223	109.021467	108.929777	122.021855	120.185615
34	135.717714	117.350306	109.283377	104.612561	117.371599	117.452031
35	115.609029	121.282582	113.589634	104.732733	124.333945	115.782383

36	135.887198	116.253578	109.957233	105.214894	125.392817	116.982782
37	122.039960	128.473525	110.302993	115.790777	119.349217	118.701603
38	122.189733	116.360102	108.269381	111.210137	116.971761	117.311591
39	119.323575	130.329695	111.606811	116.073044	125.999135	115.631464
40	132.588100	115.960071	113.752931	104.843406	119.450352	118.048993
41	118.126940	121.077280	111.156474	105.472450	113.783877	112.862456
42	116.124034	128.890267	111.791086	117.157273	117.076933	117.869985
43	116.133181	129.415425	107.878246	104.113028	130.870458	117.221227
44	125.614057	135.991289	108.731724	118.108835	138.083498	118.274122
45	133.304128	118.887046	111.231332	104.621821	117.407930	117.688456
46	148.708879	136.173300	111.845729	116.633684	128.531303	117.939213
47	126.554937	118.114292	113.336804	114.498360	115.709432	117.639896
48	131.082780	120.576771	114.978858	104.229070	130.939037	115.853032
49	117.391633	115.808997	110.922403	108.089726	120.586136	116.584017
50	136.327194	130.158628	112.418040	118.697707	117.469010	123.911283

E.6.3 Best performance

Table 51: Integral norm of the end-effector invariant error: spiral trajectory with best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 125$	$k = 125$	$q = 270, r = 0.01$		$\gamma_T = 0.372, \gamma_O = 0.377$	
1	1.251944	1.266689	1.438943	1.080970	0.903847	0.883590
2	1.257981	1.239157	1.474702	1.045402	0.893807	0.884444
3	1.260923	1.256307	1.407573	1.089744	0.887880	0.889654
4	1.269366	1.262349	1.342663	1.073822	0.883148	0.889947
5	1.262249	1.265130	1.419202	1.096290	0.891667	0.892536
6	1.216475	1.204680	1.368524	1.099475	0.860610	0.872318
7	1.265416	1.262776	1.436994	1.046192	0.891586	0.894544
8	1.254832	1.264140	1.407571	1.049505	0.879434	0.894882
9	1.269964	1.254328	1.433464	1.057758	0.890817	0.889417
10	1.263980	1.257022	1.406917	1.109977	0.889129	0.883783
11	1.245168	1.242628	1.391207	1.048787	0.889365	0.883149
12	1.270405	1.258278	1.435381	1.070822	0.888391	0.888716
13	1.246787	1.269183	1.423027	1.038000	0.881735	0.883651
14	1.250410	1.259948	1.432970	1.053481	0.892365	0.900333
15	1.252844	1.264478	1.403575	1.060005	0.890369	0.887022
16	1.261573	1.272017	1.397854	1.052874	0.901413	0.900361
17	1.185118	1.190544	1.338520	1.053463	0.861563	0.861772
18	1.220269	1.215943	1.364627	1.051306	0.868279	0.866737
19	1.205292	1.214406	1.401654	1.055033	0.886555	0.874948
20	1.2211004	1.226239	1.392592	1.057197	0.867374	0.865999
21	1.220401	1.225942	1.378204	1.089117	0.878366	0.874906
22	1.231091	1.240790	1.403604	1.073659	0.874439	0.883415
23	1.234467	1.229870	1.394075	1.160858	0.881875	0.877612
24	1.282744	1.245632	1.411647	1.066141	0.883073	0.884725
25	1.237570	1.244533	1.417958	1.072680	0.884439	0.874024
26	1.244413	1.241331	1.415672	1.102566	0.875183	0.878154
27	1.235657	1.238679	1.440704	1.061885	0.885138	0.886400
28	1.239991	1.242944	1.438148	1.113333	0.881014	0.896990
29	1.174785	1.175406	1.330187	1.047421	0.845903	0.843224
30	1.168900	1.172865	1.362908	1.071343	0.839700	0.832522
31	1.183158	1.192815	1.324391	1.066273	0.841093	0.842502
32	1.161527	1.185674	1.343357	1.042865	0.831366	0.822875
33	1.167556	1.172882	1.350391	1.056686	0.843002	0.847946
34	1.181383	1.195462	1.355997	1.088165	0.836773	0.840232
35	1.185666	1.173476	1.346758	1.048374	0.826456	0.832541

36	1.160287	1.167764	1.352118	1.070773	0.845904	0.831231
37	1.175585	1.169207	1.340050	1.054456	0.836695	0.829797
38	1.151299	1.178921	1.348516	1.099207	0.836595	0.826728
39	1.244687	1.230496	1.390652	1.100186	0.878846	0.873938
40	1.234254	1.228544	1.420625	1.102737	0.878770	0.875011
41	1.246533	1.255873	1.412589	1.059565	0.899945	0.890009
42	1.249934	1.259683	1.500013	1.072292	0.891491	0.888811
43	1.261064	1.251995	1.427182	1.066036	0.888104	0.889444
44	1.252343	1.258387	1.352529	1.048111	0.889293	0.893131
45	1.262301	1.242321	1.387169	1.109051	0.891968	0.903682
46	1.256340	1.233721	1.389836	1.055747	0.898055	0.896759
47	1.262967	1.250640	1.413116	1.104990	0.893505	0.896651
48	1.255077	1.264276	1.344451	1.051505	0.890538	0.899584
49	1.252514	1.253670	1.428971	1.054980	0.896104	0.896884
50	1.254415	1.246210	1.383968	1.057639	0.895813	0.900120

Table 53: Control output $\dot{\theta}$: spiral trajectory with best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 125$	$k = 125$	$q = 270, r = 0.01$		$\gamma_T = 0.372, \gamma_O = 0.377$	
1	346.141097	349.827162	472.117425	360.542980	352.988637	349.591627
2	347.649966	343.374138	469.646245	349.942980	349.375311	347.585887
3	348.086415	347.210123	465.199294	361.149337	349.338273	349.318956
4	348.751907	347.765738	460.229435	357.444239	347.125513	348.354491
5	347.954896	347.542819	473.127921	364.113758	349.741099	349.574697
6	340.668289	335.987836	461.882791	363.042217	340.260472	344.576243
7	347.237313	347.734610	476.036291	347.019839	349.107385	350.723510
8	344.407920	347.659288	469.881549	349.760110	347.331939	351.009900
9	348.401628	347.006354	476.365293	351.697601	351.071211	350.136397
10	347.278238	345.595606	467.178819	364.329231	348.499898	347.244598
11	343.390961	342.260491	463.539171	348.964659	348.854445	346.922403
12	348.461695	345.864670	472.990528	353.906064	348.093300	347.667768
13	342.923919	349.020299	469.246036	345.422070	347.471833	347.977612
14	343.433183	346.034641	475.120224	349.250498	349.094051	352.086467
15	343.917747	345.859312	466.002651	351.601694	347.604817	347.851571
16	346.882949	348.190437	466.704736	349.049132	353.722862	352.634058
17	334.952297	334.838564	463.919201	347.894099	342.504316	342.870775
18	341.301062	339.679787	464.361600	348.332676	344.080218	343.839555
19	336.956225	338.460902	473.879267	350.666075	347.558061	346.260639
20	341.338817	342.105778	468.200966	350.046060	343.476698	343.439504
21	340.406104	341.572726	467.736885	359.028187	347.631998	344.325828
22	342.442341	344.589706	473.560595	354.596122	345.595318	348.464850
23	342.604513	343.924575	468.122874	381.279160	347.257700	346.613201
24	346.028719	345.239370	473.178641	351.587777	346.679211	346.994073
25	343.662583	344.482738	473.756438	355.593800	347.184951	345.589958
26	343.491327	343.117158	474.444269	364.260258	343.176794	344.354681
27	342.241315	343.611562	478.145893	350.782886	346.532847	348.434611
28	343.576980	342.958607	478.665579	364.453890	346.323920	349.263521
29	336.351511	335.299651	464.475001	346.181583	339.985684	338.281809
30	333.684510	334.554601	468.083854	352.961488	338.796809	335.656492
31	336.176267	338.038524	460.070447	351.876691	337.763204	340.054766
32	329.404132	335.273416	460.816272	345.575565	333.742656	332.344659
33	331.887116	332.708505	463.029620	349.238591	337.812773	338.364338
34	335.366753	337.274823	464.493288	357.752200	335.144155	337.003560
35	334.965939	333.160255	463.228418	345.642843	334.059510	335.304741
36	330.622399	331.998074	462.329757	354.703179	337.999079	333.663580

37	331.928587	331.329874	463.398281	347.869777	336.056560	334.291056
38	329.088168	334.622494	459.643691	360.658003	334.613848	331.824484
39	346.343431	342.280705	468.585538	361.497724	346.848939	346.252592
40	344.987987	341.957617	477.701526	361.307629	346.324324	346.609811
41	347.954495	349.879641	478.279126	349.834236	355.334033	353.057648
42	348.548821	350.809558	497.738523	352.716114	351.182688	351.722571
43	352.630795	349.287121	480.455566	349.656124	351.260822	351.649302
44	348.892860	349.054664	460.456318	346.598629	350.145101	351.564641
45	350.569861	347.199663	465.800769	362.262954	351.122710	354.952285
46	348.400494	344.318759	466.665590	348.293245	353.324259	352.207760
47	349.950343	348.131113	469.025616	361.992469	350.742946	352.396919
48	350.028915	350.588532	453.816815	347.814576	350.149086	352.953094
49	346.418399	348.867801	465.721442	349.132684	351.294517	352.031947
50	348.016141	347.515749	459.329390	348.765682	351.588003	352.655829

Table 55: Joints Control Effort Norm (Eff.): spiral trajectory with best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 125$	$k = 125$	$q = 270, r = 0.01$		$\gamma_T = 0.372, \gamma_O = 0.377$	
1	0.059768	0.060247	0.061477	0.064195	0.061839	0.061705
2	0.059557	0.060305	0.066074	0.062202	0.061591	0.062086
3	0.060392	0.060265	0.060066	0.064034	0.061878	0.062169
4	0.059184	0.060383	0.059073	0.063801	0.062069	0.062024
5	0.060472	0.059441	0.061253	0.065520	0.062309	0.061583
6	0.059800	0.059399	0.060946	0.065181	0.062624	0.062539
7	0.059790	0.060399	0.061038	0.061956	0.063428	0.062122
8	0.060418	0.060446	0.060672	0.062278	0.062424	0.062418
9	0.059645	0.059410	0.061090	0.062667	0.062594	0.062146
10	0.059937	0.059766	0.059131	0.064816	0.063302	0.062642
11	0.060081	0.059702	0.060773	0.062415	0.062337	0.062388
12	0.059567	0.059115	0.061071	0.062242	0.062429	0.062213
13	0.060236	0.059587	0.061441	0.061931	0.062463	0.062005
14	0.060169	0.059577	0.060639	0.061849	0.062617	0.061784
15	0.059043	0.059061	0.060828	0.062801	0.062604	0.061546
16	0.060641	0.059439	0.060093	0.062195	0.062227	0.062229
17	0.059741	0.060047	0.060546	0.062438	0.062003	0.061539
18	0.060338	0.060166	0.061122	0.062266	0.062281	0.061987
19	0.060285	0.059978	0.061270	0.062899	0.062866	0.062872
20	0.060225	0.060182	0.061379	0.063070	0.062987	0.062940
21	0.060005	0.059985	0.061198	0.065396	0.063172	0.062329
22	0.060232	0.060919	0.061860	0.062984	0.063287	0.062465
23	0.060001	0.060744	0.060270	0.068816	0.062024	0.063054
24	0.065370	0.060852	0.061336	0.062719	0.063437	0.062638
25	0.060935	0.059727	0.061198	0.063977	0.063007	0.062091
26	0.060136	0.060562	0.061423	0.065885	0.062797	0.062747
27	0.060826	0.060778	0.061287	0.062835	0.063387	0.062606
28	0.059883	0.060250	0.061620	0.066119	0.062359	0.062563
29	0.060226	0.060497	0.061420	0.061902	0.062631	0.063351
30	0.060305	0.060182	0.062251	0.062879	0.064076	0.062734
31	0.061169	0.060948	0.061103	0.062491	0.063363	0.063082
32	0.060996	0.060131	0.061744	0.062283	0.063494	0.063226
33	0.060108	0.060622	0.061629	0.062601	0.063830	0.063355
34	0.060255	0.060854	0.061454	0.063984	0.063894	0.063308
35	0.060699	0.060659	0.061412	0.062211	0.063274	0.063874
36	0.060673	0.060464	0.061159	0.063489	0.063056	0.063747

37	0.060850	0.059773	0.060785	0.062576	0.063309	0.063637
38	0.060255	0.060372	0.062269	0.065844	0.063349	0.063699
39	0.058921	0.058515	0.059936	0.065722	0.062031	0.061300
40	0.059095	0.058639	0.059181	0.065900	0.062150	0.061977
41	0.060121	0.059726	0.060813	0.062663	0.062208	0.062371
42	0.060598	0.059948	0.061147	0.063627	0.062586	0.062793
43	0.060060	0.060194	0.061189	0.062629	0.063754	0.063022
44	0.060592	0.060349	0.059981	0.062223	0.063372	0.063149
45	0.060386	0.060185	0.061681	0.065801	0.063694	0.062740
46	0.060218	0.060261	0.061258	0.062979	0.061939	0.062720
47	0.060324	0.060109	0.062087	0.065802	0.063168	0.062967
48	0.061098	0.060385	0.059640	0.062437	0.063041	0.063327
49	0.059962	0.060342	0.062854	0.062453	0.063682	0.063010
50	0.060423	0.060298	0.062158	0.062615	0.063406	0.063287

Table 57: End effector linear velocity (L. Vel.): spiral trajectory with best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 125$	$k = 125$	$q = 270, r = 0.01$		$\gamma_T = 0.372, \gamma_O = 0.377$	
1	1.399483	1.411059	1.438439	1.494212	1.433499	1.437929
2	1.400017	1.417262	1.568803	1.447410	1.428857	1.449927
3	1.418412	1.406143	1.403008	1.499287	1.432601	1.440718
4	1.398260	1.416447	1.372328	1.486238	1.444600	1.436365
5	1.412392	1.396981	1.433841	1.530274	1.443246	1.437855
6	1.402879	1.395850	1.426109	1.517968	1.443618	1.449031
7	1.402107	1.415234	1.427624	1.438804	1.470720	1.439208
8	1.419534	1.412406	1.418259	1.450840	1.440662	1.440771
9	1.396762	1.390300	1.434810	1.455159	1.449053	1.436349
10	1.403353	1.397998	1.389909	1.514277	1.462083	1.447435
11	1.407714	1.404182	1.421796	1.455052	1.445252	1.446066
12	1.395245	1.397696	1.429480	1.456903	1.452524	1.445529
13	1.420733	1.401396	1.428216	1.440442	1.446661	1.431291
14	1.418505	1.406200	1.421525	1.439875	1.448141	1.446165
15	1.395696	1.394938	1.420126	1.463718	1.454640	1.442778
16	1.427199	1.396136	1.418573	1.444974	1.434819	1.440737
17	1.409455	1.411696	1.418848	1.441335	1.432286	1.431282
18	1.420178	1.399931	1.427040	1.447168	1.446877	1.420518
19	1.411199	1.404254	1.426007	1.469560	1.457540	1.445769
20	1.401699	1.399496	1.437814	1.465639	1.457877	1.439672
21	1.403502	1.405708	1.420896	1.510867	1.453412	1.437920
22	1.408226	1.409831	1.431354	1.453136	1.457140	1.431623
23	1.401295	1.419852	1.407665	1.598111	1.438274	1.449101
24	1.563865	1.416436	1.428679	1.447977	1.452615	1.433749
25	1.420126	1.402772	1.427174	1.460260	1.441786	1.435646
26	1.397649	1.410234	1.430534	1.511812	1.442098	1.439619
27	1.415484	1.414266	1.428655	1.450081	1.458430	1.441435
28	1.399886	1.399806	1.432494	1.516444	1.438762	1.437854
29	1.397035	1.410348	1.432434	1.433977	1.436803	1.453390
30	1.407053	1.405181	1.434874	1.458840	1.472612	1.450268
31	1.426185	1.410553	1.404748	1.449806	1.468335	1.442091
32	1.397632	1.402294	1.423830	1.437411	1.452954	1.439337
33	1.400339	1.400120	1.418400	1.448423	1.458191	1.438074
34	1.396265	1.422322	1.438159	1.484264	1.463988	1.450731
35	1.398448	1.396501	1.427694	1.435127	1.453410	1.458135
36	1.404090	1.395210	1.421203	1.468955	1.449454	1.458506

37	1.415459	1.396629	1.408293	1.438241	1.442837	1.451079
38	1.403730	1.400166	1.441585	1.509079	1.448584	1.448213
39	1.399148	1.390267	1.423571	1.510498	1.445671	1.434142
40	1.399094	1.390247	1.408028	1.515008	1.451705	1.448484
41	1.402261	1.406544	1.435034	1.447075	1.441887	1.443890
42	1.414754	1.406313	1.442983	1.476003	1.451384	1.449490
43	1.402609	1.405138	1.427436	1.445067	1.465959	1.443314
44	1.408687	1.400724	1.403025	1.437116	1.454030	1.452816
45	1.400552	1.402239	1.430559	1.514759	1.450519	1.439234
46	1.400756	1.401823	1.423230	1.446769	1.430345	1.430211
47	1.401856	1.399462	1.442064	1.510117	1.437132	1.445827
48	1.416689	1.401773	1.385275	1.449904	1.440161	1.460286
49	1.396835	1.406511	1.448149	1.441321	1.454743	1.445333
50	1.404236	1.401656	1.419384	1.440520	1.452064	1.456642

Table 59: End effector linear acceleration (L. Acc.): spiral trajectory with best performance gains.

Exp.	K	K+FF	LQR in \mathbb{R}^8	LQR in DQ	HIR	HIRT
	$k = 125$	$k = 125$	$q = 270, r = 0.01$		$\gamma_T = 0.372, \gamma_O = 0.377$	
1	121.977267	121.911183	111.710227	108.650290	121.231423	121.747685
2	119.248318	147.296076	124.443725	115.719709	122.411007	124.449226
3	121.704519	130.719880	118.294910	107.280744	121.572509	127.008601
4	118.303425	154.797714	120.362949	110.216055	122.411614	120.945340
5	127.259299	119.884171	111.998760	107.257322	122.294069	120.283021
6	121.315792	116.954452	114.088447	105.128805	123.014126	123.195106
7	122.485063	128.606592	113.631683	115.918625	129.843528	121.963415
8	132.059760	137.310429	122.468022	114.652594	119.862870	125.727656
9	118.764919	117.099860	114.264297	114.009295	123.676729	120.630941
10	124.338293	119.606343	121.976745	105.938510	123.758950	122.354796
11	127.560917	128.173617	114.547718	115.060862	123.903311	121.662205
12	119.968950	119.887204	112.318739	113.897842	127.447881	122.358488
13	143.742855	123.103753	114.472601	117.887702	127.744912	123.155356
14	133.690982	117.677758	114.683729	116.379657	122.733213	121.894354
15	120.667073	117.428802	114.665812	113.682923	122.369375	121.447519
16	136.827736	119.463832	117.357784	116.045534	122.502154	121.209447
17	133.981758	135.437982	112.988329	116.169641	119.821052	123.805566
18	129.346898	123.770257	115.688444	117.016867	128.664859	121.776364
19	134.806871	120.998063	113.454974	114.314165	129.303406	127.459432
20	125.304117	126.070758	112.667933	113.442925	135.073586	125.934184
21	123.400786	141.434088	115.975156	107.699632	124.453185	122.350133
22	121.211703	129.527019	113.875335	115.494720	129.875588	120.986310
23	118.886389	132.517303	120.222647	93.665272	121.820173	133.192051
24	130.668799	141.712936	115.804372	113.548266	125.787165	123.684034
25	142.517232	119.154675	113.970839	117.041737	125.711213	123.807801
26	117.042597	134.310922	112.765863	105.055816	128.317985	122.988970
27	129.614696	131.014342	113.504214	116.854273	127.777409	126.576895
28	119.661629	122.631440	113.035101	104.816552	121.409535	125.706211
29	126.424135	126.612859	112.062341	115.794630	121.221771	124.752493
30	118.481105	116.919784	115.635612	112.867235	127.198617	121.789353
31	153.421803	142.866045	116.487992	115.563642	127.373645	126.801895
32	123.097716	119.696452	113.585467	116.156435	124.277547	122.310050
33	117.387412	117.111783	118.163298	116.229345	123.804709	121.806934
34	117.966112	120.140205	113.037159	108.710672	135.949398	123.263393
35	120.214966	119.027963	113.818879	117.279674	120.650087	125.831767
36	118.552015	119.124121	114.820153	112.377413	122.632357	122.140665

37	132.923691	119.834833	118.499829	116.095496	120.512478	124.678089
38	128.120432	119.281412	112.951896	105.576028	123.613917	132.617521
39	136.756726	119.596206	113.531337	109.648651	126.832472	120.311925
40	127.941170	117.087351	115.021178	104.692579	122.646718	121.366421
41	128.401150	125.415820	112.194488	116.693371	131.930306	121.601653
42	129.675906	127.611068	110.496971	112.177011	127.180220	122.157785
43	127.049601	127.988935	112.190205	115.314051	141.045385	121.477968
44	128.702550	117.882020	117.917611	116.253676	129.500610	129.335994
45	130.410687	131.696299	112.431261	106.127513	126.980840	122.092519
46	125.064225	122.179118	112.384484	114.531994	124.180986	122.739825
47	117.293851	119.939351	111.386420	105.976053	125.917449	124.708388
48	139.875463	129.407853	120.483181	115.101552	130.712334	122.293689
49	120.185982	120.200799	109.973062	115.766772	127.085533	121.525075
50	123.729140	127.032783	111.024958	115.682824	123.994992	122.289483

E.7 Table from Sampling Time Histogram–Section 5.8

Table 61: Sampling time histograms mean values and standard deviation.

Controller	$T_s = 8 \text{ ms}$		$T_s = 20 \text{ ms}$	
	Mean	Std. Dev.	Mean	Std. Dev.
K	0.008012	0.001200	0.020001	0.000423
K+FF	0.008000	0.001411	0.019999	0.000355
LQR in \mathbb{R}^8	0.008239	0.000551	0.019998	0.000630
LQR in DQ	0.008175	0.000188	0.021174	0.000483
HIR	0.007999	0.001228	0.020001	0.000354
HIRT	0.007999	0.000660	0.019999	0.000397

IV. DESCRIÇÃO DO CONTEÚDO DO CD

- trabalho_de_graduação.pdf: relatório de Trabalho de Graduação;
- resumo_do_trabalho.pdf: resumo do relatório de Trabalho de Graduação;
- abstract_do_trabalho.pdf: *abstract* do relatório de Trabalho de Graduação.