



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학박사학위논문

구조 로봇을 위한 강건한 계층적
동작 계획 및 제어

**Robust Hierarchical Motion
Planning and Control for Rescue Robots**

2021 년 8 월

서울대학교 대학원

기계항공공학부

홍 성 일

구조 로봇을 위한 강건한 계층적
동작 계획 및 제어

**Robust Hierarchical Motion
Planning and Control for Rescue Robots**

지도교수 박 종 우

이 논문을 공학박사 학위논문으로 제출함

2021년 4월

서울대학교 대학원

기계항공공학부

홍 성 일

홍성일의 공학박사 학위논문을 인준함

2021년 6월

위원장 : 이 동 준

부위원장 : 박 종 우

위원 : 박 재 홍

위원 : 박 용 래

위원 : 곽 기 호

Robust Hierarchical Motion Planning and Control for Rescue Robots

SEONGIL HONG

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in the
DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

at
SEOUL NATIONAL UNIVERSITY

August 2021

ABSTRACT

Robust Hierarchical Motion Planning and Control for Rescue Robots

by

Seongil Hong

Department of Mechanical and Aerospace Engineering
Seoul National University

Over the last several years, robotics has experienced a striking development, and a new generation of robots has emerged that shows great promise in being able to accomplish complex tasks associated with human behavior. Nowadays the objectives of the robots are no longer restricted to the automaton in the industrial process but are changing into explorers for hazardous, harsh, uncooperative, and extreme environments. As these robots usually operate in dynamic and unstructured environments, they should be robust, adaptive, and reactive under various changing operation conditions.

We propose online hierarchical optimization-based planning and control methodologies for a rescue robot to execute a given mission in such a highly unstructured environment. A large number of degrees of freedom is provided to robots in order to achieve diverse kinematic and dynamic tasks. However, accomplishing such multiple objectives renders on-line reactive motion planning and control problems more difficult to solve due to the incompatible tasks. To address this problem, we exploit a hierarchical structure to precisely resolve conflicts by creating a priority in which every task is achieved as much as possible according to the levels. In particular, we concentrate on the reasoning about the task regularization to ensure the convergence and robustness of a solution in the face of singularity. As robotic systems with real-time motion planners or controllers often execute unrehearsed missions, a desired task cannot always be driven to a singularity free configuration.

We develop a generic solver for regularized hierarchical quadratic programming without resorting to any off-the-shelf QP solver to take advantage of the null-space projections for computational efficiency. Therefore, the underlying principles are thoroughly investigated. The robust optimal solution is obtained under both equality and inequality tasks or constraints while addressing all problems resulting from the regularization. Especially as a singular value decomposition centric approach is leveraged, all hierarchical solutions and Lagrange multipliers for properly handling the inequality constraints are analytically acquired in a recursive procedure. The proposed algorithm works fast enough to be used as a practical means of real-time control system, so that it can be used for online motion planning, motion control, and interaction force control in a single hierarchical optimization.

Core system design concepts of the rescue robot are presented. The goals of the robot are to safely extract a patient and to dispose a dangerous object instead of

humans. The upper body is designed humanoid in form with replaceable modularized dual arms. The lower body is featured with a hybrid tracked and legged mobile platform to simultaneously acquire versatile manipulability and all-terrain mobility. Thus, the robot can successfully execute a driving task, dangerous object manipulation, and casualty extraction missions by changing the pose and modularized equipments in an optimized manner.

Throughout the dissertation, all proposed methods are validated through extensive numerical simulations and experimental tests. We highlight precisely how the rescue robot can execute a casualty extraction and a dangerous object disposal mission both in indoor and outdoor environments that none of the existing robots has performed.

Keywords: Hierarchical optimization, regularization, singularity robustness, priority, inequality constraint, rescue robot.

Student Number: 2016-30181

Contents

Abstract	iii
List of Tables	ix
List of Figures	xi
List of Symbols	xiii
1 Introduction	1
1.1 Motivations	1
1.2 Related Works and Research Problems for Hierarchical Control . .	3
1.2.1 Classical Approaches	3
1.2.2 State-of-the-Art Strategies	4
1.2.3 Research Problems	7
1.3 Robust Rescue Robots	9
1.4 Research Goals	12
1.5 Contributions of This Thesis	13
1.5.1 Robust Hierarchical Task-Priority Control	13

1.5.2	Design Concepts of Robust Rescue Robot	16
1.5.3	Hierarchical Motion and Force Control	17
1.6	Dissertation Preview	18
2	Preliminaries for Task-Priority Control Framework	21
2.1	Introduction	21
2.2	Task-Priority Inverse Kinematics	23
2.3	Recursive Formulation of Null Space Projector	28
2.4	Conclusion	31
3	Robust Hierarchical Task-Priority Control	33
3.1	Introduction	33
3.1.1	Motivations	35
3.1.2	Objectives	36
3.2	Task Function Approach	37
3.3	Regularized Hierarchical Optimization with Equality Tasks	41
3.3.1	Regularized Hierarchical Optimization	41
3.3.2	Optimal Solution	45
3.3.3	Task Error and Hierarchical Matrix Decomposition	49
3.3.4	Illustrative Examples for Regularized Hierarchical Optimiza- tion	56
3.4	Regularized Hierarchical Optimization with Inequality Constraints .	60
3.4.1	Lagrange Multipliers	61
3.4.2	Modified Active Set Method	66
3.4.3	Illustrative Examples of Modified Active Set Method	70
3.4.4	Examples for Hierarchical Optimization with Inequality Con- straint	72

3.5	DLS-HQP Algorithm	79
3.6	Concluding Remarks	80
4	Rescue Robot Design and Experimental Results	83
4.1	Introduction	83
4.2	Rescue Robot Design	85
4.2.1	System Design	86
4.2.2	Variable Configuration Mobile Platform	92
4.2.3	Dual Arm Manipulators	95
4.2.4	Software Architecture	97
4.3	Performance Verification for Hierarchical Motion Control	99
4.3.1	Real-Time Motion Generation	99
4.3.2	Task Specifications	103
4.3.3	Singularity Robust Task Priority	106
4.3.4	Inequality Constraint Handling and Computation Time	111
4.4	Singularity Robustness and Inequality Handling for Rescue Mission	117
4.5	Field Tests	122
4.6	Concluding Remarks	126
5	Hierarchical Motion and Force Control	129
5.1	Introduction	129
5.2	Operational Space Control	132
5.3	Acceleration-Based Hierarchical Motion Control	134
5.4	Force Control	137
5.4.1	Force Control with Inner Position Loop	141
5.4.2	Force Control with Inner Velocity Loop	144
5.5	Motion and Force Control	145

5.6	Numerical Results for Acceleration-Based Motion and Force Control	148
5.6.1	Task Specifications	150
5.6.2	Force Control Performance	151
5.6.3	Singularity Robustness and Inequality Constraint Handling .	155
5.7	Velocity Resolved Motion and Force Control	160
5.7.1	Velocity-Based Motion and Force Control	161
5.7.2	Experimental Results	163
5.8	Concluding Remarks	167
6	Conclusion	169
6.1	Summary	169
6.2	Concluding Remarks	173
A	Appendix	175
A.1	Introduction to PID Control	175
A.2	Inverse Optimal Control	176
A.3	Experimental Results and Conclusion	181
	Bibliography	183
	Abstract	207

List of Tables

4.1	Hurcules hardware specifications for rescue and manipulation missions	87
4.2	Computation time (<i>msec</i>) for RHQP, HQP and DLS-HQP with respect to the number of levels.	116

List of Figures

1.1	Existing rescue robots with mobile platform: (a) CHIMP (CMU Highly Intelligent Mobile Platform) (b) RoboSimian (c) DRC-Hubo (d) Momaro (e) RoNA (f) Bear (g) RIBA	10
1.2	Organization of dissertation.	20
2.1	Real-time control block diagram for the hierarchical motion planning and control framework. As the control frequency is set at 1kHz for the Hurcules robot, a hierarchical optimization algorithm should provide the results every 1 <i>ms</i>	22
3.1	A simple robot contains a mobile base that allows to move along the horizontal axis with q_1 and an arm of unit length connected by means of a revolute joint with an angle q_2 . The first level objective is to reach a target point ${}^d s_1$ with the arm's horizontal position, and the second level task is to position the mobile base at a specified point ${}^d s_2$	39

3.2	Comparison of the reciprocal singular values in (??). If the regularization parameter is zero, the singular value $\hat{\sigma}_{k,i}$ would grow unboundedly as $\sigma_{k,i}$ approaches zero near a singularity.	48
3.3	Numerical simulation results computed via unregularized HQP (left column) and RHQP (right column). Two tasks can be realized at the same time. We compare the results in terms of time history of the magnitude of the task function (top row), distance to the optima (middle row), and time history of the robot poses according to various initial conditions (bottom row). The convergence speed of the unregularized solution is faster than that of the regularized one. However, the unregularized solution starting near singular point (case 5) requires high velocity.	58
3.4	Numerical simulation results computed via unregularized HQP (left column) and RHQP (right column). Two tasks cannot be satisfied simultaneously: there is a conflict between two tasks. We compare the results in terms of time history of the magnitude of the task function (top row), distance to the optima (middle row), and time history of the robot poses according to various initial conditions (bottom row). As it can be seen, the solution sequences of unregularized HQP does not converge to the optima while all solutions of RHQP converge to the optima respecting hierarchy.	59

3.5	Comparison of the active set method and modified active set method. (a) Active set method: 1: starting point (1': unconstrained optimum) → 2: constraint A is activated (2': optimum on constraint A) → 3: constraint A is deactivated and B is activated → 4: optimum (b) Modified active set method: 1: unconstrained optimum → 2: opti- mum (constraint B is activated).	71
3.6	Numerical simulation results computed via unregularized HQP (left column) and RHQP (right column). Two tasks can be realized at the same time. We compare the results in terms of time history of the magnitude of the task function (top row), distance to the optima (bottom row). The convergence speed of the unregularized solution is faster than that of the regularized one. All task function values converge to zeros	74
3.7	Time history of the robot poses computed via unregularized HQP (top row) and RHQP (bottom row) where target point is within the reachable space. The shaded areas represent the position range constraints of the mobile base.	75
3.8	Numerical simulation results computed via unregularized HQP (left column) and RHQP (right column). Two tasks cannot be satisfied simultaneously and there is a conflict between two tasks. We com- pare the results in terms of time history of the magnitude of the task function (top row), distance to the optima (bottom row). As it can be seen, the solution sequences of unregularized HQP cannot converge to the optima while all solutions of RHQP converge to the optima $[g_1(\mathbf{q}^*) , g_2(\mathbf{q}^*)]^T = [0, 0.5]^T$ respecting hierarchy.	77

3.9	Time history of the robot poses computed via unregularized HQP (top row) and RHQP (bottom row) where target point is within the unreachable space. The shaded areas represent the position range constraints of the mobile base.	78
4.1	Missions of the rescue robot, Hurcules (HUMANoid ResCUe robot caLAMity reSPonse): (a) casualty extraction (b) explosive ordnance disposal and transportation (c) driving activities on various terrains	86
4.2	Configurations of the rescue robot, Hurcules for manipulation (left) and rescue (right) missions.	88
4.3	Various poses of the rescue robot, Hurcules: drive and task execution postures.	89
4.4	Maximum weight lifting capacity: (a) casualty extraction (120kg) (b) explosive ordnance transportation (30kg per arm)	91
4.5	Mechanical design of the variable configuration mobile platform. . .	92
4.6	Pelvis and upper leg structure of the mobile platform.	93
4.7	Lower leg structure of the mobile platform.	93
4.8	Mechanical design of the dual arm manipulators with replaceable modular type arms.	96
4.9	Block diagram of software structures	98
4.10	Configuration of Hurcules robot: (a) Joystick and haptic device for the task space motion synthesis (b) RHQP algorithm (c) Joint configuration in which the arrows represent the axis of rotation. (d) Visual sensors	100

4.11	Representative tasks for manipulation. (a) position and orientation of hand (b) pelvis traversal position control (c) pelvis pitch angle control (d,e) swivel angle control (f) gaze control.	102
4.12	Pelvis pose and swivel angle control: (a) ready for manipulation (b) pushing the pelvis forward (15cm) (c) lowering the pelvis down (d, e, f) swivel angle control.	103
4.13	Constrained dual arm manipulation: (a) ready for manipulation (b) moving hands sideways (c) rotating hands in roll direction (d) rotating hands in roll and yaw direction.	104
4.14	Stretching motion of the left arm for grasping a distal object with the RHQP solver. The dashed line represents the activated joint limit constraint at REB (right elbow) and RS (right shoulder) Roll joints.	105
4.15	Three velocity solution trajectories of RHQP (top row), HQP without regularization (middle row) and DLS-HQP (bottom row): (a) left arm (b) right arm (c) lower body and neck.	107
4.16	Number of iterations of RHQP (Left) and DLS-HQP (Right). The RHQP algorithm can find the optimal active set in 2 iterations. The DLS-HQP solution shows cycling from 24 s to 26 s with the maximum number of iterations.	108
4.17	Reference joint angle trajectories (left) and task errors (right) of RHQP. (Left) The elbow and shoulder roll joint constraints are activated at $t = 25$ s and $t = 33$ s respectively. (Right) The RHQP solver automatically gives up low-priority tasks, i.e. \mathbf{g}_{body} , \mathbf{g}_{pel} , and \mathbf{g}_{swivel} to satisfy the hand pose tasks. Thus the task errors increase.	109

4.18	Joint space control errors with the RHQP solution: (a) left arm (b) right arm.	110
4.19	Robot poses for bimanual manipulation with the RHQP solver. (a) Initial pose (b) Internal singularity is encountered by the alignment of two wrist yaw joints when the left forearm is outstretched. (c) Right elbow (REB), right shoulder roll (RSR), and left wrist yaw2 (RWY2) joint constraints are activated. (d) Right elbow (REB) joint constraint becomes deactivated.	112
4.20	Joint angle trajectories (top) and number of iterations (bottom) for bimanual manipulation. (a) Results for RHQP (left). (c) Results for DLS-HQP (right). The RHQP algorithm deactivates the right elbow joint constraint at $t = 12$ s and does not show any cycling. The DLS-HQP solver cannot remove the right elbow constraint at that time and shows cycling with the maximum number of iterations (bottom right). Edges of shaded areas represent the time of activations or deactivations of inequality constraints.	113

4.21	<p>Snapshots of bimanual manipulation. The arrows represent the direction of each task. The robot configurations associated with the RHQP solution trajectory: $(a) \rightarrow (b) \rightarrow (c) \rightarrow (d)$ and the DLS-HQP solution trajectory: $(a) \rightarrow (b) \rightarrow (e) \rightarrow (f)$. As the robot pulls its hands toward its body, the elbow joint constraints are activated in (b). The RHQP solver removes the elbow joint constraints as soon as the hand task direction reverses as shown in (c). However, the DLS-HQP solver maintains the elbow joint limit constraint even though the constraint should be removed. Thus, the DLS-HQP gives up the pelvis task by moving the knee and hip joints to perform the hand tasks with priority.</p>	114
4.22	<p>Computation time for RHQP, HQP and DLS-HQP with respect to the number of levels. The computation costs of RHQP is nearly equal to the HQP and is cheaper than that of DLS-HQP for $p > 3$.</p>	115
4.23	<p>Joint configuration of the Hurcules robot for a rescue mission where arrows represent the axis of rotation.</p>	118
4.24	<p>Motion tasks for rescue mission. (Left) Pelvis traversal position control (Top) Gaze angle control (Right) Hand position control (Bottom) Hand pitch angle control</p>	119
4.25	<p>Snapshots of stretching motion for the rescue mission. Equipments can severely restrict the joint operation range (top left). The dashed line represents the activated or deactivated joint limit constraint.</p>	120
4.26	<p>Experimental results for RHQP solution: (a) reference joint angular velocities (b) joint space control errors.</p>	121

4.27	Motion sequences of a dangerous object acquisition mission: (1) Driving: moving to the destination (2) turning the waist (3) rotating both lower legs at the same time (4) Manipulation Ready: standing up and being ready for the grasp task (5,6,7,8) Acquisition: task space control via RHQP solver to grasp a dangerous object (9) Lifting and sitting down (10) turning the waist and rotating the right leg (11) turning the waist in the opposite direction and rotating the left lower leg (12) Transfer: pointing the torso in the forward direction and leaning the torso.	124
4.28	Motion sequences of the rescue mission: (1) Driving: moving to the destination (2) turning the waist (3) folding dual arms to avoid self collision and (4) rotating both lower legs at the same time (5) Rescue Ready: standing up and being ready for the rescue task (6,7) Acquisition: task space control via RHQP solver to lift the injured (8) Lifting (9) sitting down, turning the waist, and rotating the right leg (10) turning the waist in the opposite direction, rotating the left lower leg and pointing the torso in the forward direction of driving (11) Putting Down: deploying the transfer bed and putting safely the injured on the bed (12) Transfer: pulling out the arms and being ready to transfer.	125
4.29	Visual information of cameras corresponding to the motion sequences of the rescue mission: (a) Lifting Ready (b,c) Acquisition (d) Lifting.	126
5.1	Block diagram of operational space motion control with the RHQP solver.	138

5.2	Spring mass damper system for force control. The stabilizing damping and spring effects are obtained by the inner operational space position or velocity feedback. Thus, the force control scheme attains a stabilizing PD control action based on velocity and position measurements as well as force error.	139
5.3	Block diagram of force control.	140
5.4	Block diagram of force control with inner position loop. If the position feedback loop (dashed line) is deleted, it represents force control with inner velocity loop.	143
5.5	Block diagram of acceleration-based motion and force control . . .	148
5.6	Injured acquisition movement sequences. (a) Driving: arriving at the destination (b) open the arms to avoid self-collision (c) turning the waist (d) folding dual arms and (e) rotating both lower legs at the same time (f) Rescue Ready: standing up and being ready for the rescue task (g) Lowering: The lowering motion phase is executed under acceleration-based motion control at first. If the contact force exceeds a specified threshold, the force control begins to work. (h) Acquisition: task space control via motion and force control for the acquisition of an injured lying on the ground (i) Confirm: To move out of contact is executed by only motion control scheme. (j) Embracing (k-l) Lifting.	149
5.7	Lowering motions: (a) with force control laws (b) without force control laws.	152

5.8	Time histories of the measured force when three force control laws are applied. (a) Measured force for the proportional force control with inner position loop cannot reach the desired force at steady state because C_k is pure proportional control action. (b) The proportional-integral force control with inner position loop can reject the steady state force error, however the response is slow due to the third order system characteristics with the reduced the stability margin. (c) The proportional force control with inner velocity loop shows that the desired contact force $-5N$ is achieved. (d) Excessive contact force arises when only motion control is applied.	153
5.9	Lowering motions with and without force control laws. Two solution trajectories (angular acceleration, angular velocity) with force control law (a) and without force control law (b).	154
5.10	Snapshots of an injured acquisition motion with the motion and force control laws. The dashed lines represent the activated or deactivated joint limit constraints. The inequality constraint of the hip joint is activated at $t = 8 s$ and deactivated $t = 11 s$. The RSP (right shoulder pitch) and LSP (left shoulder pitch) joint limit constraints are activated at $11 s$. The arrows (top left corner) represent the directions of motion and force control where an orthogonal reference frame is specified.	156
5.11	Solution trajectories for hierarchical motion and force control with inner velocity loop for two compliance gains: (Left) angular acceleration (Right) angular velocity.	157

5.12	Number of iterations and joint angle trajectories. (a) The acceleration-based RHQP algorithm activates and deactivates the hip joint angle constraint at $t = 8$ s and $t = 11$ s respectively. The number of iteration is 2 at $t = 8$ s to activate the hip joint constraint. The number of iteration becomes 3 at $t = 11$ s, because the left and right shoulder joint constraints are activated and the hip joint constraint is deactivated. (b) Hip and shoulder pitch joint angle constraints are satisfied for stretching motion.	158
5.13	Numerical simulation results for measured force with hierarchical motion and force control laws.	159
5.14	Block diagram of velocity-based motion and force control	162
5.15	Injured acquisition movements with the velocity-based of RHQP. The force control gain $C_k = 0.05$ is applied.	165
5.16	Experimental results for velocity-based motion and force control. Time history of the estimated contact force for various compliance gains. As the compliance gain increases, the faster response is obtained.	166
5.17	Joint velocity solutions for motion and force control. The force control gain $C_k = 0.05$ is applied.	166
A.1	Gain tuning postures with the seven-DoF manipulator.	180
A.2	Joint space control errors for the seven-DoF manipulators.	180
A.3	Weight lifting motions with the maximum payload 120kg.	181
A.4	Joint space control errors for the weighting lifting motions with the maximum payload (120kg).	182

List of Symbols

Notation	Meaning
$\mathbb{R}, \mathbb{R}^n, \mathbb{R}^{m \times n}$	real numbers, n -dimensional vectors, and $m \times n$ matrices
$\mathbf{s}_k, \dot{\mathbf{s}}_k, \ddot{\mathbf{s}}_k$	task space position, velocity, and acceleration vectors in \mathbb{R}^{m_k}
${}^d\mathbf{s}_k, {}^d\dot{\mathbf{s}}_k, {}^d\ddot{\mathbf{s}}_k$	desired task space position, velocity, and acceleration vectors in \mathbb{R}^{m_k}
$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$	configuration space angle, angular velocity, and angular acceleration vectors in \mathbb{R}^n
$\mathbf{q}^*, \dot{\mathbf{q}}^*, \ddot{\mathbf{q}}^*$	optimal solution vectors for configuration space angle, angular velocity, and angular acceleration \mathbb{R}^n
k	index of the level $1 \leq k \leq p$
p	total number of the levels
$\phi_k(\mathbf{q})$	forward kinematics mapping: $\mathbb{R}^n \rightarrow \mathbb{R}^{m_k}$
\mathbf{J}_k	k level Jacobian matrix $\mathbf{J}_k \in \mathbb{R}^{m_k \times n}$
$\underline{\mathbf{A}}_k$	augmented matrix or vector from level 1 to k , $[\mathbf{A}_1^T \cdots \mathbf{A}_k^T]^T$
\mathbf{P}_k	null space projecting matrix of the augmented matrix $\underline{\mathbf{J}}_k$
$\mathbf{g}, \dot{\mathbf{g}}$	task function and time derivative of task function
ρ	regularization coefficient

Notation	Meaning
\mathbf{e}, \mathbf{e}^*	task error and optimal task error vectors
\mathbf{D}_k	k level regularization matrix
\mathbf{Z}_k	null space basis matrix
\mathbf{Y}_k	row space basis matrix
$\underline{\mathbf{Y}}_k$	augmented row space basis matrix $[\mathbf{Y}_1, \dots, \mathbf{Y}_k]$
σ	singular values
$\mathbf{\Sigma}$	diagonal matrix composed of singular values $\sigma_1, \dots, \sigma_l$
$\mathbf{\Sigma}^{-1}$	inverse matrix of $\mathbf{\Sigma}$
\mathbf{J}^\dagger	Moore-Penrose generalized inverse matrix of \mathbf{J}
$\mathbf{J}^\#$	Reflexive generalized inverse matrix of \mathbf{J}
\mathbf{Q}	triangular matrix of complete orthogonal decomposition
\mathbf{I}_k	k level identity matrix with an appropriate dimension
$\boldsymbol{\mu}$	Lagrange multiplier vector
\mathbf{M}^*	Lagrange multiplier matrix
\mathcal{W}	working set
(i)	iteration number of active set method
$cst(k, r)$	r^{th} constraint of the level k
$\mathbf{g}_A \prec \mathbf{g}_B$	priority order: \mathbf{g}_A task has higher priority than \mathbf{g}_B

Notation	Meaning
\mathbf{K}, \mathbf{L}	position and velocity gain matrices
${}^d\mathbf{f}, \mathbf{f}$	desired and measured forces
\mathbf{C}	Compliance gain matrix of force control
δ	Output of force controller
\mathbf{G}	stiffness matrix of the environment
\mathbf{r}	environment's rest position
t	time

1

Introduction

1.1 Motivations

Striking developments of humanoid and field robotics have taken place recently. Nowadays, the roles of robots are no longer limited to the automaton in the industrial process but are changing into explorers for hazardous, harsh, uncooperative, and extreme environments. A new generation of robots has emerged that shows great promise in being able to accomplish complex tasks associated with human behavior.

For example, DARPA Robotics Challenge (DRC) has spawned real world robots and accelerated relevant technology advancement [1, 2, 3, 4, 5]. DRC has a scenario for a large-scale disaster response given the hazard involved and the quick response needed. Robots can be one of the best solutions in mitigating such events

in a safe and timely manner. As these robots usually operate in dynamic and unstructured environments, they should be robust, adaptive, and reactive under various changing operation conditions. However, DRC shows that there are still challenging problems including falls, failed attempts at tasks, long periods of robot inactivity, and operator errors, so that it seems to be waiting for a solution for real world robotic applications [6]. Moreover, these robots cannot show the fast and stable mobility while traversing on a wide variety of rough terrains [7]. In this dissertation, we propose real-time optimization-based hierarchical control algorithms and novel hardware design concepts for a robot to reactively perform a rescue mission in such a highly unstructured environment.

The general hypothesis that the movement of humans and animals are optimal is one of the fundamentals of bionics. For many biological systems, it has long been observed that motion control can likely be the result of their optimized process, and it comes from a life long learning process. For example, the acquisition of fundamental motor skills in sports is to efficiently synthesize and arrange multiple optimized motion tasks such as locomotion, manipulation, and stability under the physical constraints. Locomotion tasks can include walking, running, jumping, hopping, and slide. Manipulation tasks are throwing, catching, kicking, striking, trapping, and dribble. Finally, stability skills are such as balance, twisting, turning and bending, and physical constraints are joint angle ranges and torque limits. These motion tasks are also differently classified according to the biomechanics perspective [8].

The dexterous human skills refers to synthesizing the multiple optimized motion tasks in an optimized manner and including decision making in case of conflicts between the tasks. There are more important and less important tasks, and the hierarchy is maintained or even changed depending on situations. These skills

are developed throughout the lifespan by participating in physical activity.

Humans and humanoid robots are multi-body systems having much in common. The large number of degrees of freedom (DoF) allows the humans and humanoid robots to adapt to a variety of environments and to execute several tasks skillfully at the same time. Thus, modern intelligent humanoid and articulated robots often attain their tree-like mechanical structure to simultaneously achieve diverse kinematic and dynamic tasks [9]. The motion tasks or primitives of a humanoid robot are the behaviors of the hands, legs, head, center of gravity, body posture, and physical constraints.

However, accomplishing such multiple objectives renders on-line reactive motion planning and control problems more difficult to solve. Because the objectives typically described in the operational space and configuration space are often conflicting, it can lead to a robot configuration in which none of these tasks is exactly satisfied [10, 11, 12, 13]. A decision making emerges to address the incompatible tasks, and we now describe this problem in more detail. In what follows, we use the constraint, objective, and task interchangeably.

1.2 Related Works and Research Problems for Hierarchical Control

1.2.1 Classical Approaches

A solution to address the conflicting tasks is to impose weights according to the importance of the tasks [14, 15], where the objective function can be a weighted sum of multiple costs. Physical constraints such as joint operation range and motor torque limits should be always satisfied regardless of motion tasks, however

this approach does not satisfy any objectives exactly inversely proportional to the weight. Moreover, the selection of a suitable weight depending on the task variations is very difficult and time consuming [16]. For instance, the weight employed for the manipulation tasks may be completely different from the weight for the locomotion tasks.

Another well-known strategy is the use of a hierarchical structure to precisely resolve conflicts by creating a priority [17, 18]. This resolution strategy has been widely used for controlling redundant manipulators and humanoid robots, where every task is achieved as much as possible at different priority level without interfering with the fulfillment of the higher priority tasks [19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33]. The task-priority framework allows the simultaneous execution of multiple objectives in a hierarchical manner, analogous to natural human motion [34, 35]. However, a significant drawback of the classical prioritization algorithm is its inability to explicitly incorporate inequality tasks while guaranteeing singularity robustness [16, 36].

1.2.2 State-of-the-Art Strategies

A direct approach to dealing with inequality constraints is to take each constraint into quadratic program (QP) and solve each level of QP sequentially using an off-the-shelf QP solver [37, 38, 39, 40, 41, 42]. This cascade-QP is computationally inefficient because the level k optimization problem is first solved in the k^{th} QP, and then repeatedly solved at each of the following levels as a constraint. Furthermore, when advancing within the hierarchy, the computation time also increases because of the increased size of the augmented constraint and the iterative process used to find the optimal active set.

At each level QP, the optimal active set may be completely different. For instance, suppose that a humanoid robot manipulates a heavy object with dual arms in the front. When the dual arm task is solved, the center of mass inequality constraint is activated to keep the point inside the support polygon. In the next lower level, the pelvis is positioned on the back, and then the front center of mass position constraint is deactivated. The active set method can activate and deactivate a constraint several times as moving along the hierarchy or cascade, and finally the repetition of all these iterative processes can be inefficient.

This shortcoming is tackled by hierarchical quadratic program (HQP) [43, 44]. The HQP algorithm exploits the prioritized solution computed using a complete orthogonal decomposition (COD) and the null space projections. Thus, by considering all level tasks at once, the algorithm can find the solution more than five-times faster than the cascade-QP and even faster than a weighted least-squares optimization [44, 45]. The HQP method is known to be 10 times faster than the classical method of [18] for an equality constrained problem. Despite its outstanding computational efficiency, HQP is of limited utility in the real world because singularity robustness is not guaranteed.

As robotic systems with real-time motion planners or controllers often execute unrehearsed missions, a desired task cannot always be driven to a singularity free configuration. At near the singularity, a large, unpredictable, and discontinuous motor torque may cause physical damage to the robot and its environment, or even inadvertently bring harm to humans. The most serious problem of singularity is not at singular points themselves, but rather in the neighborhood of singular points. In the vicinity of singularity, even for a small change of the output, an excessive change of the input is often required. The inversion process inherently suffers from this sensitivity close to singular points. When avoiding the singularity

region, the reachable space of a robot is significantly reduced.

As an alternative approach for hierarchical optimization, a trust region method is utilized, although a major difficulty is determining a suitable size of the trust region for a real-time control system [16, 46]. Thus, instead of recalculating solutions with a different size of the trust region, an empirical rule is used within a limited number of iterations. If the trust region is too small, the solution of the model may not be close to the optimal point, although it can be close enough to the optimal point with a substantial step. Otherwise, if the radius is too large, the solution can be far from the optimal point, and another iteration should be performed with the smaller region.

Regularized hierarchical differential dynamic programming is suggested where the optimal control framework, i.e. receding horizon control, is exploited by establishing strict task prioritization between the cost functions [47]. However, the computation time is too long for a fast on-line control system because one iteration takes 1.3 s and dozens of iterations are needed until the convergence is obtained. At each level, the algorithm involves iterating forward pass computing the cost of all tasks for a given locally optimal control policy, and it is followed by a backward pass which finds a locally optimal solution. The two processes are based on the differential dynamic programming and iterative linear quadratic regulator (iLQR) [48, 49].

Machine learning approaches can make use of hierarchical structure, but these studies often focus on the machine learning itself for learning operational space control [50] or manipulability transfer [51]. Therefore, they do not investigate hierarchical motion control under the inequality constraints.

Kinematic singularity problem can be addressed by forward dynamics approach

[52]. Forward dynamic simulation and an impedance controller are utilized to generate a stable joint reference trajectory for a given task space trajectory. However, the control structure is complex. Because the controller is composed of forward dynamic control, nonlinear robust inner-loop compensator for calculating model response [53], and stable PD controller given by [54]. Moreover, this forward dynamics scheme for singularity robustness is only applied to non-redundant robotic manipulators [52].

1.2.3 Research Problems

In general, in order to penalize such a large control input or solution and ensure the convergence, regularization is widely used in robotics, optimal control, numerical optimization, and machine learning with slightly different goals but in a similar way. In numerical optimization, the regularization is leveraged to guarantee good feasibility and convergence by preventing ill conditioning of the iteration matrix with small values of regularization coefficients [55, 56]. For machine learning and parameter identification, regularization plays an important role that allows complex models to be trained on limited size of data sets, thereby adding restrictions on the parameters in order to reduce the generalization error [57, 58, 59, 60]. Regularization of robotics and optimal control also penalizes the norm of the solution or control input but takes much larger coefficient due to the physical limit of actuators. As regularization is invasive to control performance, it has a strong relationship with robustness. Thus extensive studies have been carried out [61, 62, 63, 64, 65, 66].

Despite of its importance, singularity robustness is not explicitly discussed in

an inequality-constrained hierarchical optimization due to the mathematical complexity [46]. Indeed, it is known to be an open problem to seamlessly integrate regularization with inequality-constrained hierarchical optimization [16, 36]. To achieve singularity robustness, a widely used and powerful approach would be to replace the pseudo inverse with the damped least-squares (DLS) inverse in the solution obtained from unregularized equality-constrained optimization [62, 63, 67, 68, 69].

Although the HQP algorithm without regularization has also adopted this DLS strategy, the mixed problem of task regularization and inequality-constrained hierarchical optimization has not been thoroughly investigated [44, 70, 71]. Hereafter, we refer to this as DLS-HQP, which simply replaces the pseudo inverse in the solution with the DLS-inverse. The DLS-HQP approach does not formulate a regularized optimization problem nor address difficulties resulting from regularization such as a non-convexity, regularization-induced errors, difficulty to find Lagrange multipliers, and not leaving redundant resources from higher to lower levels. Especially, we show that, without considering the regularization errors in Lagrange multipliers, the optimization algorithm brings about a *cycling* in the neighborhood of singularity, and it yields the same solution and working set repeatedly to an earlier iteration [16, 55]. Moreover, the algorithm cannot properly deactivate an inequality constraint.

The authors who proposed the HQP algorithm posed this singularity robustness problem resulting from HQP or DLS-HQP as follows [16]: *“The solutions to the non-regularized prioritized problem (HQP in this thesis) can be obtained very efficiently, with the help of specific matrix factorization and lexicographic active set methods when handling inequality constraints, which do not apply to the regularized case [44, 45]. An advantageous aspect of regularization however is that it precludes ill-conditioning when approaching singularities, and ill-conditioning can prevent the*

proper termination of the active set methods used to handle inequality constraints, by inducing a cycling in their iterations.”

In [36], the same problem is also pointed out: *“However, the HQP solver is also very accurate, which means that for highly ill-conditioned problems, it will doubtlessly answer with very high-value control. This is not acceptable in robotics, where we generally do not want to apply a 10^{15} rad/s velocity on the robot joints because the target to grasp is 1cm too far from the robot. ... However, we are not able yet to propose an extension of this regularization to an inequality HQP. In particular, we did not find yet a good solution to take into account the regularization in the active-search loop.”*

In this dissertation, we are going to address this problem by tightly integrating regularization and inequality-constrained hierarchical optimization for real-time motion planning and control.

1.3 Robust Rescue Robots

While most of DRC teams use bipedal humanoid robots, some pursued statically stable hybrid mobile platform with a combination of legs and wheels. As the legged robots are able to strategically choose contact points on the ground, they can move even on really rough terrain [72]. However, most of the advanced walking humanoid robots still suffer from precise balancing control on irregular terrain, and they are apt to fall down while doing tasks as shown in DRC [7]. On the other hand, the robots with hybrid mobile platform showed highly successful results by utilizing a pose transformation. These kinds of robots include CHIMP, RoboSimian, DRC-Hubo, and Momaro as presented in Figure 1.1.

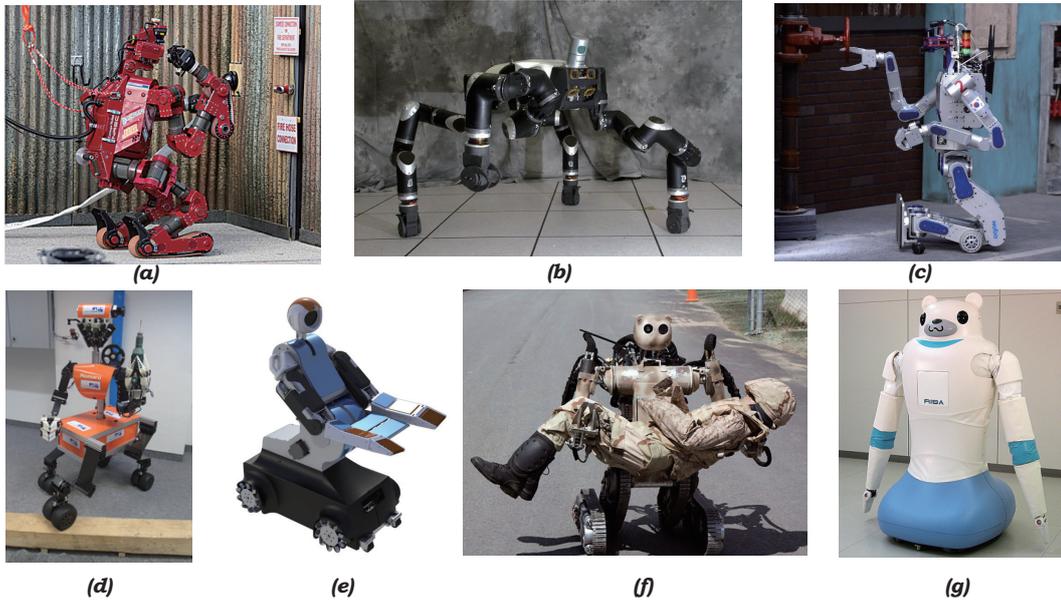


Figure 1.1: Existing rescue robots with mobile platform: (a) CHIMP (CMU Highly Intelligent Mobile Platform) (b) RoboSimian (c) DRC-Hubo (d) Momaro (e) RoNA (f) Bear (g) RIBA

CHIMP placed third in the DRC finals. It is designed to maintain static stability to avoid balancing control [73]. As the robot is equipped with powered tracks on its arms and legs, and it can move over uneven terrain. When the manipulation task execution is required, CHIMP stands up on two legs and manipulates an object with the other two arms as shown in Figure 1.1. CHIMP does not perform any bipedal walking motions.

RoboSimian is a statically stable quadrupedal robot and showed its robustness and versatility with an apelike shape. Due to the four generic limbs, it can execute manipulation tasks and drive with four wheels on the limbs [74]. DRC-Hubo was the winner of the DRC finals. This adaptable robot takes advantage of bipedal

walking and driving over flat terrain using wheels incorporated into its knee and ankles. DRC-Hubo transforms between a standing posture for walking and a kneeling posture for driving [75, 76]. As the Momaro robot can drive omnidirectional, it can move in restricted spaces. With six limbs, two are used for manipulation and the remaining four legs are utilized for locomotion [77].

As you can see from the robots mentioned above, a hybrid mobile platform provides a large, stable, and flexible support polygon, and the statically stable platform is advantageous for movement in various terrain. However, most importantly, these DRC robots do not attain all-terrain mobility on the challenging environments and do not have a high load carrying capacity for executing rescue missions.

Worldwide there are a few robots capable of lifting and transferring an injured person. As a modern rescue robot, a battlefield extraction assist robot (BEAR) is developed by Vecna Robotics [78]. It is known that BEAR can lift over $227kg$ using the hydraulic actuators. However, until now it has not shown the most crucial capability to lift a patient lying on the ground and to operate in a real unstructured environment as far as the author knows.

There is another well known mobile rescue robot whose name is RoNA (RObotic Nurse Assistant). The purpose is to lift and transfer a patient from a bed to chair and reposition the patient on the bed [79, 80]. It is able to lift up to $136kg$ and utilizes a holonomic mobile platform that move laterally and diagonally and rotate within its own footprint. However, this holonomic platform is usually suitable for only indoor environments, and the outdoor application is severely restricted. A nursing-care assistant robot RIBA (Robot for Interactive Body Assistance) is also working only in an indoor environment [81]. The configurations of these mobile rescue robots are shown in Figure 1.1.

In summary, these robots do not attain both dexterous manipulability and all-terrain mobility to operate in a highly unstructured environment.

1.4 Research Goals

In this dissertation, we propose new hardware design concepts and optimization-based control strategies for the rescue robot to reactively operate in a highly unstructured environment. We develop a regularized hierarchical quadratic program (RHQP) solver for real-time motion planning and control without resorting to any off-the-shelf optimization software for fast transcription of the objectives into the full-body joint space motion. Particularly, we are reasoning about task regularization to ensure the convergence and robustness of the solution in the face of singularity.

We find the robust optimal solution for hierarchical least-squares optimization under both equality and inequality tasks. As a singular value decomposition (SVD) centric approach is leveraged, all hierarchical solutions and Lagrange multipliers for properly handling the inequality constraints are analytically obtained in a recursive procedure. Moreover, we modify the active set method for fast computation, and the proposed algorithm works fast enough to be used as a practical means of real-time motion control system. A novel system design concepts of a rescue robot are presented. Although field robotics is concerned with the latest robotic engineering technologies, we concentrate on the core design concepts that can be shared with all other robots.

Throughout the thesis, we demonstrate the effectiveness of the proposed hierarchical optimization solver and the design concepts through extensive numerical simulations and experimental tests. We highlight precisely how the rescue robot

can successfully execute a casualty extraction and an explosive ordnance disposal in both indoor and outdoor environments that none of the existing robots has performed.

We now describe in more detail the main contributions of this thesis.

1.5 Contributions of This Thesis

1.5.1 Robust Hierarchical Task-Priority Control

We thoroughly investigate the underlying principles of regularized hierarchical optimization for real-time motion planning and control in a single hierarchical loop. While addressing all of the problems mentioned in Section 1.2, the RHQP solver guarantees the following advantages despite of the introduction of regularization.

- **Singularity Robustness**

For singularity robustness, task regularization is considered from the very beginning of the hierarchical optimization. We progressively explore the strategic components in a unified and consistent way to integrate regularization and inequality-constrained hierarchical optimization. This challenging problem is also posed in [16, 36], but the mixed problem is not fully discussed due to the mathematical complexity such as a non-convexity, regularization-induced errors, difficulty to find Lagrange multipliers, and not leaving redundant resources from higher to lower levels. To tackle the problems in a unified and computationally efficient way, a singular value decomposition centric approach is used. Therefore, the robust optimal solutions and Lagrange multipliers are recursively and analytically obtained.

- **Inequality Constraint Handling**

For what concerns the most important aspects of inequality constraints, it can be recognized that inequalities are active or not, depending on the situation. Because the resource or degrees of freedom should be determined to perform given active inequality tasks. A naive way to deal with the inequality constraint is to clamp or activate it as the equality constraint on the limit value to restrict any further motion if a constraint is violated [67]. However, when to relax the clamped constraint is problematic.

The RHQP solver is able to precisely control inequality constraints using a modified active set method with Lagrange multipliers. The modified algorithm is simple and fast because it does not use a step direction nor a step length usually used in numerical optimization. Moreover, it operates regardless of whether the initial point is feasible or not for an initial working set. The modified active set method focuses only on finding a blocking constraint and deactivating an unnecessary constraint.

- **Elimination of Cycling**

A *cycling* means that the active set method does not move from the current iterate, but returns the same solution and working set repeatedly to an earlier iteration. It causes an infinite sequence [16, 41, 55, 82, 83, 84]. The proposed algorithm finds the optimal solution and optimal active set without showing any cycling and deactivates an unnecessary constraint at the right time. We show that, without considering the regularization induced errors in Lagrange multipliers, the algorithms such as HQP and DLS-HQP result in a cycling in the neighborhood of singularity.

- **Top-Most Priority of Physical Inequality Constraints**

Generally, an artificial potential function can be used to indirectly treat inequality constraint by projecting the function's gradient in the null space of the higher priority tasks as the least priority objective. Even far from the constraints the potential function always influences on the whole DoF, and therefore it is difficult to impose a high priority on the potential function. For example, if joint angle ranges are expressed by the potential function with the top-most priority, and then there is no redundant resource for the lower level optimization. Top priority can be imposed on the potential function as the clamping method by activating the function on the limit value if a constraint is violated [85]. However, when to deactivate it is still ambiguous.

However, RHQP imposes the top-most priority on the physical inequality constraints such a joint operation range. This eliminates the possibility that the motion tasks violate the physical constraints.

- **Computational Efficiency**

As mentioned earlier in Section 1.2, the cascade-QP is computationally inefficient because the level k optimization problem is first solved in the k^{th} QP, and then repeatedly solved at each of the following levels as a constraint.

Indeed, one of the most attractive features of the RHQP solver is that it allows us to find the optimal solution very efficiently by considering all level's hierarchical tasks together with only one active set. The Lagrange multiplier matrix eliminates the iterative process used to find each level optimal active set.

All necessary Lagrange multiplier and optimal solution are obtained in a recursive procedure that exploits the result of the previous level to aid in obtaining the result for the current level. Consequently, under the equality and inequality constraints, the proposed algorithm works fast. As the control frequency is set at 1kHz for the rescue robot, the hierarchical optimization algorithm should provide the calculated results every 1 *ms*. The average computation time for a 19-DoF rescue robot is less than 0.1 *ms* on a typical personal computer.

- **Performance versus Robustness**

We mathematically show that regularization requires for us to trade off robustness against performance. Because the priority task can interfere with all following level tasks in spite of the absence of task confliction. This small coupling error induced by regularization is the only price we have to pay instead of guaranteeing convergence and robustness of the solution.

- **Generalization of Hierarchical Quadratic Programming**

We show that the optimal solutions of unregularized HQP in [44] can be exactly recovered from the RHQP solutions by setting each level regularization coefficient to zero. Therefore, we provide a generalization of the HQP framework.

1.5.2 Design Concepts of Robust Rescue Robot

The ultimate goal of the rescue robot, Hurcules (HUMANoid ResCUe robot for caLamity rESponse) is to extract an impaired person lying on the ground in the battlefields or hazardous environments and to transfer him or her to a safe place.

The other key mission is to dispose and transfer a dangerous or explosive object instead of humans.

The important design concepts are presented to realize how a mechanical structure can be developed to have an effect to both manipulation capability and all-terrain mobility. The Hurcules robot is uniquely designed for mobile manipulation with statically stable platform, so that the robot can execute complex manipulation tasks without concerning about balancing control. Yet the robot makes use of a stable tank like posture when the versatile mobility is required.

The upper body of the robot is designed humanoid in form with replaceable modularized dual arms. The lower body is featured with a hybrid tracked and legged platform to simultaneously acquire versatile manipulability and mobility on difficult terrain. As a result, the robot can efficiently perform a driving task, dangerous object manipulation, and casualty extraction missions by changing the pose and modularized equipments in an optimized manner.

1.5.3 Hierarchical Motion and Force Control

While maintaining all of the strength of RHQP as mentioned in 1.5.1, the RHQP algorithm is extended to motion and force control. Interaction force control is one of the crucial requirements for the successful manipulation tasks, and a resolved acceleration-based hierarchical approach is suitable for both motion and force control utilizing accelerations, velocities, and positions. While modern hierarchical solvers can handle motion and force control, the mixed problems of singularity robustness and inequality-constrained optimization are not thoroughly investigated [12, 13, 30, 31, 44, 71, 86, 87]. Nearly rank deficient Jacobians coming from kinematic or algorithmic singularities lead to numerical instability, excessive

joint accelerations, and large torques. These approaches may require that the singularity problem should be solved outside the control loop.

In order to address this problem in a single hierarchical control loop, we focus on the integration of the RHQP algorithm and the motion and force control schemes. A spring-mass-damper system is realized with the additional operational space inner position or velocity control loops. Then the force control is conducted according to this controlled dynamical system by closure of the force control loop. As often the force measurement is noisy, we do not directly use a derivative action of the force measurement. Instead, the stabilizing damping effect is obtained through the virtual damper implemented by the operational space velocity feedback. Therefore, the force control scheme requires a stabilizing PD control action based on velocity and position measurements as well as force error. To achieve zero steady-state force error, an integral action is often applied to the system. However, integral action can lead to the slow response, wind-up, and reduced stability margin. Thus, instead of resorting to the integral action, we reject steady-state force error by suitably designing the operational space dynamics.

The RHQP based motion and force control scheme is able to ensure singularity robustness and proper handling of the inequality constraint. The effectiveness is demonstrated through numerical simulations and experimental tests with the position-controlled Hércules robot. Consequently, a desired interaction force is regulated from the motion control schemes in conjunction with the outer force feedback loop so that the robot successfully extracts an injured lying on the ground.

1.6 Dissertation Preview

Dissertation is organized as shown in Figure 1.2. Chapter 2 briefly describes the task-priority motion planning and control framework. Chapter 3 is devoted to discussion on RHQP subject to both equality and inequality tasks. We recursively obtain the singularity robust optimal solution, the optimal task error, hierarchically decomposed Jacobian matrix, and Lagrange multipliers while resolving all difficulties resulting from regularization. In Chapter 4, we present the core design concepts of the Hircules rescue robot and then demonstrate the effectiveness of the proposed RHQP solver through extensive numerical simulations and experimental tests. The integration of the RHQP algorithm and the motion and force control laws is the topic of Chapters 5. In Chapter 6, we conclude this dissertation with a summary of our main results and discuss further research directions.

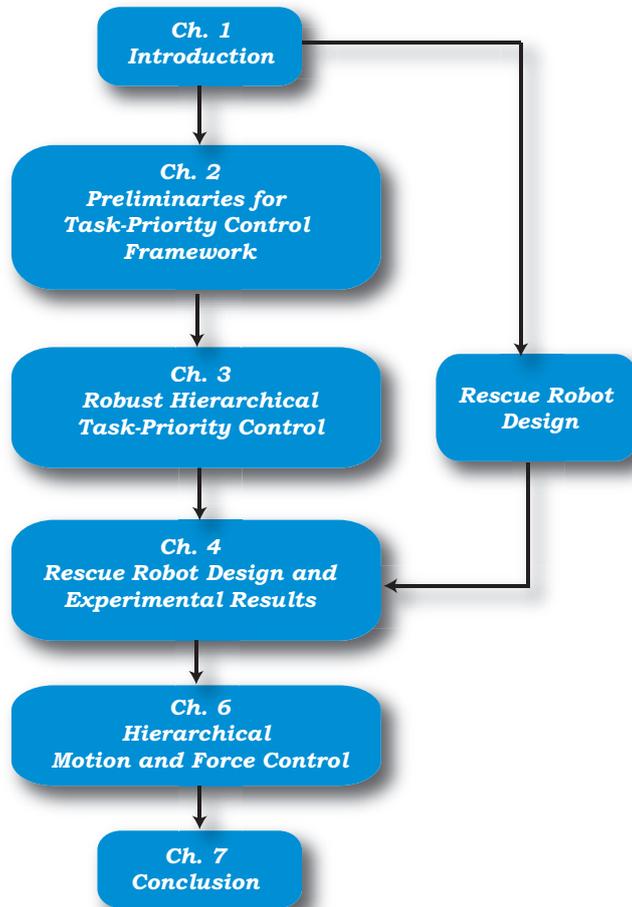


Figure 1.2: Organization of dissertation.

2

Preliminaries for Task-Priority Control Framework

2.1 Introduction

The task priority framework is suitable to handle the redundant problems for managing multiple tasks. We briefly describe a task-priority motion planning and control framework from an optimization perspective for further understanding of the proposed RHQP algorithm. By formulating the hierarchical problem in the least-squares optimization [43, 88], the optimal solution is derived in a unified and consistent way, and therefore it can be naturally extended to RHQP. We focus on finding the solution and null space projecting matrix in a recursive fashion to reduce the matrix dimension and number of usage of SVD for fast computation. To obtain the null space projecting matrix in a recursive form, Greville's method is leveraged. It provides a means to compute the Moore-Penrose generalized inverse by resorting to a finite iterative procedure [89].

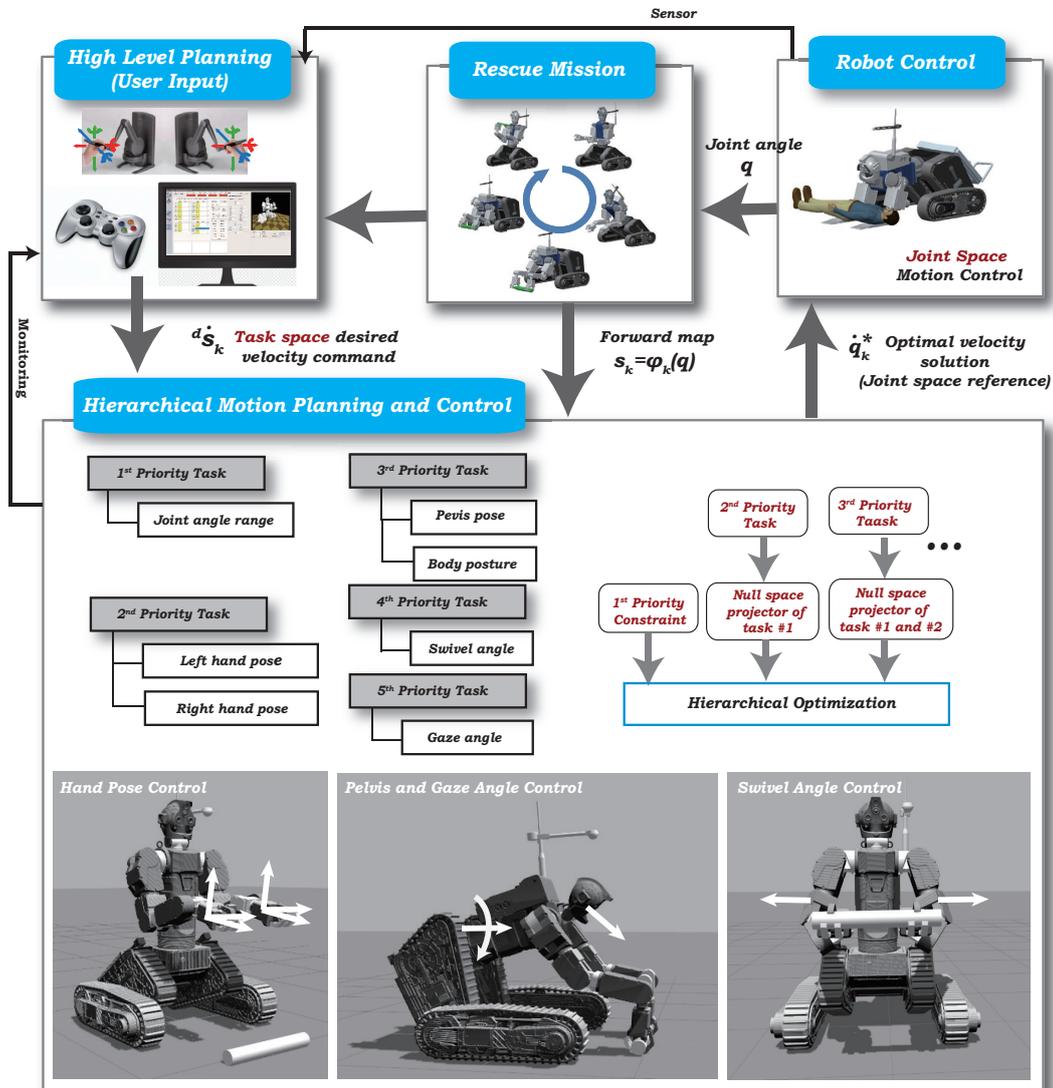


Figure 2.1: Real-time control block diagram for the hierarchical motion planning and control framework. As the control frequency is set at 1kHz for the Hurcules robot, a hierarchical optimization algorithm should provide the results every 1 *ms*.

2.2 Task-Priority Inverse Kinematics

The task-priority strategy has been widely used for controlling redundant manipulators and humanoid robots, where every task is accomplished as much as possible at different priority level [17, 18, 20, 27, 67]. As the hierarchical strategy is to establish priority between tasks, the lower level task produces only a self-motion which does not interfere with the fulfillment of the higher priority tasks. This implies that the lower priority task is satisfied only in the null space of the higher priority task, and the concept is extended to multiple tasks.

The schematic block diagram for the task priority framework is presented in Figure 2.1. The index k ($1 \leq k \leq p$) refers to the level, where 1 denotes the highest priority, and p is the least rank of priority or total number of levels. The index reflects the relative importance among tasks. As the control frequency is set at 1kHz for the Hurcules robot, a hierarchical optimization algorithm should provide the results every 1 *ms*. In what follows, we use the constraint, objective, and task interchangeably, and the effector designates a frame or bodies to which we impose tasks.

From now on, we explain the task-priority inverse kinematics problem from the least-squares optimization perspective. Considering the following general k level task characterized by the pose of an effector as a function of the joint variables

$$\mathbf{s}_k = \boldsymbol{\phi}_k(\mathbf{q}), \quad (2.2.1)$$

where the vector $\mathbf{s}_k \in \mathbb{R}^{m_k}$ is the position vector defined in the task space and the vector $\mathbf{q} \in \mathbb{R}^n$ is a joint angle. As the vector function $\boldsymbol{\phi}_k(\mathbf{q})$ is generally nonlinear, the linear differential relation is usually used

$$\dot{\mathbf{s}}_k = \mathbf{J}_k(\mathbf{q})\dot{\mathbf{q}} \quad (2.2.2)$$

with $\dot{\mathbf{q}}$ the joint velocity, $\dot{\mathbf{s}}_k$ is the task space velocity vector, and \mathbf{J}_k is the Jacobian matrix

$$\mathbf{J}_k(\mathbf{q}) = \frac{\partial \phi_k(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{m_k \times n}. \quad (2.2.3)$$

The Jacobian matrix allows the effector's linear and angular velocities to be described as a function of the joint variables with respect to the reference frame. The inverse kinematic problem is to find the joint variable $\dot{\mathbf{q}}$ corresponding to a given effector position and orientation $\dot{\mathbf{s}}_k$ in (2.2.2).

In the case of kinematic structures with a large number of degrees of freedom, solving a set of such linear equations (2.2.2) often appears to be under-determined, over-constrained or may be incompatible. The task hierarchy establishes a priority or hierarchy between the tasks, and it usually utilizes task Jacobian and its null space projection operators. Consequently, the hierarchy implies that each k level equation (2.2.2) is solved only in the null space of the priority tasks with the number of $\{1, \dots, (k-1)\}$.

Let us consider the following least-squares optimization problem with equality constraints

$$\min_{\dot{\mathbf{q}}_k} \frac{1}{2} \|\mathbf{J}_k \dot{\mathbf{q}}_k - {}^d \dot{\mathbf{s}}_k\|^2 \quad (2.2.4)$$

$$\text{subject to } \underline{\mathbf{J}}_{k-1} \dot{\mathbf{q}}_k = \underline{\mathbf{J}}_{k-1} \dot{\mathbf{q}}_{k-1}^* \quad (2.2.5)$$

where the vector ${}^d \dot{\mathbf{s}}_k \in \mathbb{R}^{m_k}$ is the desired velocity vector defined in the task space, $\dot{\mathbf{q}}_{k-1}^*$ is the optimal solution for $\{1, \dots, (k-1)\}$ tasks, and the augmented Jacobian matrix is

$$\underline{\mathbf{J}}_k = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_k \end{bmatrix}. \quad (2.2.6)$$

The optimization problem with the quadratic objective function (2.2.4) and the linear constraint (2.2.5) is called quadratic program (QP) [55, 56]. The problem now is to obtain the optimal solution $\dot{\mathbf{q}}_k^* \in \mathbb{R}^n$ that satisfies the constraint (2.2.5) and minimizes the objective function (2.2.4) in the least-squares sense. The constraint implies that the solution $\dot{\mathbf{q}}_k^*$ should not change all previous constraints represented by the augmented task Jacobian $\underline{\mathbf{J}}_{k-1}$ and the previous level optimal solution $\dot{\mathbf{q}}_{k-1}^*$. Thus hierarchical structure ensures that the j level task is not affected by the k level for all $j < k$.

A generic solution to this least-squares problem is

$$\dot{\mathbf{q}}_k = \dot{\mathbf{q}}_{k-1}^* + \mathbf{P}_{k-1} \mathbf{t}_k \quad (2.2.7)$$

where \mathbf{t}_k is an arbitrary parameter vector to satisfy additional objectives with the lower priority. $\mathbf{P}_{k-1} \in \mathbb{R}^{n \times n}$ is a null space projecting matrix of the augmented Jacobian $\underline{\mathbf{J}}_{k-1}$ and it is calculated by [18]

$$\mathbf{P}_k = \mathbf{I} - \underline{\mathbf{J}}_k^\dagger \underline{\mathbf{J}}_k \quad (2.2.8)$$

where $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix and $\underline{\mathbf{J}}_k^\dagger$ is the pseudo-inverse of a matrix $\underline{\mathbf{J}}_k$ satisfying four criteria also known as *Moore-Penrose conditions* [90, 91]

$$\underline{\mathbf{J}}_k \underline{\mathbf{J}}_k^\dagger \underline{\mathbf{J}}_k = \underline{\mathbf{J}}_k, \quad (2.2.9)$$

$$\underline{\mathbf{J}}_k^\dagger \underline{\mathbf{J}}_k \underline{\mathbf{J}}_k^\dagger = \underline{\mathbf{J}}_k^\dagger, \quad (2.2.10)$$

$$(\underline{\mathbf{J}}_k \underline{\mathbf{J}}_k^\dagger)^T = \underline{\mathbf{J}}_k \underline{\mathbf{J}}_k^\dagger, \quad (2.2.11)$$

$$(\underline{\mathbf{J}}_k^\dagger \underline{\mathbf{J}}_k)^T = \underline{\mathbf{J}}_k^\dagger \underline{\mathbf{J}}_k. \quad (2.2.12)$$

In order to find the solution, the constrained minimization problem can be

transformed into the unconstrained optimization problem by substituting the solution (2.2.7) into (2.2.4). The result is

$$\min_{\mathbf{t}_k} \frac{1}{2} \|\mathbf{J}_k \dot{\mathbf{q}}_{k-1}^* + \mathbf{J}_k \mathbf{P}_{k-1} \mathbf{t}_k - {}^d \dot{\mathbf{s}}_k\|^2. \quad (2.2.13)$$

The constraint (2.2.5) is satisfied by the solution (2.2.7). The solution \mathbf{t}_k^* minimizing (2.2.13) is directly obtained by taking the partial derivative with respect to \mathbf{t}_k . Then we have

$$\mathbf{t}_k^* = (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger {}^d \dot{\mathbf{s}}_k - (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger \mathbf{J}_k \dot{\mathbf{q}}_{k-1}^*. \quad (2.2.14)$$

Substituting (2.2.14) into (2.2.7) leads to the solution as follows:

$$\dot{\mathbf{q}}_k^* = \dot{\mathbf{q}}_{k-1}^* + \mathbf{P}_{k-1} (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger ({}^d \dot{\mathbf{s}}_k - \mathbf{J}_k \dot{\mathbf{q}}_{k-1}^*). \quad (2.2.15)$$

The matrix $\mathbf{P}_{k-1} (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger$ can be rewritten as:

$$\begin{aligned} \mathbf{P}_{k-1} (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger &= \mathbf{P}_{k-1} (\mathbf{J}_k \mathbf{P}_{k-1})^T ((\mathbf{J}_k \mathbf{P}_{k-1}) (\mathbf{J}_k \mathbf{P}_{k-1})^T)^{-1} \\ &= \mathbf{P}_{k-1} \mathbf{P}_{k-1} \mathbf{J}_k^T ((\mathbf{J}_k \mathbf{P}_{k-1}) (\mathbf{J}_k \mathbf{P}_{k-1})^T)^{-1} \\ &= \mathbf{P}_{k-1}^T \mathbf{J}_k^T ((\mathbf{J}_k \mathbf{P}_{k-1}) (\mathbf{J}_k \mathbf{P}_{k-1})^T)^{-1} \\ &= (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger, \end{aligned} \quad (2.2.16)$$

where \mathbf{P}_k is both symmetric ($\mathbf{P}_k^T = \mathbf{P}_k$) and idempotent ($\mathbf{P}_k \mathbf{P}_k = \mathbf{P}_k$). From (2.2.16), the equivalent solution to (2.2.15) would be

$$\dot{\mathbf{q}}_k^* = \dot{\mathbf{q}}_{k-1}^* + (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger ({}^d \dot{\mathbf{s}}_k - \mathbf{J}_k \dot{\mathbf{q}}_{k-1}^*). \quad (2.2.17)$$

This recursive solution has the same form proposed by [18].

For the justification of the special form of (2.2.17), we suppose neither \mathbf{J}_k nor $\mathbf{J}_k \mathbf{P}_{k-1}$ is singular. Let us consider k -th task. Substituting the solution (2.2.17)

into the k level task as follows:

$$\begin{aligned}\mathbf{J}_k \dot{\mathbf{q}}_k^* &= \mathbf{J}_k \dot{\mathbf{q}}_{k-1}^* + \mathbf{J}_k (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger ({}^d \dot{\mathbf{s}}_k - \mathbf{J}_k \dot{\mathbf{q}}_{k-1}^*) \\ &= {}^d \dot{\mathbf{s}}_k,\end{aligned}\quad (2.2.18)$$

where we use following relation by referring to $\mathbf{P}_{k-1} (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger = (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger$ in (2.2.16)

$$\mathbf{J}_k (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger = (\mathbf{J}_k \mathbf{P}_{k-1}) (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger = \mathbf{I}.\quad (2.2.19)$$

Hence, from (2.2.18), the k -th task is satisfied by the optimal solution (2.2.17).

Now think of $(k-1)$ -th task

$${}^d \dot{\mathbf{s}}_{k-1} = \mathbf{J}_{k-1} \dot{\mathbf{q}}_{k-1}^*.\quad (2.2.20)$$

The $(k-1)$ -th task should be satisfied if $\dot{\mathbf{q}}_{k-1}^*$ is replaced with $\dot{\mathbf{q}}_k^*$ as follows:

$${}^d \dot{\mathbf{s}}_{k-1} = \mathbf{J}_{k-1} \dot{\mathbf{q}}_k^*.\quad (2.2.21)$$

To prove this, we substitute the optimum (2.2.17) into the right hand side of (2.2.21), and we obtain the result as

$$\mathbf{J}_{k-1} \dot{\mathbf{q}}_{k-1}^* + \mathbf{J}_{k-1} (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger ({}^d \dot{\mathbf{s}}_k - \mathbf{J}_k \dot{\mathbf{q}}_{k-1}^*) = {}^d \dot{\mathbf{s}}_{k-1}\quad (2.2.22)$$

where

$$\mathbf{J}_{k-1} (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger = \mathbf{0}.\quad (2.2.23)$$

Since \mathbf{P}_{k-1} is the projector onto the null space of the Jacobian for $\{1, \dots, k-1\}$. Therefore, the optimal solution (2.2.17) dose not disturb the higher priority tasks.

A singularity occurs at which the task Jacobian is rank-deficient compared to its nominal rank, and the mobility is reduced. For example, if \mathbf{J}_k is singular by

itself, the configuration is termed a *kinematic singularity*. Otherwise, if each level Jacobian has full row rank but more than two Jacobians together become singular, it is referred to an *algorithmic singularity*. The corresponding tasks conflict [18, 68]. The incompatible objectives emerge in the form of linearly dependent rows of the Jacobian matrices.

The most serious problem of singularity is not at singular points themselves, but rather in the neighborhood of singular points. In the neighborhood of singular points, even for a small change of ${}^d\dot{\mathbf{s}}_k$, an excessive change of $\dot{\mathbf{q}}_k$ is often required. Generally, the inversion process inherently suffers from this sensitivity close to singular points. Moreover, the singularity can significantly reduce the reachable space of the robots.

2.3 Recursive Formulation of Null Space Projector

By leveraging SVD, we can find the null space projector in a recursive way to enhance a computational efficiency. The SVD of the matrix of $\mathbf{J}_k\mathbf{P}_{k-1} \in \mathbb{R}^{m_k \times n}$ in (2.2.17) is

$$\mathbf{J}_k\mathbf{P}_{k-1} = \begin{bmatrix} \mathbf{U}_k & \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_k^T \\ \mathbf{Z}_k^T \end{bmatrix} \quad (2.3.24)$$

$$= \mathbf{U}_k\boldsymbol{\Sigma}_k\mathbf{Y}_k^T \quad (2.3.25)$$

where $[\mathbf{U}_k \ \mathbf{V}_k]$ and $[\mathbf{Y}_k \ \mathbf{Z}_k]$ are orthonormal square matrices and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{l_k \times l_k}$ is the diagonal matrix composed of the singular values $\sigma_1, \dots, \sigma_{l_k}$. In terms of this decomposition, $\mathbf{U}_k \in \mathbb{R}^{m_k \times l_k}$ and $\mathbf{V}_k \in \mathbb{R}^{m_k \times (m_k - l_k)}$ are the bases of the column and left null space of $\mathbf{J}_k\mathbf{P}_{k-1}$, respectively. Each $\mathbf{Y}_k \in \mathbb{R}^{n \times l_k}$ and $\mathbf{Z}_k \in \mathbb{R}^{n \times (n - l_k)}$ corresponds to the basis of row and null space. As we do not assume that $\mathbf{J}_k\mathbf{P}_{k-1}$

is full rank, l_k may be less than both m_k and n . The expression of (2.3.25) is known as the thin SVD.

The pseudo-inverse is obtained directly:

$$(\mathbf{J}_k \mathbf{P}_{k-1})^\dagger = \begin{bmatrix} \mathbf{Y}_k & \mathbf{Z}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_k^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k^T \\ \mathbf{V}_k^T \end{bmatrix} \quad (2.3.26)$$

$$= \mathbf{Y}_k \boldsymbol{\Sigma}_k^{-1} \mathbf{U}_k^T \quad (2.3.27)$$

where the inverse matrix $\boldsymbol{\Sigma}_k^{-1}$ is cheaply obtained by inverting each main diagonal element of $\boldsymbol{\Sigma}_k$. The advantage of the SVD is that it gives a completely reliable means to solve least square problems even if the matrix does not have full rank [92] and also provides the condition number for us to conveniently monitor the feasibility of the solution.

It is computationally wasteful if we do not use any information of \mathbf{P}_{k-1} for acquiring \mathbf{P}_k . Thus we explore to obtain the recursive form of \mathbf{P}_k by using the result of the previous null space projector \mathbf{P}_{k-1} . The recursive calculation of \mathbf{P}_k is firstly proposed in [10, 67]. In this thesis, we derive it in a slightly different way for simplicity. Toward this recursive form we use Greville's method that provides a means to compute the Moore-Penrose generalized inverse [89].

At first, we need the matrix $\underline{\mathbf{J}}_k$ to be partitioned as

$$\underline{\mathbf{J}}_k = \begin{bmatrix} \underline{\mathbf{J}}_{k-1} \\ \mathbf{J}_k \end{bmatrix}. \quad (2.3.28)$$

Assume that $\mathbf{J}_k \mathbf{P}_{k-1} \neq \mathbf{0}$, and then generalized inverse of the partitioned matrix

can be written as follows [89]:

$$\begin{aligned}
\mathbf{P}_k &= \mathbf{I} - \underline{\mathbf{J}}_k^\dagger \underline{\mathbf{J}}_k \\
&= \mathbf{I} - \begin{bmatrix} \underline{\mathbf{J}}_{k-1} \\ \underline{\mathbf{J}}_k \end{bmatrix}^\dagger \begin{bmatrix} \underline{\mathbf{J}}_{k-1} \\ \underline{\mathbf{J}}_k \end{bmatrix} \\
&= \mathbf{I} - \begin{bmatrix} \underline{\mathbf{J}}_{k-1}^\dagger - (\mathbf{J}_k - \mathbf{J}_k \underline{\mathbf{J}}_{k-1}^\dagger \underline{\mathbf{J}}_{k-1})^\dagger \mathbf{J}_k \underline{\mathbf{J}}_{k-1}^\dagger & (\mathbf{J}_k - \mathbf{J}_k \underline{\mathbf{J}}_{k-1}^\dagger \underline{\mathbf{J}}_{k-1})^\dagger \\ & \begin{bmatrix} \underline{\mathbf{J}}_{k-1} \\ \underline{\mathbf{J}}_k \end{bmatrix} \end{bmatrix} \\
&= \mathbf{I} - \underline{\mathbf{J}}_{k-1}^\dagger \underline{\mathbf{J}}_{k-1} - \left(\mathbf{J}_k (\mathbf{I} - \underline{\mathbf{J}}_{k-1}^\dagger \underline{\mathbf{J}}_{k-1}) \right)^\dagger \mathbf{J}_k \underline{\mathbf{J}}_{k-1}^\dagger \underline{\mathbf{J}}_{k-1} + \left(\mathbf{J}_k (\mathbf{I} - \underline{\mathbf{J}}_{k-1}^\dagger \underline{\mathbf{J}}_{k-1}) \right)^\dagger \mathbf{J}_k \\
&= \mathbf{I} - \underline{\mathbf{J}}_{k-1}^\dagger \underline{\mathbf{J}}_{k-1} - (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger \mathbf{J}_k \underline{\mathbf{J}}_{k-1}^\dagger \underline{\mathbf{J}}_{k-1} + (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger \mathbf{J}_k \\
&= \mathbf{I} - \underline{\mathbf{J}}_{k-1}^\dagger \underline{\mathbf{J}}_{k-1} + (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger (\mathbf{J}_k \mathbf{P}_{k-1}) \\
&= \mathbf{P}_{k-1} - (\mathbf{J}_k \mathbf{P}_{k-1})^\dagger \mathbf{J}_k \mathbf{P}_{k-1}, \tag{2.3.29}
\end{aligned}$$

where Greville's method is exploited in the second line. If the matrix decomposition (2.3.25) and (2.3.27) are used, then from (2.3.29) more computationally efficient formula is acquired as follows:

$$\mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{Y}_k \mathbf{Y}_k^T. \tag{2.3.30}$$

In this way, the previous computational effort for \mathbf{P}_{k-1} is used to good advantage and not wasted. However, the computation of the null space projecting matrix given in (2.2.8) requires two SVD calculations to find $\underline{\mathbf{J}}_k^\dagger$ and $(\mathbf{J}_k \mathbf{P}_{k-1})^\dagger$. Moreover, the size of the augmented Jacobian $\underline{\mathbf{J}}_k$ increases as moving along the hierarchy.

It should be clear that the size of augmented projection matrix $\mathbf{P}_k \in \mathbb{R}^{n \times n}$ keeps a constant matrix size for all k , and the number of arithmetic operations for $\mathbf{J}_k \mathbf{P}_{k-1}$ do not decrease as the hierarchy advances sequentially (as k increases). In the next chapter, computationally more efficient method will be described by using the basis matrix where the computational effort decreases as k increases.

2.4 Conclusion

We described the task-priority algorithm to deal with multiple conflict tasks. The hierarchical inverse kinematics from the least-squares optimization perspective is presented for further understanding of the proposed RHQP algorithm. For computational efficiency, the null space projecting matrix is found in a recursive form based on the Greville's method. However, the usage of the null space basis to be described in the next chapter is more efficient.

The task-priority resolution strategy has been widely used for controlling redundant manipulators and humanoid robots. However, a significant drawback of the classical prioritization algorithm is its inability to explicitly treat inequality tasks. A modern optimization-based control can handle the inequality constrained problems by resorting to an off-the-shelf numerical optimizer, but it does not fully discuss the mixed problem of singularity robustness and inequality constrained problem while guaranteeing computational efficiency.

For real world robotic applications, the robustness and safety issues and the satisfaction of the physical constraints are undoubtedly the first and foremost problem that should be ensured. Because, near singular configurations, a desired task space motion can often require extremely large configuration space control input. If the singularity robustness is not fully guaranteed, it can damage the robot hardware and the environment or even inadvertently hurt humans.

We are going to directly face with the singularity robust optimization problem by developing a generic solver for regularized hierarchical least-squares optimization subject to both equality and inequality tasks. This proposed solver also

eliminates the possibility that the motion tasks violate the physical inequality constraints. We do not use any off-the-shelf QP solver to take advantage of the null-space projections for computational efficiency. In particular, the underlying principles of the hierarchical optimization are thoroughly investigated.

3

Robust Hierarchical Task-Priority Control

3.1 Introduction

Modern intelligent humanoid and articulated robots often attain their tree-like mechanical structure with a large number of DoF to simultaneously achieve diverse kinematic and dynamic tasks [9]. As redundant degree of freedom can be conveniently exploited to satisfy a number of tasks or constraints, it provides more dexterity and versatility to the robot. Singularity avoidance, obstacle avoidance, compliant motion, and posture control are some examples of the tasks to be additionally fulfilled along with the primary task [90].

The task space or operational space function approach proposes to design a motion in a space dedicated to the task to be performed. Once the task space motion is generated and then the remaining work is to translate this reference into the configuration space motion. The transcription makes it clear that a fast

and reliable solver is required for real-time motion planning and control. However, on-line motion planning and control problems become difficult to solve due to the complexity of the tree-like structure and various forms of constraints and tasks. Moreover, as the multiple tasks are seldom orthogonal, they lead to conflicts regarding to their fulfillment.

The use of hierarchical or task-priority strategy for redundancy resolution has been extensively studied since the early 1980s to precisely resolve conflicts by establishing a priority order among the tasks [17, 18, 91]. This method has been widely used for controlling redundant manipulators, humanoid robots, and animation characters, where every task is achieved as much as possible at different priority level without interfering with the fulfillment of the higher priority tasks [20, 21, 22, 67, 39, 24, 27, 30]. However, a significant drawback of the classical prioritization algorithm is its inability to explicitly incorporate inequality tasks.

A naive way to deal with an inequality constraint is to clamp or activate it as the equality constraint on the limit value to restrict any further configuration space motion if a constraint is violated [67]. However, when to relax the clamped constraint is problematic. An artificial potential function can be used to indirectly treat inequality constraint by projecting the function's gradient in the null space of the higher level tasks as the least priority objective. The potential function always influences on the whole DoF, and therefore it is difficult to have a high priority. For example, if joint angle ranges are expressed by the potential functions with the top-most priority, and then there is no redundant DoF for the lower level optimization. Even though we can impose top priority on the potential function as the clamping method by activating the potential function on the limit value [85], when to deactivate it is still ambiguous as well.

3.1.1 Motivations

Nowadays the rapid advances in computer technology certainly contribute to popularizing real-time optimization-based motion planning and control for complex robotic systems. The powerful off-the-shelf QP solver enables to numerically solve inequality-constrained hierarchical optimization problems [37, 38, 40, 41, 42]. This cascade-QP can be computationally inefficient [37, 38]. Because the level k optimization problem is first solved in the k^{th} QP, and then repeatedly solved at each of the following levels as a constraint. Furthermore, when advancing within the hierarchy, the computation time also increases because of the increased size of the augmented constraint and the iterative process used to find the optimal active set for each level QP. Moreover, at each level QP of the cascade, the optimal active set may be completely different.

For example, suppose that a humanoid robot manipulates a heavy object with dual arms in the front. When the dual arm task is solved, the center of mass inequality constraint is activated to keep the point inside the support polygon. In the next hierarchy, the pelvis is positioned on the back, and then the front center of mass position constraint is deactivated. The active set method may activate and deactivate a constraint several times when moving along the hierarchy or cascade. As a result, this repetition of all these iterative processes can be inefficient.

This computational inefficiency of the cascade-QP is tackled by hierarchical quadratic program (HQP) proposed by [43, 44]. The HQP algorithm exploits the prioritized solution computed using a complete orthogonal decomposition (COD) and the null space projections to efficiently solve the equality and inequality constrained QP. Thus it can find the solution more than five-times faster than the cascade-QP and even faster than a weighted least-squares optimization [44, 45].

The HQP method is known to be 10 times faster than the classical method of [18] for the equality constrained problem. Despite its outstanding computational efficiency, HQP is of limited utility in the real world because the singularity robustness is not guaranteed.

3.1.2 Objectives

We are going to find a robust optimal solution for on-line hierarchical least-squares optimization subject to both equality and inequality constraints. We focus our attention on reasoning about task regularization to ensure the convergence and robustness of the solution in the face of singularity. The mixed problem of regularization and inequality constrained hierarchical optimization has hardly been discussed due to the mathematical complexity such as a non-convexity, regularization-induced errors, difficulty to find Lagrange multipliers, and not leaving redundant resources from higher to lower levels. Task regularization is considered from the very beginning of the problem and we address the problem by formulating regularized hierarchical quadratic programming. Then we obtain the solution in a unified and computationally efficient way by leveraging singular value decomposition (SVD) and the modified active set method.

Especially, a singular value decomposition centric approach is leveraged, and therefore all hierarchical solutions and Lagrange multipliers for properly handling the inequality constraints are analytically obtained. We mathematically show that regularization requires for us to trade off robustness against performance, and the optimal solutions of unregularized HQP in [44] can be exactly recovered from the RHQP solutions by setting each level regularization coefficient to zero.

The RHQP solver allows us to find the optimal solution very efficiently by

considering all level's hierarchical tasks together with only one active set. The Lagrange multiplier matrix eliminates the iterative process used to find each level optimal active set. We propose the modified active set algorithm which is simple and fast. Because it does not use a step direction nor a step length usually used in numerical optimization.

The active set method for solving an inequality constrained QP is to classify inequality constraints into active and inactive constraints. The active constraints are treated as equality ones while the inactive constraints are ignored. Thus, this chapter is organized as follows. We begin by introducing the task function approach. The hierarchical optimization under the equality constraints is firstly investigated, and then inequality-constrained optimization problem is solved. The effectiveness and performance of the proposed RHQP solver are proved through numerical simulations with a simple mobile robot where the robustness and convergence of the solution are guaranteed in the face of singularity.

3.2 Task Function Approach

The aim of the task function approach or operational space control is to express the objective in the space dedicated to the task to be performed or output value of a sensor. When executing multiple tasks simultaneously, the corresponding task functions are grouped, sorted in priority order and then solved. Let $\mathbf{q} \in \mathbb{R}^n$ be the joint position or configuration vector of the robot and the task function $\mathbf{g}_k : \mathbb{R}^n \rightarrow \mathbb{R}^{m_k}$ be the control objective. The index k ($1 \leq k \leq p$) refers to the level, where 1 denotes the highest priority, and p is the least rank of priority or total number of levels. The index reflects the relative importance among tasks. In what follows, we use the constraint, objective, and task interchangeably, and the effector designates

a frame or bodies to which we impose tasks.

The task function is given by

$$\mathbf{g}_k := \mathbf{s}_k - {}^d\mathbf{s}_k, \quad (3.2.1)$$

where

$$\mathbf{s}_k = \phi_k(\mathbf{q}). \quad (3.2.2)$$

The vector ${}^d\mathbf{s}_k$ is the desired task such an effector pose and $\mathbf{s}_k \in \mathbb{R}^{m_k}$ is the position vector. The function $\phi_k(\mathbf{q})$ is a forward mapping from the configuration space to the task space. As the map $\phi_k(\mathbf{q})$ is often nonlinear function of \mathbf{q} without trivial inverse, a linearized function is used

$$\dot{\mathbf{g}}_k = \mathbf{J}_k(\mathbf{q})\dot{\mathbf{q}} - {}^d\dot{\mathbf{s}}_k, \quad (3.2.3)$$

where

$$\mathbf{J}_k(\mathbf{q}) = \frac{\partial \phi_k(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{m_k \times n} \quad (3.2.4)$$

is the task Jacobian and ${}^d\dot{\mathbf{s}}_k$ is the desired task space velocity. To bring the task function to zero, a proportional correction is introduced such as

$$\dot{\mathbf{g}}_k := -\mathbf{K}_k \mathbf{g}_k, \quad (3.2.5)$$

and we thus get the following ordinary differential equation [38, 93]

$$\mathbf{J}_k \dot{\mathbf{q}} = -\mathbf{K}_k \mathbf{g}_k + {}^d\dot{\mathbf{s}}_k, \quad (3.2.6)$$

where \mathbf{K}_k is a positive definite (usually diagonal) gain matrix, and $\dot{\mathbf{q}}$ is a feasible joint velocity to be determined. If the task is feasible, then the task function tends to zero along the trajectory

$$\dot{\mathbf{g}}_k + \mathbf{K}_k \mathbf{g}_k = \mathbf{0} \quad (3.2.7)$$

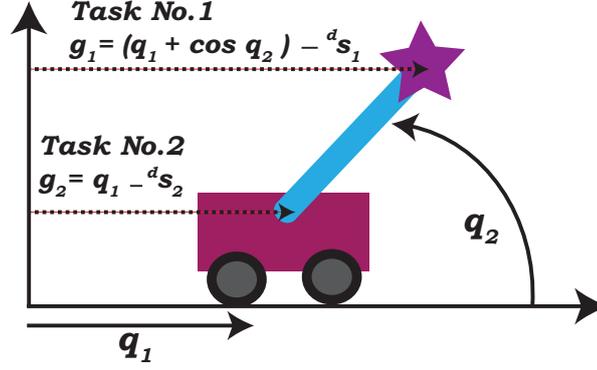


Figure 3.1: A simple robot contains a mobile base that allows to move along the horizontal axis with q_1 and an arm of unit length connected by means of a revolute joint with an angle q_2 . The first level objective is to reach a target point ${}^d s_1$ with the arm's horizontal position, and the second level task is to position the mobile base at a specified point ${}^d s_2$.

with a convergence rate that depends on the eigenvalues of matrix \mathbf{K}_k . It ensures the exponential convergence of the configuration vector \mathbf{q} to a limit \mathbf{q}^* while satisfying asymptotically $\mathbf{g}_k = \mathbf{0}$.

We can express a set of linear equality constraints or tasks in (3.2.6) as $\mathbf{J}_k \mathbf{x}_k = \mathbf{y}_k$ and inequalities as $\mathbf{J}_k \mathbf{x}_k \leq \mathbf{y}_k$. For the case of (3.2.6), $\mathbf{y}_k := -\mathbf{K}_k \mathbf{g}_k + {}^d \dot{\mathbf{s}}_k$ can be used. In fact, any collection of equality and inequality constraints with $=$, \geq , and \leq can be expressed in this form by simple rearrangement of the inequality.

To further clarify the concept of the task function, let us consider a simple mobile robot. The robot contains the mobile base to move along the horizontal axis with q_1 and an arm of unit length connected by means of a revolute joint with an angle q_2 as shown in Figure 3.1. Suppose two tasks are assigned with a desired effector's pose ${}^d \mathbf{s}$. The first priority task is to reach a target point ${}^d s_1$ with

the arm's horizontal position $s_1 = q_1 + \cos(q_2)$, and second level is to position the mobile base $s_2 = q_1$ at a specified point ${}^d s_2$ as close as possible. The task function in (3.2.1) is described as

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} (q_1 + \cos q_2) - {}^d s_1 \\ q_1 - {}^d s_2 \end{bmatrix}, \quad (3.2.8)$$

and the associated augmented Jacobian matrix is

$$\underline{\mathbf{J}}_2(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{bmatrix} = \begin{bmatrix} 1 & -\sin q_2 \\ 1 & 0 \end{bmatrix}. \quad (3.2.9)$$

The vector \mathbf{y} is specified as follows:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -K_1 g_1 + {}^d s_1 \\ -K_2 g_2 + {}^d s_2 \end{bmatrix}. \quad (3.2.10)$$

A singularity occurs at which the task Jacobian is rank-deficient compared to its nominal rank, and the mobility is reduced. For instance, if \mathbf{J}_k is singular by itself, the configuration is termed a *kinematic singularity*. Otherwise, if each level Jacobian has full row rank but more than two Jacobians together become singular, it is referred to an *algorithmic singularity*. These corresponding tasks conflict [18, 68]. The conflicting objectives emerges in the form of linearly dependent rows of the Jacobian matrices. For example, the joint angle $q_2 = 2j\pi$ ($j = 0, 1, 2, \dots$) in (3.2.9) renders the two Jacobians to be linearly dependent, and the two objectives tend to exploit the same joints so that there is no resource to satisfy multiple objectives simultaneously. In other words, the solution lies on algorithmic singularity.

The most serious problem of singularity is not at singular points themselves, but rather in the neighborhood of singular points. In the neighborhood of singular points, even for a small change of \mathbf{y}_k , an excessive change of \mathbf{x}_k is often required.

Generally, the inversion process inherently suffers from this sensitivity close to singular points. Therefore, the singularity can significantly reduce the reachable space of the robots.

3.3 Regularized Hierarchical Optimization with Equality Tasks

3.3.1 Regularized Hierarchical Optimization

We take into account task regularization from the beginning of hierarchical optimization. Let us consider a set of least-squares optimization with task regularization under equality constraints. The optimization problem is cast into the quadratic cost and the quadratic constraint as follows:

$$\text{Find } \mathbf{x}_k^* \in \underset{\mathbf{x}}{\text{argmin}} \quad \frac{1}{2} \|\mathbf{J}_k \mathbf{x} - \mathbf{y}_k\|_2^2 + \frac{1}{2} \rho_k^2 \|\mathbf{D}_k \mathbf{x}\|_2^2 \quad (3.3.11a)$$

$$\text{subject to } \mathbf{c}_j(\mathbf{x}) = \mathbf{c}_j(\mathbf{x}_j^*) \quad \forall j < k \quad (3.3.11b)$$

where

$$\mathbf{c}_j(\mathbf{x}) = \frac{1}{2} \|\mathbf{J}_j \mathbf{x} - \mathbf{y}_j\|_2^2 + \frac{1}{2} \rho_j^2 \|\mathbf{D}_j \mathbf{x}\|_2^2, \quad (3.3.12)$$

and \mathbf{D}_j is the selection matrix for regularization. The hierarchical structure ensures that the j level task is not affected by the k level for all $j < k$.

The hierarchical optimization (3.3.11) is to determine \mathbf{x}_k^* minimizing the Euclidean norm of the task error and the solution subject to the strict quadratic constraints (3.3.11b). The regularization parameter ρ_k controls the relative importance of the least-square error compared with the regularization term. Note that regularization involves adding a penalty to both objective function (3.3.11a) and constraints (3.3.11b). However, the reference [47] uses regularization only in the objective function, not in the constraints, to hand down remaining usable resources

to the next levels. Because if \mathbf{D}_j has full column rank, there is no resource to optimize the following level tasks. In other words, there is no degrees of freedom left afterwards. The regularizer should penalize each level task independently.

It is reasonable to consider the first level optimization alone and then relaxed second level optimization under the first level strict constraint. Then, we can obtain the overall solution of hierarchical optimization in a recursive fashion. The first level unconstrained optimization has the following cost function:

$$\mathbf{c}_1(\mathbf{x}) := \frac{1}{2} \mathbf{x}^T \mathbf{H}_1 \mathbf{x} - \mathbf{h}_1^T \mathbf{x} \quad (3.3.13)$$

with the associated Hessian

$$\mathbf{H}_1 := \mathbf{J}_1^T \mathbf{J}_1 + \rho_1^2 \mathbf{D}_1^T \mathbf{D}_1, \quad (3.3.14)$$

and the gradient vector

$$\mathbf{h}_1 := \mathbf{J}_1^T \mathbf{y}_1. \quad (3.3.15)$$

Among the possible \mathbf{x}_1 , the minimum length solution \mathbf{x}_1^* is obtained by using the generalized inverse [92]

$$\mathbf{x}_1^* = \mathbf{H}_1^\dagger \mathbf{h}_1 = (\mathbf{J}_1^T \mathbf{J}_1 + \rho_1^2 \mathbf{D}_1^T \mathbf{D}_1)^\dagger \mathbf{J}_1^T \mathbf{y}_1. \quad (3.3.16)$$

The Hessian matrix

$$\mathbf{H}_k := \mathbf{J}_k^T \mathbf{J}_k + \rho_k^2 \mathbf{D}_k^T \mathbf{D}_k \quad (3.3.17)$$

is usually used by assigning the regularization parameter $\rho_k \neq 0$ with $\mathbf{D}_k^T \mathbf{D}_k = \mathbf{I}_k$ to render the matrix positive definite where \mathbf{I}_k is the identity matrix. However, for a hierarchical optimization, the Hessian matrices of all levels should not have full column rank except for the least priority task. Because if $\mathbf{D}_k^T \mathbf{D}_k = \mathbf{I}_k$, then

there is no resource to optimize the following level tasks. Clearly, the regularization selection matrix \mathbf{D}_k should have the same row space basis to \mathbf{J}_k that contributes only to the level k to leave a parameter vector for lower level optimization. Later we will specify a particular form of \mathbf{D}_k , and now we suppose this condition is satisfied.

The next problem is to optimize an additional second level task by setting $k = 2$, $j = 1$ in (3.3.11). The constraint (3.3.11b) leads to a quadratically constrained quadratic programming, but it can be relaxed to the linear constraint [47]. Suppose the solution has the form of

$$\mathbf{x}_k = \mathbf{x}_{k-1}^* + \hat{\mathbf{x}}_k, \quad (3.3.18)$$

where $\hat{\mathbf{x}}_k$ is used to optimize k level objective function. With the solution (3.3.18), the quadratic constraint $\mathbf{c}_1(\mathbf{x}_2) = \mathbf{c}_1(\mathbf{x}_1^*)$ is respected if the following condition is satisfied

$$\frac{1}{2}\hat{\mathbf{x}}_2^T \mathbf{H}_1 \hat{\mathbf{x}}_2 + \mathbf{x}_1^{*T} \mathbf{H}_1 \hat{\mathbf{x}}_2 - \mathbf{h}_1^T \hat{\mathbf{x}}_2 = 0. \quad (3.3.19)$$

A sufficient condition to hold (3.3.19) is to choose $\hat{\mathbf{x}}_2$ in the null space of \mathbf{H}_1 and \mathbf{h}_1^T by making all three terms zeros. That is

$$\hat{\mathbf{x}}_2 = \mathbf{Z}_1(\mathbf{H}_1, \mathbf{h}_1^T) \mathbf{t}_2, \quad (3.3.20)$$

where \mathbf{t}_2 is an arbitrary parameter vector and \mathbf{Z}_1 is the null space basis of \mathbf{H}_1 and \mathbf{h}_1^T . In general, the solution (3.3.18) under the condition (3.3.20) is suboptimal for the regularized hierarchical optimization due to ensuring only sufficient condition in (3.3.19), but it respects hierarchical structure.

The second level optimization problem can be reformulated as

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{J}_2 \mathbf{x} - \mathbf{y}_2\|_2^2 + \frac{1}{2} \rho_2^2 \|\mathbf{D}_2 \mathbf{x}\|_2^2 \quad (3.3.21)$$

$$\text{subject to } \mathbf{x} = \mathbf{x}_1^* + \mathbf{Z}_1 \mathbf{t}_2. \quad (3.3.22)$$

Substituting (3.3.22) into (3.3.21) leads to the unconstrained optimization, and the optimal solution is obtained which minimizes $\mathbf{c}_1(\mathbf{x})$ at best and $\mathbf{c}_2(\mathbf{x})$ if possible. That is

$$\mathbf{x}_2^* = \mathbf{x}_1^* + \mathbf{Z}_1 \mathbf{t}_2^*, \quad (3.3.23)$$

where

$$\begin{aligned} \mathbf{t}_2^* &= \left((\mathbf{J}_2 \mathbf{Z}_1)^T (\mathbf{J}_2 \mathbf{Z}_1) + \rho_2^2 \mathbf{Z}_1^T \mathbf{D}_2^T \mathbf{D}_2 \mathbf{Z}_1 \right)^\dagger \\ &\quad \times \left((\mathbf{J}_2 \mathbf{Z}_1)^T (\mathbf{y}_2 - \mathbf{J}_2 \mathbf{x}_1^*) - \rho_2^2 \mathbf{Z}_1^T \mathbf{D}_2^T \mathbf{D}_2 \mathbf{x}_1^* \right). \end{aligned} \quad (3.3.24)$$

This solution can be generalized to an arbitrary number of tasks, and the following recursive solution is given by

$$\begin{aligned} \mathbf{x}_p^* &= \sum_{k=1}^p \mathbf{Z}_{k-1} \left((\mathbf{J}_k \mathbf{Z}_{k-1})^T (\mathbf{J}_k \mathbf{Z}_{k-1}) + \rho_k^2 \mathbf{Z}_{k-1}^T \mathbf{D}_k^T \mathbf{D}_k \mathbf{Z}_{k-1} \right)^\dagger \\ &\quad \times \left((\mathbf{J}_k \mathbf{Z}_{k-1})^T (\mathbf{y}_k - \mathbf{J}_k \mathbf{x}_{k-1}^*) - \rho_k^2 \mathbf{Z}_{k-1}^T \mathbf{D}_k^T \mathbf{D}_k \mathbf{x}_{k-1}^* \right) \end{aligned} \quad (3.3.25)$$

where \mathbf{Z}_0 is an $n \times n$ identity matrix. The vector $\hat{\mathbf{x}}_k$ in (3.3.18) is

$$\hat{\mathbf{x}}_k = \mathbf{Z}_{k-1} (\underline{\mathbf{H}}_{k-1}, \underline{\mathbf{h}}_{k-1}^T) \mathbf{t}_k \quad (3.3.26)$$

where $\underline{\mathbf{H}}_{k-1}$ and $\underline{\mathbf{h}}_{k-1}$ are the stacked matrix and the vector augmented from level 1 to $k-1$ and they are empty for $k-1=0$ such as

$$\underline{\mathbf{H}}_{k-1} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_{k-1} \end{bmatrix}. \quad (3.3.27)$$

In particular, the unregularized solution is recovered from the regularized optimal solution (3.3.25) with $\rho_k = 0$ as follows:

$$\mathbf{x}_p^* = \sum_{k=1}^p \left\{ \mathbf{Z}_{k-1} (\mathbf{J}_k \mathbf{Z}_{k-1})^\dagger (\mathbf{y}_k - \mathbf{J}_k \mathbf{x}_{k-1}^*) \right\}. \quad (3.3.28)$$

This unregularized solution (3.3.28) is exactly same to the well-known formula proposed by [44]. Hence, we provide a generalization of the HQP framework.

3.3.2 Optimal Solution

Now we shall express the optimal solution \mathbf{x}_k^* more compactly while obtaining \mathbf{Z}_k and \mathbf{D}_k recursively. The SVD provides a consistent solution for a least-squares problem whether the matrix attains dependent columns or dependent rows [92].

The SVD of $\mathbf{J}_k \mathbf{Z}_{k-1} \in \mathbb{R}^{m_k \times o_k}$ in the optimal solution (3.3.25) is given by

$$\mathbf{J}_k \mathbf{Z}_{k-1} = \begin{bmatrix} \mathbf{U}_k & \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{Y}}_k^T \\ \tilde{\mathbf{Z}}_k^T \end{bmatrix} \quad (3.3.29)$$

$$= \mathbf{U}_k \boldsymbol{\Sigma}_k \tilde{\mathbf{Y}}_k^T \quad (3.3.30)$$

where $[\mathbf{U}_k \ \mathbf{V}_k]$ and $[\tilde{\mathbf{Y}}_k \ \tilde{\mathbf{Z}}_k]$ are orthonormal square matrices. Thus following relations hold

$$\mathbf{U}_k \mathbf{U}_k^T + \mathbf{V}_k \mathbf{V}_k^T = \mathbf{I}_k, \quad (3.3.31a)$$

$$\mathbf{U}_k^T \mathbf{U}_k = \mathbf{I}_k, \quad \mathbf{V}_k^T \mathbf{V}_k = \mathbf{I}_k, \quad (3.3.31b)$$

$$\mathbf{U}_k^T \mathbf{V}_k = \mathbf{0}, \quad \mathbf{V}_k^T \mathbf{U}_k = \mathbf{0}, \quad (3.3.31c)$$

where \mathbf{I}_k is the identity matrix with an appropriate dimension. The diagonal matrix $\boldsymbol{\Sigma}_k$ is composed of the singular values $\sigma_1, \dots, \sigma_{l_k}$. As we do not assume that $\mathbf{J}_k \mathbf{Z}_{k-1}$ is full rank, l_k may be less than both m_k and o_k . In terms of this decomposition, $\mathbf{U}_k \in \mathbb{R}^{m_k \times l_k}$ and $\mathbf{V}_k \in \mathbb{R}^{m_k \times (m_k - l_k)}$ are the bases of the column and left

null space of $\mathbf{J}_k \mathbf{Z}_{k-1}$, respectively. Each $\tilde{\mathbf{Y}}_k \in \mathbb{R}^{o_k \times l_k}$ and $\tilde{\mathbf{Z}}_k \in \mathbb{R}^{o_k \times (o_k - l_k)}$ corresponds to the basis of row and null space. The expression (3.3.30) is known as the thin SVD and $\mathbf{J}_k \mathbf{Z}_{k-1}$ is a projection of the k -th Jacobian into the all previous null-space.

The pseudo-inverse of $\mathbf{J}_k \mathbf{Z}_{k-1}$ is obtained directly:

$$(\mathbf{J}_k \mathbf{Z}_{k-1})^\dagger = \begin{bmatrix} \tilde{\mathbf{Y}}_k & \tilde{\mathbf{Z}}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_k^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k^T \\ \mathbf{V}_k^T \end{bmatrix} \quad (3.3.32)$$

$$= \tilde{\mathbf{Y}}_k \boldsymbol{\Sigma}_k^{-1} \mathbf{U}_k^T \quad (3.3.33)$$

where the inverse matrix $\boldsymbol{\Sigma}_k^{-1}$ is obtained by simply inverting each main diagonal element of $\boldsymbol{\Sigma}_k$. The pseudo-inverse matrix in (3.3.33) satisfies four criteria also known as *Moore-Penrose conditions* (2.2.9)-(2.2.12).

New basis matrices are defined as follows:

$$\begin{bmatrix} \mathbf{Y}_k & \mathbf{Z}_k \end{bmatrix} = \mathbf{Z}_{k-1} \begin{bmatrix} \tilde{\mathbf{Y}}_k & \tilde{\mathbf{Z}}_k \end{bmatrix}, \quad (3.3.34)$$

where \mathbf{Y}_k is $n \times l_k$ matrix and \mathbf{Z}_k is $n \times (n - l_1 - l_2 \cdots - l_k)$ matrix. If we specify the regularization matrix \mathbf{D}_k in (3.3.11a) as

$$\mathbf{D}_k := \mathbf{Y}_k^T \quad (3.3.35)$$

then regularization acts on the parameter vector of the row space participating only in the k level tasks. Thus, it regularizes each level task independently and leaves the null space basis denoted by \mathbf{Z}_k , which can be used to optimize the next level objectives. This regularization matrix \mathbf{D}_k in (3.3.35) and the pseudo-inverse

$(\mathbf{J}_k \mathbf{Z}_{k-1})^\dagger$ in (3.3.33) enable the optimal solution \mathbf{x}_k^* of (3.3.25) to be much compactly expressed as

$$\mathbf{x}_k^* = \sum_{j=1}^k \left(\mathbf{Y}_j \hat{\Sigma}_j^{-1} \mathbf{U}_j^T (\mathbf{y}_j - \mathbf{J}_j \mathbf{x}_{j-1}^*) \right) \quad (3.3.36)$$

$$= \mathbf{x}_{k-1}^* + \underbrace{\mathbf{Y}_k \hat{\Sigma}_k^{-1} \mathbf{U}_k^T (\mathbf{y}_k - \mathbf{J}_k \mathbf{x}_{k-1}^*)}_{\hat{\mathbf{x}}_k^*} \quad (3.3.37)$$

where

$$\hat{\Sigma}_k^{-1} := (\Sigma_k^2 + \rho_k^2 \mathbf{I}_k)^\dagger \Sigma_k, \quad (3.3.38)$$

$\mathbf{Y}_k^T \mathbf{x}_{k-1}^* = \mathbf{0}$ are used, and $\mathbf{I}_k \in \mathbb{R}^{l_k \times l_k}$ is an identity matrix. The i^{th} singular value of the matrix $\hat{\Sigma}_k^{-1}$ is

$$\hat{\sigma}_{k,i} = \frac{\sigma_{k,i}}{\sigma_{k,i}^2 + \rho_k^2}, \quad 1 \leq i \leq l_k, \quad (3.3.39)$$

where $\sigma_{k,i}$ is the i^{th} diagonal component of Σ_k . Without the regularization parameter ρ_k , $\hat{\sigma}_{k,i}$ would grow unboundedly as $\sigma_{k,i}$ approaches zero near a singularity as shown in Figure 3.2.

Note that the unregularized solution of (3.3.28) can be obtained from the regularized solution (3.3.37) by replacing $\hat{\Sigma}_k^{-1}$ with Σ_k^{-1} . The vector $\hat{\mathbf{x}}_k^*$ of (3.3.37) contributes to optimize the k level task. It simultaneously mitigates the influence of the high level tasks by removing its contribution from the current task by utilizing $\mathbf{Y}_k \hat{\Sigma}_k^{-1} \mathbf{U}_k^T \mathbf{J}_k \mathbf{x}_{k-1}^*$. As we will see in the following optimal task error, this term does not perfectly compensate the effect caused by the priority tasks due to regularization. Therefore, we always have a regularization error as long as $\rho_k \neq 0$ is used. The optimum $\hat{\mathbf{x}}_k^*$ in (3.3.37) is also acquired as

$$\hat{\mathbf{x}}_k^* = \mathbf{Y}_k \mathbf{Y}_k^T \mathbf{x}_k^* \quad (3.3.40)$$

$$= \mathbf{Y}_k \hat{\Sigma}_k^{-1} \mathbf{U}_k^T (\mathbf{y}_k - \mathbf{J}_k \mathbf{x}_{k-1}^*) \quad (3.3.41)$$

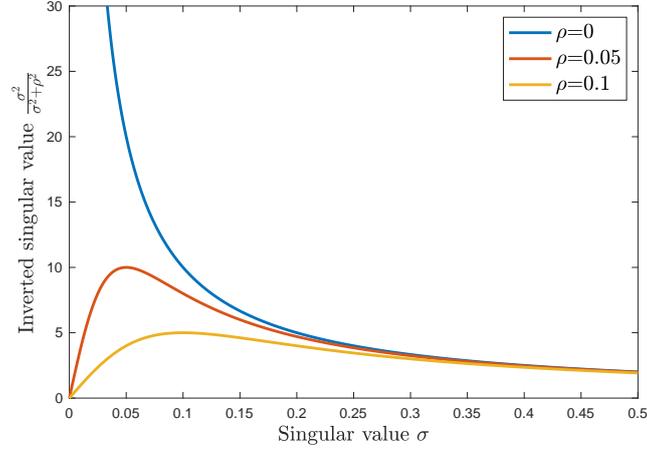


Figure 3.2: Comparison of the reciprocal singular values in (3.3.39). If the regularization parameter is zero, the singular value $\hat{\sigma}_{k,i}$ would grow unboundedly as $\sigma_{k,i}$ approaches zero near a singularity.

where $\mathbf{Y}_k^T \mathbf{x}_{k-1}^* = \mathbf{0}$ is used.

It is worth remarking that as the hierarchy advances sequentially (as k increases), the number of columns in \mathbf{Z}_k decreases, and thereby the computation of the SVD for the matrix product $\mathbf{J}_k \mathbf{Z}_{k-1} \in \mathbb{R}^{m_k \times (n-l_1-l_2-\dots-l_{k-1})}$ becomes cheaper. Thus, exploiting $\mathbf{J}_k \mathbf{Z}_{k-1}$ is computationally more efficient than using $\mathbf{J}_k \mathbf{P}_{k-1}$ because the null space projecting matrix

$$\mathbf{P}_k = \mathbf{Z}_k \mathbf{Z}_k^T \in \mathbb{R}^{n \times n} \quad (3.3.42)$$

has a constant matrix size. Please refer to the Section 2.2 for a detailed explanation.

3.3.3 Task Error and Hierarchical Matrix Decomposition

The aim of this section is to find the optimal task error and a hierarchical matrix decomposition recursively which will be used to find Lagrange multipliers for controlling the inequality constraints.

From the general optimal solution (3.3.37), the first level solution is obtained

$$\mathbf{x}_1^* = \mathbf{Y}_1 \hat{\Sigma}_1^{-1} \mathbf{U}_1^T \mathbf{y}_1. \quad (3.3.43)$$

Then the corresponding optimal task error vector can be acquired using the orthonormal property of (3.3.2)

$$\begin{aligned} \mathbf{e}_1^* &= \mathbf{J}_1 \mathbf{x}_1^* - \mathbf{y}_1 \\ &= \mathbf{U}_1 \Sigma_1 \hat{\Sigma}_1^{-1} \mathbf{U}_1^T \mathbf{y}_1 - \mathbf{y}_1 \\ &= (\mathbf{U}_1 \Sigma_1 \hat{\Sigma}_1^{-1} \mathbf{U}_1^T - \mathbf{U}_1 \mathbf{U}_1^T - \mathbf{V}_1 \mathbf{V}_1^T) \mathbf{y}_1 \\ &= -\mathbf{V}_1 \mathbf{V}_1^T \mathbf{y}_1 - \mathbf{U}_1 \mathbf{T}_1 \mathbf{U}_1^T \mathbf{y}_1, \end{aligned} \quad (3.3.44)$$

where

$$\mathbf{T}_k := \mathbf{I}_k - \Sigma_k \hat{\Sigma}_k^{-1}. \quad (3.3.45)$$

The i^{th} diagonal term of the matrix \mathbf{T}_k is

$$T_{k,i} = \frac{\rho_k^2}{\sigma_{k,i}^2 + \rho_k^2}, \quad 1 \leq i \leq l_k. \quad (3.3.46)$$

The first term of task error in (3.3.44) associated with \mathbf{V}_1 represents the unrealizable part of the task due to kinematic singularity. Because the top priority task cannot be influenced by any other tasks. The second term corresponds to the regularization error resulting from penalizing the solution, and it cannot be $\mathbf{0}$ if $\rho_1 \neq 0$.

For the second level, SVD of $\mathbf{J}_2\mathbf{Z}_1$ in (3.3.25) is

$$\mathbf{J}_2\mathbf{Z}_1 = \begin{bmatrix} \mathbf{U}_2 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{Y}}_2^T \\ \tilde{\mathbf{Z}}_2^T \end{bmatrix} \quad (3.3.47)$$

$$= \mathbf{U}_2\boldsymbol{\Sigma}_2\tilde{\mathbf{Y}}_2^T. \quad (3.3.48)$$

The basis $[\mathbf{U}_2 \ \mathbf{V}_2]$ decomposes the range space of $\mathbf{J}_2\mathbf{Z}_1 \in \mathbb{R}^{m_2 \times (n-l_1)}$ along its column space and the orthogonal to it. We define another basis (3.3.34) as follows:

$$\begin{bmatrix} \mathbf{Y}_2 & \mathbf{Z}_2 \end{bmatrix} := \mathbf{Z}_1 \begin{bmatrix} \tilde{\mathbf{Y}}_2 & \tilde{\mathbf{Z}}_2 \end{bmatrix}, \quad (3.3.49)$$

which provides bases of the row and null space of $\mathbf{J}_2\mathbf{Z}_1$. As the matrix $\mathbf{Y}_2 \in \mathbb{R}^{n \times l_2}$ is the basis of the row space, it can be used to optimize the current second level task. The matrix $\mathbf{Z}_2 \in \mathbb{R}^{n \times (n-l_1-l_2)}$ is the null space basis of both \mathbf{J}_1 and \mathbf{J}_2 so that it would be exploited for optimizing the next level.

As we specify the regularization matrix as $\mathbf{D}_2 = \mathbf{Y}_2^T$, the optimum \mathbf{x}_2^* can be rewritten from the optimal solution (3.3.37)

$$\mathbf{x}_2^* = \mathbf{x}_1^* + \underbrace{\mathbf{Y}_2\hat{\boldsymbol{\Sigma}}_2^{-1}\mathbf{U}_2^T(\mathbf{y}_2 - \mathbf{J}_2\mathbf{x}_1^*)}_{\hat{\mathbf{x}}_2^*} \quad (3.3.50)$$

$$= \begin{bmatrix} \mathbf{Y}_1 & \mathbf{Y}_2 \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1^T \mathbf{x}_1^* \\ \mathbf{Y}_2^T \hat{\mathbf{x}}_2^* \end{bmatrix} \quad (3.3.51)$$

where $\mathbf{Y}_1\mathbf{Y}_1^T\mathbf{x}_1^* = \mathbf{x}_1^*$ is used. By using (3.3.51), the optimal task error \mathbf{e}_2^* can be

acquired as follows:

$$\mathbf{e}_2^* = \mathbf{J}_2 \mathbf{x}_2^* - \mathbf{y}_2 \quad (3.3.52)$$

$$= \begin{bmatrix} \mathbf{J}_2 \mathbf{Y}_1 & \mathbf{J}_2 \mathbf{Y}_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1^T \mathbf{x}_1^* \\ \mathbf{Y}_2^T \mathbf{x}_2^* \\ t_3 \end{bmatrix} - \mathbf{y}_2 \quad (3.3.53)$$

$$= \underbrace{\begin{bmatrix} \mathbf{U}_2 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{E}_2 & \boldsymbol{\Sigma}_2 & \mathbf{0} \\ \mathbf{F}_2 & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{J}_2} \begin{bmatrix} \mathbf{Y}_1^T \\ \mathbf{Y}_2^T \\ \mathbf{Z}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1 & \mathbf{Y}_2 & \mathbf{Z}_2 \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1^T \mathbf{x}_1^* \\ \mathbf{Y}_2^T \mathbf{x}_2^* \\ t_3 \end{bmatrix} - \mathbf{y}_2 \quad (3.3.54)$$

where $\mathbf{Y}_2^T \mathbf{x}_1^* = \mathbf{0}$ is used. The matrices

$$\mathbf{E}_2 := \mathbf{U}_2^T \mathbf{J}_2 \mathbf{Y}_1 \quad (3.3.55)$$

$$\mathbf{F}_2 := \mathbf{V}_2^T \mathbf{J}_2 \mathbf{Y}_1 \quad (3.3.56)$$

are newly defined, and the relations $\mathbf{J}_2 \mathbf{Y}_2 = \mathbf{J}_2 \mathbf{Z}_1 \tilde{\mathbf{Y}}_2 = \mathbf{U}_2 \boldsymbol{\Sigma}_2$ is used. From (3.3.54), we can see that the hierarchically decomposed matrix of \mathbf{J}_2 .

Proceeding from the first to second level solution and matrix decomposition, a recursive formulation can be established to include any number of tasks with a decreasing order of priority. The optimum (3.3.37) can be rewritten as

$$\mathbf{x}_k^* = \begin{bmatrix} \underline{\mathbf{Y}}_{k-1} & \mathbf{Y}_k \end{bmatrix} \begin{bmatrix} \underline{\mathbf{Y}}_{k-1}^T \mathbf{x}_{k-1}^* \\ \mathbf{Y}_k^T \mathbf{x}_k^* \end{bmatrix}, \quad (3.3.57)$$

where

$$\underline{\mathbf{Y}}_k := \begin{bmatrix} \mathbf{Y}_1 & \cdots & \mathbf{Y}_k \end{bmatrix} \quad (3.3.58)$$

is defined and $\underline{\mathbf{Y}}_{k-1} \underline{\mathbf{Y}}_{k-1}^T \mathbf{x}_{k-1}^* = \mathbf{x}_{k-1}^*$ and (3.3.40) are used. The optimal task error \mathbf{e}_k^* is obtained by using (3.3.57) as follows:

$$\mathbf{e}_k^* = \mathbf{J}_k \mathbf{x}_k^* - \mathbf{y}_k \quad (3.3.59)$$

$$= \begin{bmatrix} \mathbf{J}_k \underline{\mathbf{Y}}_{k-1} & \mathbf{J}_k \mathbf{Y}_k & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{Y}}_{k-1}^T \mathbf{x}_{k-1}^* \\ \mathbf{Y}_k^T \mathbf{x}_k^* \\ \mathbf{t}_{k+1} \end{bmatrix} - \mathbf{y}_k \quad (3.3.60)$$

$$= \underbrace{\begin{bmatrix} \mathbf{U}_k & \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \mathbf{E}_k & \boldsymbol{\Sigma}_k & \mathbf{0} \\ \mathbf{F}_k & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{Y}}_{k-1}^T \\ \mathbf{Y}_k^T \\ \mathbf{Z}_k^T \end{bmatrix}}_{\mathbf{J}_k} \begin{bmatrix} \underline{\mathbf{Y}}_{k-1} & \mathbf{Y}_k & \mathbf{Z}_k \end{bmatrix} \begin{bmatrix} \underline{\mathbf{Y}}_{k-1}^T \mathbf{x}_{k-1}^* \\ \mathbf{Y}_k^T \mathbf{x}_k^* \\ \mathbf{t}_{k+1} \end{bmatrix} - \mathbf{y}_k, \quad (3.3.61)$$

where

$$\mathbf{E}_k := \mathbf{U}_k^T \mathbf{J}_k \underline{\mathbf{Y}}_{k-1}, \quad (3.3.62)$$

$$\mathbf{F}_k := \mathbf{V}_k^T \mathbf{J}_k \underline{\mathbf{Y}}_{k-1}. \quad (3.3.63)$$

The relation $\mathbf{J}_k \mathbf{Y}_k = \mathbf{J}_k \mathbf{Z}_{k-1} \tilde{\mathbf{Y}}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k$ and the orthonormal property of (3.3.2) are used.

In (3.3.61), we can see that the hierarchically decomposed matrix of \mathbf{J}_k for any $k \geq 1$

$$\mathbf{J}_k = \mathbf{W}_k \mathbf{S}_k \mathbf{Y}_k^T, \quad (3.3.64)$$

where

$$\mathbf{W}_k := \begin{bmatrix} \mathbf{U}_k & \mathbf{V}_k \end{bmatrix}, \quad (3.3.65)$$

$$\mathbf{S}_k := \begin{bmatrix} \mathbf{E}_k & \boldsymbol{\Sigma}_k \\ \mathbf{F}_k & \mathbf{0} \end{bmatrix}. \quad (3.3.66)$$

The matrix \mathbf{U}_k is the range or reachable space basis, and thus the task associated with this basis is realizable. However, the objective corresponding to \mathbf{V}_k cannot be fulfilled. The hierarchically decomposed Jacobian matrix of (3.3.64) can be rewritten as

$$\mathbf{J}_k = (\mathbf{V}_k \mathbf{F}_k + \mathbf{U}_k \mathbf{E}_k) \mathbf{Y}_{k-1}^T + \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{Y}_k^T. \quad (3.3.67)$$

The optimal task error is obtained by using (3.3.67), so that

$$\mathbf{e}_k^* = \mathbf{J}_k \mathbf{x}_k^* - \mathbf{y}_k \quad (3.3.68)$$

$$= (\mathbf{V}_k \mathbf{F}_k + \mathbf{U}_k \mathbf{E}_k) \mathbf{Y}_{k-1}^T \mathbf{x}_k^* + \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{Y}_k^T \mathbf{x}_k^* - \mathbf{y}_k \quad (3.3.69)$$

$$\begin{aligned} &= (\mathbf{V}_k \mathbf{F}_k + \mathbf{U}_k \mathbf{E}_k - \mathbf{U}_k \mathbf{\Sigma}_k \hat{\mathbf{\Sigma}}_k^{-1} \mathbf{E}_k) \mathbf{Y}_{k-1}^T \mathbf{x}_{k-1}^* \\ &\quad + (\mathbf{U}_k \mathbf{\Sigma}_k \hat{\mathbf{\Sigma}}_k^{-1} \mathbf{U}_k^T - \mathbf{I}_k) \mathbf{y}_k, \end{aligned} \quad (3.3.70)$$

where the following relation is used in (3.3.69)

$$\mathbf{Y}_k^T \mathbf{x}_k^* = \hat{\mathbf{\Sigma}}_k^{-1} \mathbf{U}_k^T (\mathbf{y}_k - \mathbf{J}_k \mathbf{x}_{k-1}^*) \quad (3.3.71)$$

$$= \hat{\mathbf{\Sigma}}_k^{-1} \mathbf{U}_k^T (\mathbf{y}_k - \mathbf{J}_k \mathbf{Y}_{k-1} \mathbf{Y}_{k-1}^T \mathbf{x}_{k-1}^*), \quad (3.3.72)$$

$$= \hat{\mathbf{\Sigma}}_k^{-1} \mathbf{U}_k^T \mathbf{y}_k - \hat{\mathbf{\Sigma}}_k^{-1} \mathbf{E}_k \mathbf{Y}_{k-1}^T \mathbf{x}_{k-1}^*. \quad (3.3.73)$$

If we use the orthonormal property of $[\mathbf{U}_k \ \mathbf{V}_k]$, then the task error \mathbf{e}_k^* in (3.3.70) can be split into two parts associated with the left null space basis \mathbf{V}_k and the range space basis \mathbf{U}_k as follows:

$$\begin{aligned} \mathbf{e}_k^* &= (\mathbf{V}_k \mathbf{F}_k + \mathbf{U}_k \mathbf{E}_k - \mathbf{U}_k \mathbf{\Sigma}_k \hat{\mathbf{\Sigma}}_k^{-1} \mathbf{E}_k) \mathbf{Y}_{k-1}^T \mathbf{x}_{k-1}^* \\ &\quad + (\mathbf{U}_k \mathbf{\Sigma}_k \hat{\mathbf{\Sigma}}_k^{-1} \mathbf{U}_k^T - \mathbf{U}_k \mathbf{U}_k^T - \mathbf{V}_k \mathbf{V}_k^T) \mathbf{y}_k \end{aligned} \quad (3.3.74)$$

$$= \mathbf{V}_k (\mathbf{F}_k \mathbf{Y}_{k-1}^T \mathbf{x}_{k-1}^* - \mathbf{V}_k^T \mathbf{y}_k) + \mathbf{U}_k \mathbf{T}_k (\mathbf{E}_k \mathbf{Y}_{k-1}^T \mathbf{x}_{k-1}^* - \mathbf{U}_k^T \mathbf{y}_k), \quad (3.3.75)$$

where (3.3.45) is used.

Finally, the optimal task error can be rewritten

$$\mathbf{e}_k^* = \mathbf{v}_k^* + \mathbf{u}_k^*, \quad (3.3.76)$$

where

$$\mathbf{v}_k^* = \mathbf{V}_k(\mathbf{F}_k \mathbf{Y}_{k-1}^T \mathbf{x}_{k-1}^* - \mathbf{V}_k^T \mathbf{y}_k) \quad (3.3.77)$$

$$= \mathbf{V}_k \mathbf{V}_k^T (\mathbf{J}_k \mathbf{x}_{k-1}^* - \mathbf{y}_k) \quad (3.3.78)$$

$$\mathbf{u}_k^* = \mathbf{U}_k \mathbf{T}_k (\mathbf{E}_k \mathbf{Y}_{k-1}^T \mathbf{x}_{k-1}^* - \mathbf{U}_k^T \mathbf{y}_k) \quad (3.3.79)$$

$$= \mathbf{U}_k \mathbf{T}_k \mathbf{U}_k^T (\mathbf{J}_k \mathbf{x}_{k-1}^* - \mathbf{y}_k). \quad (3.3.80)$$

The optimal task error (3.3.76) has a strong intuitive appeal. The first term \mathbf{v}_k^* represents infeasible part due to the conflict with the priority tasks or a kinematic singularity itself while the second part \mathbf{u}_k^* denotes the task error induced by regularization to penalize the solution. In particular, if the k level is not in the kinematic or algorithmic singularities, then \mathbf{V}_k and \mathbf{v}_k^* are empty and the k level task is feasible. However, the regularization error \mathbf{u}_k^* does not vanish as long as $\rho_k \neq 0$. Therefore, the priority task given by \mathbf{x}_{k-1}^* interferes with the current k level and all following level tasks. This regularization error is the price we have to pay instead of guaranteeing the convergence and robustness of the solution near singular points. To address the regularization error the regularization coefficient adaptation is widely used [28, 61, 62, 63, 64, 65, 66]. Regularization coefficient ρ_k increases from zero when the solution moves toward singularity. Thus the regularization induced errors can be eliminated far from the singular points.

Based on the decomposed Jacobian matrix in (3.3.64), a stacked hierarchical

matrix decomposition (HMD) of the augmented Jacobian $\underline{\mathbf{J}}_k$ is obtained

$$\begin{aligned} \underline{\mathbf{J}}_k &= \begin{bmatrix} \underline{\mathbf{W}}_{k-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underline{\mathbf{U}}_k & \underline{\mathbf{V}}_k \end{bmatrix} \begin{bmatrix} \underline{\mathbf{S}}_{k-1} & \mathbf{0} & \mathbf{0} \\ \underline{\mathbf{E}}_k & \underline{\mathbf{\Sigma}}_k & \mathbf{0} \\ \underline{\mathbf{F}}_k & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{Y}}_{k-1}^T \\ \underline{\mathbf{Y}}_k^T \\ \underline{\mathbf{Z}}_k^T \end{bmatrix} \\ &= \underline{\mathbf{W}}_k \underline{\mathbf{S}}_k \underline{\mathbf{Y}}_k^T. \end{aligned} \quad (3.3.81)$$

where

$$\underline{\mathbf{W}}_k := \begin{bmatrix} \underline{\mathbf{W}}_{k-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underline{\mathbf{U}}_k & \underline{\mathbf{V}}_k \end{bmatrix} \quad (3.3.82)$$

$$\underline{\mathbf{S}}_k := \begin{bmatrix} \underline{\mathbf{S}}_{k-1} & \mathbf{0} \\ \underline{\mathbf{E}}_k & \underline{\mathbf{\Sigma}}_k \\ \underline{\mathbf{F}}_k & \mathbf{0} \end{bmatrix}. \quad (3.3.83)$$

According to the optimal solution (3.3.37), the optimum of the k level can be rewritten as follows:

$$\mathbf{x}_k^* := \underline{\mathbf{J}}_k^{\rho\sharp} \underline{\mathbf{y}}_k \quad (3.3.84)$$

$$= \begin{bmatrix} (\mathbf{I}_k - \underline{\mathbf{Y}}_k \hat{\underline{\Sigma}}_k^{-1} \underline{\mathbf{U}}_k^T \underline{\mathbf{J}}_k) \underline{\mathbf{J}}_{k-1}^{\rho\sharp} & \underline{\mathbf{Y}}_k \hat{\underline{\Sigma}}_k^{-1} \underline{\mathbf{U}}_k^T \end{bmatrix} \begin{bmatrix} \underline{\mathbf{y}}_{k-1} \\ \underline{\mathbf{y}}_k \end{bmatrix} \quad (3.3.85)$$

$$= \underline{\mathbf{Y}}_k \underline{\mathbf{S}}_k^{\rho\sharp} \underline{\mathbf{W}}_k^T \underline{\mathbf{y}}_k, \quad (3.3.86)$$

where $\mathbf{x}_{k-1}^* = \underline{\mathbf{J}}_{k-1}^{\rho\sharp} \underline{\mathbf{y}}_{k-1}$ is used and $\underline{\mathbf{S}}_k^{\rho\sharp}$ is defined as follows:

$$\underline{\mathbf{S}}_k^{\rho\sharp} := \begin{bmatrix} \underline{\mathbf{S}}_{k-1}^{\rho\sharp} & \mathbf{0} & \mathbf{0} \\ -\hat{\underline{\Sigma}}_k^{-1} \underline{\mathbf{E}}_k \underline{\mathbf{S}}_{k-1}^{\rho\sharp} & \hat{\underline{\Sigma}}_k^{-1} & \mathbf{0} \end{bmatrix}. \quad (3.3.87)$$

It is worth mentioning that the matrix $\underline{\mathbf{J}}_k^{\rho\sharp}$ in (3.3.84) is just defined to represent the optimum by convention. It respects none of the Moore-Penrose conditions (2.2.9)-(2.2.12). However, a generalized inverse $\underline{\mathbf{J}}_k^\sharp$ can be obtained by setting

$\rho_k = 0$ for all k as follows:

$$\underline{\mathbf{J}}_k^\# := \begin{bmatrix} (\mathbf{I}_k - \mathbf{Y}_k \underline{\boldsymbol{\Sigma}}_k^{-1} \mathbf{U}_k^T \mathbf{J}_k) \underline{\mathbf{J}}_{k-1}^\# & \mathbf{Y}_k \underline{\boldsymbol{\Sigma}}_k^{-1} \mathbf{U}_k^T \end{bmatrix} \quad (3.3.88)$$

$$= \underline{\mathbf{Y}}_k \underline{\mathbf{S}}_k^\# \underline{\mathbf{W}}_k^T \quad (3.3.89)$$

where $\underline{\mathbf{S}}_k^\#$ can be acquired from (3.3.87) with $\rho_k = 0$ as follows:

$$\underline{\mathbf{S}}_k^\# = \begin{bmatrix} \underline{\mathbf{S}}_{k-1}^\# & \mathbf{0} & \mathbf{0} \\ -\underline{\boldsymbol{\Sigma}}_k^{-1} \mathbf{E}_k \underline{\mathbf{S}}_{k-1}^\# & \underline{\boldsymbol{\Sigma}}_k^{-1} & \mathbf{0} \end{bmatrix}. \quad (3.3.90)$$

As $\underline{\mathbf{S}}_k^\# \underline{\mathbf{S}}_k$ is an identity matrix, the following relation holds

$$\underline{\mathbf{J}}_k^\# \mathbf{J}_k = \underline{\mathbf{Y}}_k \underline{\mathbf{Y}}_k^T. \quad (3.3.91)$$

Three of the four Moore-Penrose conditions are satisfied by $\underline{\mathbf{J}}_k^\#$ except that $\underline{\mathbf{J}}_k \underline{\mathbf{J}}_k^\#$ is symmetric in (2.2.11), and thus $\underline{\mathbf{J}}_k^\#$ is a reflexive generalized inverse of $\underline{\mathbf{J}}_k$. The relation (3.3.91) will be used to find the Lagrange multiplier.

Finally, we can check that the following conditions in (3.3.26) are also satisfied if the regularization matrix (3.3.35) is used

$$\underline{\mathbf{H}}_k \mathbf{Z}_k = (\underline{\mathbf{Y}}_k \underline{\mathbf{S}}_k^2 \underline{\mathbf{Y}}_k^T + \rho_k^2 \underline{\mathbf{Y}}_k \underline{\mathbf{Y}}_k^T) \mathbf{Z}_k = \mathbf{0} \quad (3.3.92)$$

$$\underline{\mathbf{h}}_k^T \mathbf{Z}_k = \underline{\mathbf{y}}_k^T \underline{\mathbf{W}}_k \underline{\mathbf{S}}_k \underline{\mathbf{Y}}_k^T \mathbf{Z}_k = \mathbf{0}, \quad (3.3.93)$$

where $\underline{\mathbf{Y}}_k^T \mathbf{Z}_k = \mathbf{0}$ is used.

3.3.4 Illustrative Examples for Regularized Hierarchical Optimization

To verify the effectiveness of RHQP, a simple mobile manipulator example is borrowed from [16]. Consider again the mobile robot as shown in Figure 3.1. We use the integration interval $\Delta t = 0.1$ s, task function gains $K_1 = K_2 = 1$ in (3.2.6), and the regularization coefficient $\rho_1 = \rho_2 = 0.5$. This problem can be interpreted

as the first-order closed-loop inverse kinematics to generate joint space motions [94, 95]. The common purpose of two task functions is to drive their values to the origin $[0, 0]^T$ as close as possible.

Let us consider two cases according to whether the target is reachable or not. At first, consider the target point ${}^d\mathbf{s} = [0.5, 1.0]^T$. This point is within the reachable space, and there is no task confliction. We compare the solutions of unregularized HQP and RHQP in terms of the magnitude of the task function, distance to the optimum, and robot configurations.

As depicted in Figure 3.3, all task function values converge to zeros regardless of regularization. Note that the unregularized solution starting near singular point (case 5) requires high velocity to abruptly change the initial wrong direction of task function as presented in Figure 3.3 (top left corner). The convergence speed of the unregularized solution is faster than that of the regularized one. The robots with the unregularized solution have already reached the target point at 3 s, but the robots with regularization reach the target at 5 s as you can see in Figure 3.3. Because RHQP penalizes the velocity solution for singularity robustness.

Next, the desired point is assigned as ${}^d\mathbf{s} = [2.0, 0.0]^T$. From the task function (3.2.8), the two task functions cannot be zeros at the same time. This unreachability means that these two functions cannot be realized concurrently and the two tasks are conflicting. The position of mobile base $q_1 = 1$ and joint angle $q_2 = 2j\pi$ ($j = 0, 1, 2, \dots$) make the first level task function zero, but the second task cannot be perfectly executed. The task Jacobian of (3.2.9) becomes linearly dependent, and the two objectives tend to exploit the same joints so that there is no resource to satisfy multiple objectives simultaneously. In other words, the solution lies on algorithmic singularity.

However, the second task is achieved as much as possible by making the joint

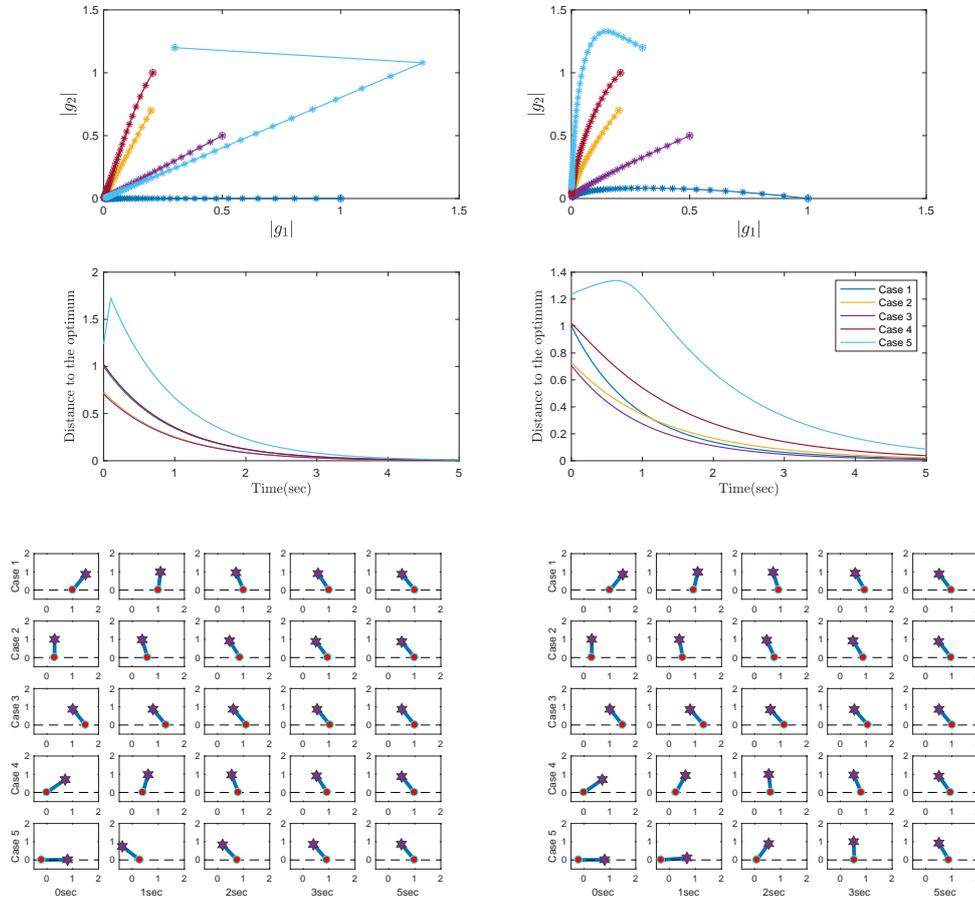


Figure 3.3: Numerical simulation results computed via unregularized HQP (left column) and RHQP (right column). Two tasks can be realized at the same time. We compare the results in terms of time history of the magnitude of the task function (top row), distance to the optima (middle row), and time history of the robot poses according to various initial conditions (bottom row). The convergence speed of the unregularized solution is faster than that of the regularized one. However, the unregularized solution starting near singular point (case 5) requires high velocity.

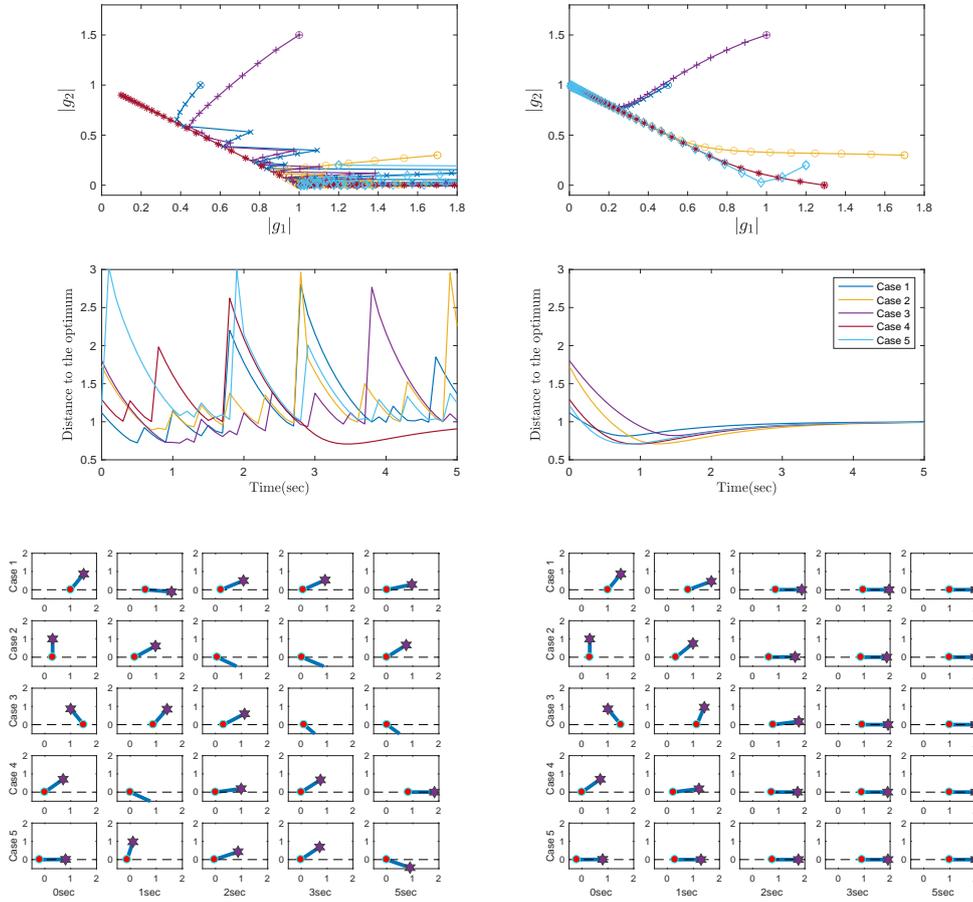


Figure 3.4: Numerical simulation results computed via unregularized HQP (left column) and RHQP (right column). Two tasks cannot be satisfied simultaneously: there is a conflict between two tasks. We compare the results in terms of time history of the magnitude of the task function (top row), distance to the optima (middle row), and time history of the robot poses according to various initial conditions (bottom row). As it can be seen, the solution sequences of unregularized HQP does not converge to the optima while all solutions of RHQP converge to the optima respecting hierarchy.

angle q_2 zero. If q_2 has a nonzero value, then the second level objective has a larger task error. As can be seen from Figure 3.4, the solution sequences of unregularized HQP does not converge to the optima

$$\left[|g_1(\mathbf{q}^*)| \quad |g_2(\mathbf{q}^*)| \right]^T = \begin{bmatrix} 0 & 1 \end{bmatrix}^T \quad (3.3.94)$$

except for the Case 4 which happen to be on the reachable boundary. However, all RHQP solutions converge to the optima while respecting hierarchy.

To summarize, every task is achieved as much as possible according to the levels in the task priority structure and the regularization can guarantee the convergence of the solution even if the solution is near singular points. Reference [16] resorts to a trust region method, where the convergence can be obtained only when the trust bound is carefully adjusted based on heuristics. In contrast to the trust region scheme, we can obtain the robustly converging sequences in the face of singularities with a constant regularization coefficients under the strict hierarchical structure.

3.4 Regularized Hierarchical Optimization with Inequality Constraints

Now we turn our attention to the inequality tasks. We are going to fully integrate task regularization and the inequality-constrained hierarchical optimization to guarantee the robustness and convergence of the solution in the face of singularity. All Lagrange multipliers necessary to handle inequality constraints are analytically obtained by leveraging the robust optimal solution, task error, and the hierarchical matrix decomposition developed in the previous sections. Then we propose the modified active set algorithm which is simple and fast because it does

not use a step direction nor a step length usually used in numerical optimization. The sign of Lagrange multipliers is exploited to handle inequality constraints.

3.4.1 Lagrange Multipliers

We rewrite the optimization problem (3.3.11) as the regularized hierarchical quadratic program (RHQP) composed of a set of p stacked of all individual linear inequality constraints as follows:

$$\min_{\mathbf{x}_k, \mathbf{e}_k, \mathbf{w}_k} \quad \frac{1}{2} \|\mathbf{e}_k\|_2^2 + \frac{1}{2} \|\mathbf{w}_k\|_2^2 \quad (3.4.95a)$$

$$\text{subject to} \quad \begin{bmatrix} \mathbf{J}_k \\ \rho_k \mathbf{Y}_k^T \end{bmatrix} \mathbf{x}_k - \begin{bmatrix} \mathbf{y}_k \\ \mathbf{0} \end{bmatrix} \leq \begin{bmatrix} \mathbf{e}_k \\ \mathbf{w}_k \end{bmatrix} \quad (3.4.95b)$$

$$\begin{bmatrix} \underline{\mathbf{J}}_{k-1} \\ \underline{\rho} \mathbf{Y}_{k-1}^T \end{bmatrix} \mathbf{x}_k - \begin{bmatrix} \underline{\mathbf{y}}_{k-1} \\ \mathbf{0} \end{bmatrix} \leq \begin{bmatrix} \underline{\mathbf{e}}_{k-1}^* \\ \underline{\mathbf{w}}_{k-1}^* \end{bmatrix} \quad (3.4.95c)$$

where $\underline{\mathbf{J}}_{k-1}$, $\underline{\rho} \mathbf{Y}_{k-1}$, $\underline{\mathbf{y}}_{k-1}$, $\underline{\mathbf{w}}_{k-1}^*$ and $\underline{\mathbf{e}}_{k-1}^*$ are the stacked matrices and the vectors augmented from level 1 to $k-1$, i.e.,

$$\underline{\mathbf{J}}_{k-1} = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_{k-1} \end{bmatrix}, \quad (3.4.96)$$

$$\underline{\rho} \mathbf{Y}_{k-1} = \begin{bmatrix} \rho_1 \mathbf{Y}_1 & \cdots & \rho_{k-1} \mathbf{Y}_{k-1} \end{bmatrix}. \quad (3.4.97)$$

The fixed stacked error $\underline{\mathbf{e}}_{k-1}^*$ and the stacked regularized solution $\underline{\mathbf{w}}_{k-1}^*$ are accrued from the previous level QP. The current level solution \mathbf{x}_k^* should respect the constraint (3.4.95c), which is referred to the strict constraint. When tasks are incompatible, the solution with the strict constraints inevitably restricts the fulfillment of the tasks (3.4.95b), which can be best optimized in the least-square sense

by increasing \mathbf{e}_k and \mathbf{w}_k to the minimum magnitude. Therefore, the task (3.4.95b) is also denoted by the relaxed constraint.

As discussed in the previous Section 3.3.3, if the regularization matrix $\mathbf{D}_k = \mathbf{Y}_k^T$ in (3.3.35) and the solution \mathbf{x}_k^* in (3.3.37) are used, then the strict constraints $\rho \mathbf{Y}_{k-1}^T \mathbf{x}_k = \underline{\mathbf{w}}_{k-1}^*$ of (3.4.95c) is automatically satisfied due to the property of $\mathbf{Y}_{k-1}^T \mathbf{x}_k^* = \mathbf{Y}_{k-1}^T \mathbf{x}_{k-1}^*$. Therefore, the k level optimization can be conducted irrespective of this constraint and RHQP of (3.4.95) can be rewritten as

$$\min_{\mathbf{x}_k, \mathbf{e}_k} \quad \frac{1}{2} \|\mathbf{e}_k\|_2^2 + \frac{1}{2} \rho_k^2 \|\mathbf{Y}_k^T \mathbf{x}_k\|_2^2 \quad (3.4.98a)$$

$$\text{subject to} \quad \mathbf{J}_k \mathbf{x}_k - \mathbf{y}_k \leq \mathbf{e}_k \quad (3.4.98b)$$

$$\underline{\mathbf{J}}_{k-1} \mathbf{x}_k - \underline{\mathbf{y}}_{k-1} \leq \underline{\mathbf{e}}_{k-1}^* \quad (3.4.98c)$$

The role of slack variable \mathbf{e}_k is to relax the constraint (3.4.98b) when the task is infeasible.

The Lagrangian function associated with the constrained problem is introduced as follows:

$$\begin{aligned} \mathcal{L}_k := & \frac{1}{2} \mathbf{e}_k^T \mathbf{e}_k + \frac{1}{2} \rho_k^2 \mathbf{x}_k^T \mathbf{Y}_k \mathbf{Y}_k^T \mathbf{x}_k + \boldsymbol{\mu}_k^T (\mathbf{J}_k \mathbf{x}_k - \mathbf{y}_k - \mathbf{e}_k) \\ & + \underline{\boldsymbol{\mu}}_k^T (\underline{\mathbf{J}}_{k-1} \mathbf{x}_k - \underline{\mathbf{y}}_{k-1} - \underline{\mathbf{e}}_{k-1}^*) \end{aligned} \quad (3.4.99)$$

where $\boldsymbol{\mu}_k$ and $\underline{\boldsymbol{\mu}}_k$ are the Lagrange multipliers corresponding to the relaxed constraint (3.4.98b) and the strict constraint (3.4.98c), respectively. Suppose that \mathbf{x}_k^* and \mathbf{e}_k^* is a local solution of (3.4.98), and then there are Lagrange multiplier vectors $\boldsymbol{\mu}_k^*$ and $\underline{\boldsymbol{\mu}}_k^*$. Differentiating the Lagrangian over the dual variables ($\boldsymbol{\mu}_k$ and $\underline{\boldsymbol{\mu}}_k$) and the primal variables (\mathbf{e}_k and \mathbf{x}_k) gives the Karush-Kuhn-Tucker or KKT

conditions satisfied at $(\mathbf{x}_k^*, \mathbf{e}_k^*, \boldsymbol{\mu}_k^*, \underline{\boldsymbol{\mu}}_k^*)$:

$$\mathbf{J}_k \mathbf{x}_k^* - \mathbf{y}_k - \mathbf{e}_k^* \leq \mathbf{0} \quad (3.4.100)$$

$$\underline{\mathbf{J}}_{k-1} \mathbf{x}_k^* - \underline{\mathbf{y}}_{k-1} - \underline{\mathbf{e}}_{k-1}^* \leq \mathbf{0} \quad (3.4.101)$$

$$\boldsymbol{\mu}_k^* - \mathbf{e}_k^* = \mathbf{0} \quad (3.4.102)$$

$$\underline{\mathbf{J}}_{k-1}^T \underline{\boldsymbol{\mu}}_k^* + \mathbf{J}_k^T \boldsymbol{\mu}_k^* + \rho_k^2 \mathbf{Y}_k \mathbf{Y}_k^T \mathbf{x}_k^* = \mathbf{0} \quad (3.4.103)$$

$$\boldsymbol{\mu}_k^* \geq \mathbf{0} \quad (3.4.104)$$

$$\underline{\boldsymbol{\mu}}_k^* \geq \mathbf{0} \quad (3.4.105)$$

$$\boldsymbol{\mu}_k^{*T} (\mathbf{J}_k \mathbf{x}_k^* - \mathbf{y}_k - \mathbf{e}_k^*) = 0 \quad (3.4.106)$$

$$\underline{\boldsymbol{\mu}}_k^{*T} (\underline{\mathbf{J}}_{k-1} \mathbf{x}_k^* - \underline{\mathbf{y}}_{k-1} - \underline{\mathbf{e}}_{k-1}^*) = 0 \quad (3.4.107)$$

The equations (3.4.106) and (3.4.107) are complementarity conditions, which mean either constraint is active or $\boldsymbol{\mu}_k^* = \mathbf{0}$, $\underline{\boldsymbol{\mu}}_k^* = \mathbf{0}$, or possibly both.

As we know the optimal task error \mathbf{e}_k^* from (3.3.76), the dual optimum $\boldsymbol{\mu}_k^*$ is found using (3.4.102)

$$\boldsymbol{\mu}_k^* = \mathbf{v}_k^* + \mathbf{u}_k^*. \quad (3.4.108)$$

The Lagrange multiplier $\underline{\boldsymbol{\mu}}_k^*$ is also obtained from (3.4.103) by using (3.3.91) as follows:

$$\underline{\boldsymbol{\mu}}_k^* = -\underline{\mathbf{J}}_{k-1}^{*T} \mathbf{J}_k^T (\mathbf{v}_k^* + \mathbf{u}_k^*). \quad (3.4.109)$$

As in the case of $\boldsymbol{\mu}_k^*$, the multiplier $\underline{\boldsymbol{\mu}}_k^*$ in (3.4.109) has also been separated into the infeasible task and the regularization error associated with \mathbf{v}_k^* and \mathbf{u}_k^* , respectively.

Let us verify that the solution (3.4.109) satisfies the optimality condition (3.4.103). By using $\boldsymbol{\mu}_k^*$ in (3.4.108), the optimality condition (3.4.103) is rewritten in the form

$$\underline{\mathbf{J}}_{k-1}^T \underline{\boldsymbol{\mu}}_k^* = -\mathbf{J}_k^T (\mathbf{v}_k^* + \mathbf{u}_k^*) - \rho_k^2 \mathbf{Y}_k \mathbf{Y}_k^T \mathbf{x}_k^*. \quad (3.4.110)$$

Premultiplying both sides of the solution (3.4.109) by matrix $\underline{\mathbf{J}}_{k-1}^T$ and using (3.3.91) we get

$$\underline{\mathbf{J}}_{k-1}^T \underline{\boldsymbol{\mu}}_k^* = -\underline{\mathbf{Y}}_{k-1} \underline{\mathbf{Y}}_{k-1}^T \underline{\mathbf{J}}_k^T (\mathbf{v}_k^* + \mathbf{u}_k^*). \quad (3.4.111)$$

Now we show that each right side of (3.4.110) and (3.4.111) is equal. Referring to \mathbf{v}_k^* in (3.3.78), the first term of (3.4.111) is written as

$$\begin{aligned} \underline{\mathbf{Y}}_{k-1} \underline{\mathbf{Y}}_{k-1}^T \underline{\mathbf{J}}_k^T \mathbf{v}_k^* &= \underline{\mathbf{Y}}_{k-1} (\underline{\mathbf{Y}}_{k-1}^T \underline{\mathbf{J}}_k^T \mathbf{V}_k) \mathbf{V}_k^T (\mathbf{J}_k \mathbf{x}_{k-1}^* - \mathbf{y}_k) \\ &= \underline{\mathbf{Y}}_{k-1} \mathbf{F}_k^T \mathbf{V}_k^T (\mathbf{J}_k \mathbf{x}_{k-1}^* - \mathbf{y}_k) \\ &= \left(\underbrace{\underline{\mathbf{Y}}_{k-1} (\mathbf{E}_k^T \mathbf{U}_k^T + \mathbf{F}_k^T \mathbf{V}_k^T) + \mathbf{Y}_k \boldsymbol{\Sigma}_k \mathbf{U}_k^T}_{\underline{\mathbf{J}}_k^T} \right) \\ &\quad \times \mathbf{V}_k \mathbf{V}_k^T (\mathbf{J}_k \mathbf{x}_{k-1}^* - \mathbf{y}_k) \\ &= \underline{\mathbf{J}}_k^T \mathbf{v}_k^* \end{aligned} \quad (3.4.112)$$

where the orthonormal property (3.3.2) and the hierarchically decomposed Jacobian of (3.3.67) are used.

Similarly, by recalling \mathbf{u}_k^* in (3.3.80), the second term of right hand side in (3.4.111) associated with the regularization error is expressed as

$$\begin{aligned} \underline{\mathbf{Y}}_{k-1} \underline{\mathbf{Y}}_{k-1}^T \underline{\mathbf{J}}_k^T \mathbf{u}_k^* &= \underline{\mathbf{Y}}_{k-1} (\underline{\mathbf{Y}}_{k-1}^T \underline{\mathbf{J}}_k^T \mathbf{U}_k) \mathbf{T}_k \mathbf{U}_k^T (\mathbf{J}_k \mathbf{x}_{k-1}^* - \mathbf{y}_k) \\ &= \underline{\mathbf{Y}}_{k-1} \mathbf{E}_k^T \mathbf{T}_k \mathbf{U}_k^T (\mathbf{J}_k \mathbf{x}_{k-1}^* - \mathbf{y}_k) \\ &= \left(\underbrace{\underline{\mathbf{Y}}_{k-1} (\mathbf{E}_k^T \mathbf{U}_k^T + \mathbf{F}_k^T \mathbf{V}_k^T) + \mathbf{Y}_k \boldsymbol{\Sigma}_k \mathbf{U}_k^T}_{\underline{\mathbf{J}}_k^T} \right) \\ &\quad \times \mathbf{U}_k \mathbf{T}_k \mathbf{U}_k^T (\mathbf{J}_k \mathbf{x}_{k-1}^* - \mathbf{y}_k) - \underbrace{\mathbf{Y}_k \boldsymbol{\Sigma}_k \mathbf{T}_k \mathbf{U}_k^T (\mathbf{J}_k \mathbf{x}_{k-1}^* - \mathbf{y}_k)}_{\rho_k^2 \hat{\mathbf{x}}_k^*} \\ &= \underline{\mathbf{J}}_k^T \mathbf{u}_k^* + \rho_k^2 \mathbf{Y}_k \mathbf{Y}_k^T \mathbf{x}_k^* \end{aligned} \quad (3.4.113)$$

where the orthonormal property of (3.3.2), relation $\Sigma_k \mathbf{T}_k = \rho_k^2 \hat{\Sigma}_k^{-1}$, the solution (3.3.40), and the hierarchically decomposed Jacobian matrix in (3.3.67) are utilized. Thus we have verified that the solution (3.4.109) satisfies the optimality condition (3.4.110).

By recalling $\mathbf{J}_k^\#$ in (3.3.88) and $\underline{\boldsymbol{\mu}}_k^*$ in (3.4.108), Lagrange multiplier $\underline{\boldsymbol{\mu}}_k^*$ of (3.4.109) is written as:

$$\underline{\boldsymbol{\mu}}_k^* = \begin{bmatrix} \underline{\boldsymbol{\mu}}_{1,k}^* \\ \vdots \\ \underline{\boldsymbol{\mu}}_{k-1,k}^* \end{bmatrix} = \begin{bmatrix} -\mathbf{J}_{k-2}^{\#T} (\mathbf{J}_k^T \underline{\boldsymbol{\mu}}_k^* + \mathbf{J}_{k-1}^T \underline{\boldsymbol{\mu}}_{k-1,k}^*) \\ -\mathbf{U}_{k-1} \Sigma_{k-1}^{-1} \mathbf{Y}_{k-1}^T \mathbf{J}_k^T \underline{\boldsymbol{\mu}}_k^* \end{bmatrix}. \quad (3.4.114)$$

We thus can recursively obtain the components $\underline{\boldsymbol{\mu}}_{j,k}^*$ of the Lagrange multipliers $\underline{\boldsymbol{\mu}}_k^*$ from $j = k - 1$ to 1

$$\underline{\boldsymbol{\mu}}_{j,k}^* = -\mathbf{U}_j \Sigma_j^{-1} \mathbf{Y}_j^T \left(\sum_{i=j+1}^{k-1} \mathbf{J}_i^T \underline{\boldsymbol{\mu}}_{i,k}^* + \mathbf{J}_k^T \underline{\boldsymbol{\mu}}_k^* \right). \quad (3.4.115)$$

These two Lagrange multipliers can be expressed in the matrix form [44]:

$$\mathbf{M}_p^* = \begin{bmatrix} \underline{\boldsymbol{\mu}}_1^* & \underline{\boldsymbol{\mu}}_2^* & \cdots & \underline{\boldsymbol{\mu}}_p^* \\ \underline{\boldsymbol{\mu}}_2^* & \underline{\boldsymbol{\mu}}_3^* & \cdots & \underline{\boldsymbol{\mu}}_p^* \\ \vdots & \vdots & \ddots & \vdots \\ \underline{\boldsymbol{\mu}}_{p-1}^* & \underline{\boldsymbol{\mu}}_p^* & \cdots & \underline{\boldsymbol{\mu}}_p^* \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{\mu}}_1^* & \underline{\boldsymbol{\mu}}_{1,2}^* & \cdots & \underline{\boldsymbol{\mu}}_{1,p-1}^* & \underline{\boldsymbol{\mu}}_{1,p}^* \\ & \underline{\boldsymbol{\mu}}_2^* & \cdots & \underline{\boldsymbol{\mu}}_{2,p-1}^* & \underline{\boldsymbol{\mu}}_{2,p}^* \\ & & & \vdots & \vdots \\ & & & \underline{\boldsymbol{\mu}}_{p-1}^* & \underline{\boldsymbol{\mu}}_{p-1,p}^* \\ & & & & \underline{\boldsymbol{\mu}}_p^* \end{bmatrix}. \quad (3.4.116)$$

From (3.4.108) and (3.4.114), each column of the Lagrangian multiplier matrix \mathbf{M}_p^* is calculated sequentially from level $k = 1$ to p . Moreover, all necessary components of Lagrange multiplier $\underline{\boldsymbol{\mu}}_k$ are obtained in recursive procedure.

Indeed, one of the most attractive features of the RHQP solver is that the Lagrange multiplier matrix \mathbf{M}_p^* of (3.4.116) allows us to control inequality constraints very efficiently by considering all level's hierarchical tasks together with

only one active set. The sign of the components of two Lagrange multipliers in (3.4.104) and (3.4.105) of \mathbf{M}_p^* play a significant role to treat inequality constraints. The multiplier matrix eliminates the iterative process used to find each level optimal active set.

If the active set is optimal for all levels, then all constraints are satisfied by the optimal solution and all $\underline{\mu}_k^*$ and μ_k^* of the active constraints can be nonnegative. Otherwise, if some of the components of the multipliers of \mathbf{M}_k^* are strictly negative, the constraint having the lowest component can be deactivated according to the priority. However, the cascade-QP should perform this iterative process at each level of QP.

In the next section, hierarchy is considered for treating inequality constraints with the sign convention. As we have found a way to compute the Lagrange multipliers to control inequality constraints, we will exploit them in the modified active-set algorithm. The Lagrange multipliers of unregularized HQP of [44] can be obtained from the RHQP solution μ_k^* in (3.4.108) and $\underline{\mu}_k^*$ in (3.4.109) by setting each level regularization coefficient to zero. Therefore, we generalize the HQP framework by integrating regularization.

3.4.2 Modified Active Set Method

The basic underlying idea of the active set method for solving an inequality constrained QP is to classify inequality constraints into active and inactive constraints. The active constraints are treated as equality ones while the inactive constraints are essentially ignored. However, we usually do not know the optimal active set, an iterative process is involved to find a set of equality constraints, termed a *working set*. The working set \mathcal{W} is expressed by the finite set of indices (k, r) or $cst(k, r)$ which denotes the r^{th} constraint of the level k .

Algorithm 1 Regularized hierarchical quadratic programming

```

1: Input: Initial guess  $\mathcal{W}^{(0)}$ 
2: Output:  $\hat{\mathbf{q}}^*(t_{k+1})$ ,  $\mathbf{q}^*(t_{k+1})$ 
3: Initialize:  $\mathcal{W} = \mathcal{W}^{(0)}$ ,  $\mathcal{S} = \emptyset$ 
4: repeat
5:   // Hierarchical matrix decomposition
6:   HMD of  $\mathcal{W} = \text{HMD}(\mathcal{W})$  Eq. (3.3.81)
7:    $[\mathbf{x}^*, \mathbf{e}^*] = \text{RHQP\_primal}(\text{HMD of } \mathcal{W})$  Eq. (3.3.86) and (3.3.76)
8:    $\mathbf{M}_p^* = \text{RHQP\_dual}(\mathbf{x}^*, \mathbf{e}^*, \text{HMD of } \mathcal{W})$  Eq. (3.4.116)
9:    $[\beta, \text{cst}_\beta] = \min \mathbf{M}_p^*$ 
10:   $[\alpha, \text{cst}_\alpha] = \text{blocking\_cst}(\mathbf{x}^*, \mathcal{W}^c)$  Eq. (3.4.117)
11:  // Selection of strongly active constraints
12:   $\mathcal{S} \leftarrow (\mathcal{S} \cup \{\text{cst} \mid \underline{\boldsymbol{\mu}}_{\text{cst}}^* > 0\} \cup \{\text{cst} \mid \underline{\boldsymbol{\mu}}_{\text{cst}}^* > 0\}) \cap \mathcal{W}$ 
13:  if  $\alpha < 1$  then
14:    // Add a blocking constraint
15:     $\mathcal{W} \leftarrow \mathcal{W} \cup \{\text{cst}_\alpha\}$ 
16:  else if  $(\beta < 0)$  and  $(\text{cst}_\beta \notin \mathcal{S})$  then
17:    // Remove an inactive constraint
18:     $\mathcal{W} \leftarrow \mathcal{W} \setminus \{\text{cst}_\beta\}$ 
19:  else
20:    break
21:  end if
22: until  $\alpha = 1$  and  $\beta > 0$ 
23:  $\hat{\mathbf{q}}^*(t_{k+1}) \leftarrow \mathbf{x}^*$ ,  $\mathbf{q}^*(t_{k+1}) \leftarrow \mathbf{q}(t_k) + \int_0^{\Delta t} \mathbf{x}^* d\zeta$ 
24: // Time update for all parameters
25:  $\underline{\mathbf{J}} \leftarrow \underline{\mathbf{J}}(\mathbf{q}^*(t_{k+1}))$ ,  $\underline{\mathbf{y}} \leftarrow \underline{\mathbf{y}}(\mathbf{q}^*(t_{k+1}))$ 
26: return  $\hat{\mathbf{q}}^*(t_{k+1})$ ,  $\mathbf{q}^*(t_{k+1})$ 

```

The method starts with an initial guess which subset of the inequality constraints hold as equalities at the optimum whose subset is called the active set. At each iteration, it solves the equality only QP subject to only the active constraints as equalities. Once the solution is obtained from this sub-QP, we apply modification to the current active set. A blocking constraint violated by the solution is added to the working set and an unnecessary constraint is removed. The algorithm then iterates until the optimal active set is found, for which the associated equality QP provides the optimum of the inequality QP [55, 56]. The RHQP solver is working with one active set because the Lagrange multiplier matrix \mathbf{M}_p^* contains all information to handle inequality constraints for all level's hierarchy.

The proposed algorithm is summarized in Algorithm 1. At each time step t_k , the RHQP solver operates as follows. First, it begins by making an initial guess of the optimal active set. We use a warm start for the selection of $\mathcal{W}^{(0)}$. In robotic problems, the smoothness and continuity of the robot state are usually guaranteed at least locally, and therefore, it is likely that the optimal active set of the time step t_k can be an initial working set at the next time step t_{k+1} .

Second, the algorithm performs the hierarchical matrix decomposition (3.3.81) with the working set $\mathcal{W}^{(i)}$ at the iteration i . Then the candidate primal optimum $\mathbf{x}_p^{*(i)}$ and each level task error $\mathbf{e}_k^{*(i)}$ are computed through the function of the RHQP_primal. The function RHQP_dual allows us to obtain the Lagrange multiplier matrix $\mathbf{M}_p^{*(i)}$ in (3.4.116).

Third, once the candidate solution and the Lagrange multipliers are obtained, the optimal active set should be found. The first part of this constraint control algorithm activates a blocking constraint from $(k, r) \notin \mathcal{W}^{(i)}$ violated by $\mathbf{x}^{*(i)}$, while the second part removes an unnecessary constraint. If $\mathbf{x}^{*(i)}$ is feasible with respect to all the constraints in the complementary set of $\mathcal{W}^{(i)}$, then there is no blocking

constraint.

Conversely, if $\mathbf{x}^{*(i)}$ is not feasible, we find a new working set $\mathcal{W}^{(i+1)}$ by appending the blocking constraint to $\mathcal{W}^{(i)}$. We select the blocking constraint maximally violated by $\mathbf{x}^{*(i)}$ with

$$\alpha^{(i)} = \min \left\{ 1, \min_{k,r} \{ \alpha_{k,r}^{(i)} \} \right\} \quad (3.4.117)$$

where

$$\alpha_{k,r}^{(i)} = \begin{cases} \left| \frac{\mathbf{y}_{k,r}}{\mathbf{J}_{k,r} \mathbf{x}^{*(i)}} \right| & \text{if } \mathbf{J}_{k,r} \mathbf{x}^{*(i)} > \mathbf{y}_{k,r}, \quad (k,r) \notin \mathcal{W}^{(i)} \\ 1 & \text{otherwise} \end{cases}$$

If $\alpha^{(i)} = 1$, then no new constraint is activated with $\mathbf{x}^{*(i)}$, and thus there is no blocking constraint on this iteration. Otherwise, if $\alpha^{(i)} < 1$, the constraint (k,r) with the smallest $\alpha_{k,r}^{(i)}$ is appended to a new working set $\mathcal{W}^{(i+1)}$. Then, a new iteration begins by performing the hierarchical matrix decomposition with $\mathcal{W}^{(i+1)}$. As we do not use a step direction nor a step length usually used in numerical optimization [55], our algorithm is simple. Moreover, it operates regardless of whether the initial point $\mathbf{x}^{*(0)}$ is feasible or not for an initial working set.

Now it is time to remove an unnecessary constraint in the working set if there is no blocking constraint. Toward this end, the sign of the components of two Lagrange multipliers in (3.4.104) and (3.4.105) play a significant role. If the active set is optimal, then all constraints are respected at the optimal point $\mathbf{x}^{*(i)}$, and all components of $\underline{\boldsymbol{\mu}}_k^*$ and $\boldsymbol{\mu}_k^*$ are nonnegative. Otherwise, if some of the components of the multipliers are strictly negative, the constraint having the lowest component can be deactivated.

Here, care must be taken when deactivating the constraints due to the hierarchy. Strongly active constraints with the strictly positive Lagrange multipliers

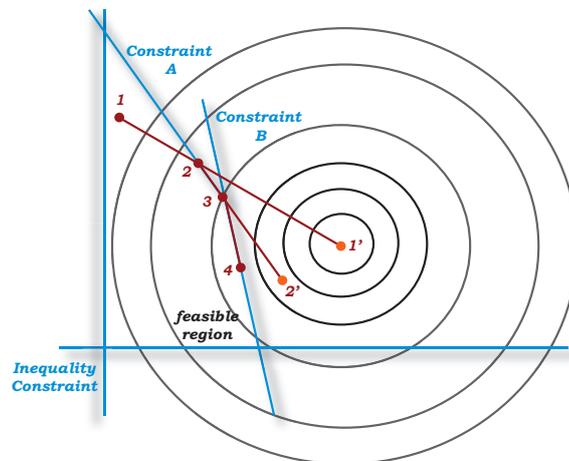
are stored in the set $\mathcal{S}^{(i)}$ to prevent a lower level task from deactivating a strong active constraint with higher priority [44]. Thus, if a constraint has the smallest negative component in the Lagrange multipliers and at the same time it does not have a strictly positive component of the multipliers of all previous levels, then it should be dropped. If a constraint needs to be activated or deactivated, it implies that the current working set is not optimal.

Once the optimal active set and the solution are found, and then all parameters and constraints are updated based on the primal optimum for next time step optimization. The RHQP solver can find the robust optimal solution efficiently because it uses a single active set composed of all possible combinations of active constraints for all levels.

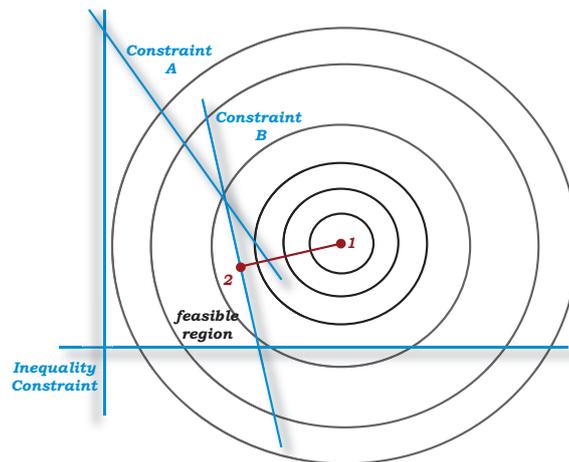
3.4.3 Illustrative Examples of Modified Active Set Method

To compare the active set method and modified one used in the numerical optimization and RHQP respectively, an illustrative example is shown in Figure 3.5. As the active constraints restricts the domain of feasibility in neighborhoods of feasible point, they are treated as equality constraints. While inactive constraints have no influence in the vicinity of feasible point, they are essentially ignored. It should be clear that if the optimal active set were known, then the optimization problem could be solved with equality constraints only.

For a general active set method, once a working set $\mathcal{W}^{(i)}$ at the current iteration i is obtained, then we have to choose a direction and step size to move along its direction to obtain maximum benefit without violating all constraints. A new working set $\mathcal{W}^{(i+1)}$ is constructed by adding the blocking constraints to $\mathcal{W}^{(i)}$. On the other hand, we remove an index (k, r) corresponding to the smallest negative Lagrange multipliers of $\mu_{k,r}$ and $\underline{\mu}_{k,r} \in \mathcal{W}^{(i)}$. Then a new working set $\mathcal{W}^{(i+1)}$ is



(a) Active set method



(b) Modified active set method

Figure 3.5: Comparison of the active set method and modified active set method.

(a) Active set method: 1: starting point ($1'$: unconstrained optimum) \rightarrow 2: constraint A is activated ($2'$: optimum on constraint A) \rightarrow 3: constraint A is deactivated and B is activated \rightarrow 4: optimum (b) Modified active set method: 1: unconstrained optimum \rightarrow 2: optimum (constraint B is activated).

established. This process is repeated until the optimal active set is obtained and it is depicted in Figure 3.5 (a).

The modified active set method does not determine a step length nor step direction as shown in Figure 3.5 (b). Instead, it focuses only on finding the blocking constraint $(k, r) \notin \mathcal{W}^{(i)}$ that is maximally violated by the current optimal point $\mathbf{x}^{*(i)}$ and removing an unnecessary constraint. If there is a blocking constraint, a new working set $\mathcal{W}^{(i+1)}$ is constructed by appending the blocking constraint to $\mathcal{W}^{(i)}$. Then optimal solution can be found directly from (3.3.37) or (3.3.86) with the new blocking constraint. The process to remove an unnecessary constraint is similar to the active set method except hierarchy. As the warm start for the selection of the initial working set $\mathcal{W}^{(0)}$ is used, the optimum is found at once if there is no change in $\mathcal{W}^{(0)}$ between t_{k-1} and t_k .

As a step direction and a step length are not required, the modified active set algorithm is simple and fast. Moreover, it operates regardless of whether the initial point $\mathbf{x}^{*(0)}$ is feasible or not for an initial working set.

3.4.4 Examples for Hierarchical Optimization with Inequality Constraint

Let us consider again the mobile robot as shown in Figure 3.1. Suppose two tasks are assigned with a given desired effector's pose ${}^d\mathbf{s}$. The first priority task is to reach a target point ${}^d s_1$ as close as possible with the arm's horizontal position $q_1 + \cos(q_2)$, and the second level task is to position the mobile base within $q_l \leq q_1 \leq q_u$. The task function described in (3.2.1) can be cast as

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} (q_1 + \cos q_2) - {}^d s_1 \\ -q_1 + q_l \\ q_1 - q_u \end{bmatrix} \quad (3.4.118)$$

and the associated augmented Jacobian matrix is

$$\underline{\mathbf{J}}_3(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \mathbf{J}_3 \end{bmatrix} = \begin{bmatrix} 1 & -\sin q_2 \\ -1 & 0 \\ 1 & 0 \end{bmatrix}. \quad (3.4.119)$$

The vector \mathbf{y} is specified as follows:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -K_1 g_1 + {}^d s_1 \\ -K_2 g_2 \\ -K_3 g_3 \end{bmatrix}. \quad (3.4.120)$$

We use the integration interval $\Delta t = 0.1$ s, task function gains $K_1 = 1$, $K_2 = K_3 = 10$ in (3.2.6), and the regularization coefficient $\rho_1 = \rho_2 = \rho_3 = 0.5$. The purpose of task functions are to drive the values $g_1(\mathbf{q})$ to the origin as close as possible and satisfy the second level constraint

$$\begin{bmatrix} g_2 \\ g_3 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (3.4.121)$$

Consider two cases according to whether the two priority level tasks are simultaneously satisfied or not. At first, consider the target point ${}^d s_1 = 0.1$ and the position limits of mobile base are $q_l = 0.2$ and $q_u = 0.5$ in (3.4.118). The first priority reference point ${}^d s_1$ is reachable while satisfying the position limit constraints having the second priority. There is no task confliction. We compare the solutions of unregularized HQP and RHQP in terms of the magnitude of the task function, distance to the optimum, and robot configurations.

As depicted in Figure 3.6, all task function values converge to zeros regardless of regularization. The shaded areas represent the position range constraints of the mobile base. The convergence speed of the unregularized solution is faster than

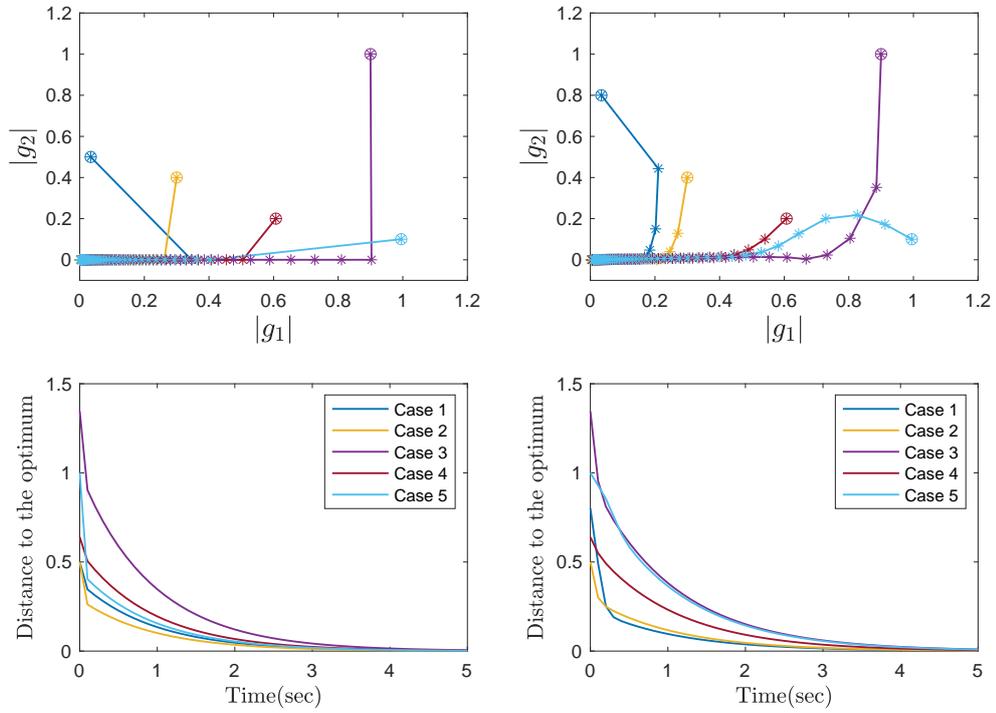


Figure 3.6: Numerical simulation results computed via unregularized HQP (left column) and RHQP (right column). Two tasks can be realized at the same time. We compare the results in terms of time history of the magnitude of the task function (top row), distance to the optima (bottom row). The convergence speed of the unregularized solution is faster than that of the regularized one. All task function values converge to zeros

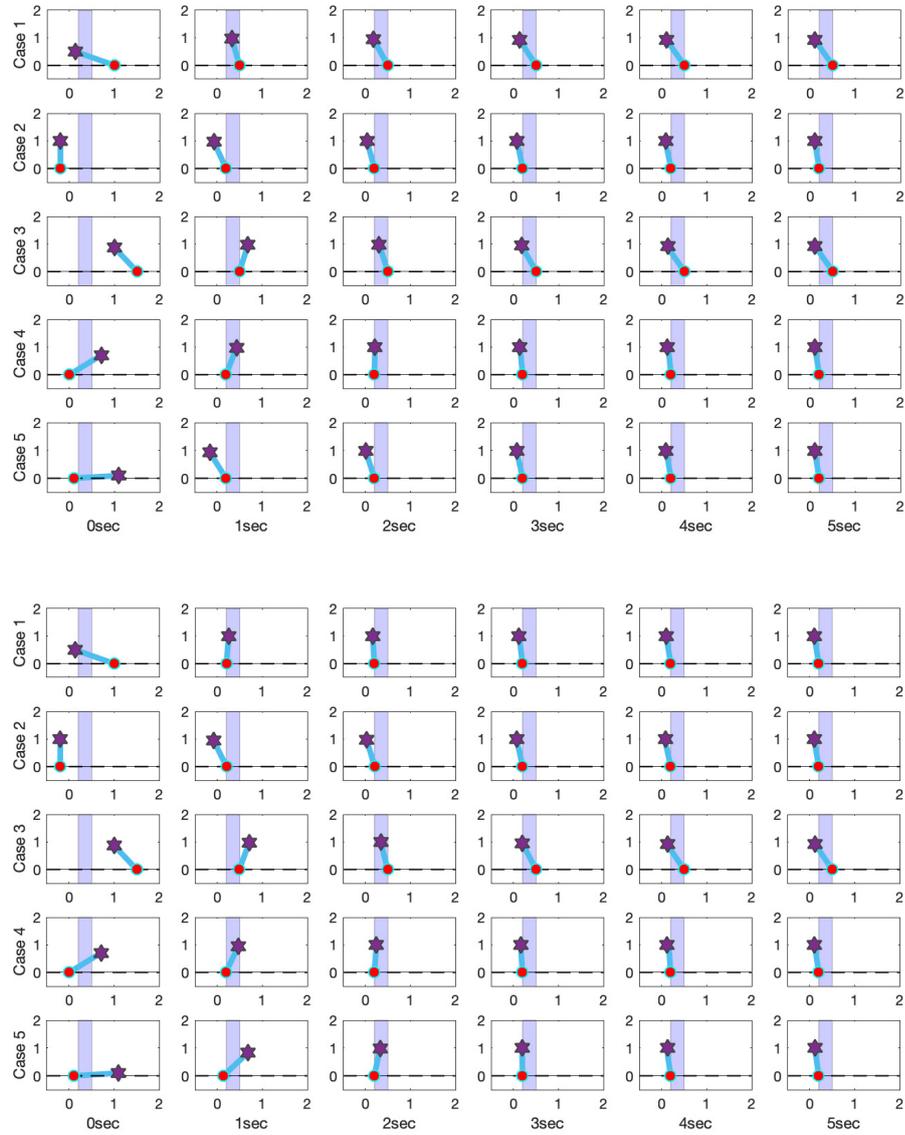


Figure 3.7: Time history of the robot poses computed via unregularized HQP (top row) and RHQP (bottom row) where target point is within the reachable space. The shaded areas represent the position range constraints of the mobile base.

that of the regularized one. Because RHQP penalizes the optimal velocity solution for singularity robustness. In Figure 3.7, the time history of the robot poses are presented. The position ranges are satisfied by properly activating an inequality constraint. The robots with the unregularized solution have already reached the target point at 3 s, but the robots with regularization reach the target at 4 s as you can see in Figure 3.7.

Next, the reference point of the hand is specified at $^d s_1 = 2.0$ and the position limits of mobile base are $q_l = 0.2$ and $q_u = 0.5$ in (3.4.118). The two task functions cannot be satisfied at the same time and the two tasks are conflicting. The joint angle $q_2 = 2j\pi$ ($j = 0, 1, 2, \dots$) makes the first level task function zero, but the second priority inequality constraints cannot be perfectly satisfied. The solution lies on algorithmic singularity. As can be seen from Figure 3.8 and 3.9, all solution sequences of unregularized HQP cannot converge to the optima. However, all RHQP solutions converge to the optima

$$\left[|g_1(\mathbf{q}^*)| \quad |g_2(\mathbf{q}^*)| \right]^T = \begin{bmatrix} 0 & 0.5 \end{bmatrix}^T \quad (3.4.122)$$

while respecting hierarchy. The second level task is achieved at $t = 1$ s as shown in Figure 3.9 (bottom row), but from $t = 2$ s this constraint cannot be satisfied anymore for executing the priority task. Nevertheless, the second task is achieved as much as possible by making the joint angle $q_2 = 0$. If q_2 has a nonzero value, then the second level objective has a larger task error.

As a result, we can confirm again every task is achieved as much as possible according to the priority, and the regularization can guarantee the convergence of the solution even if the solution is near singular points.

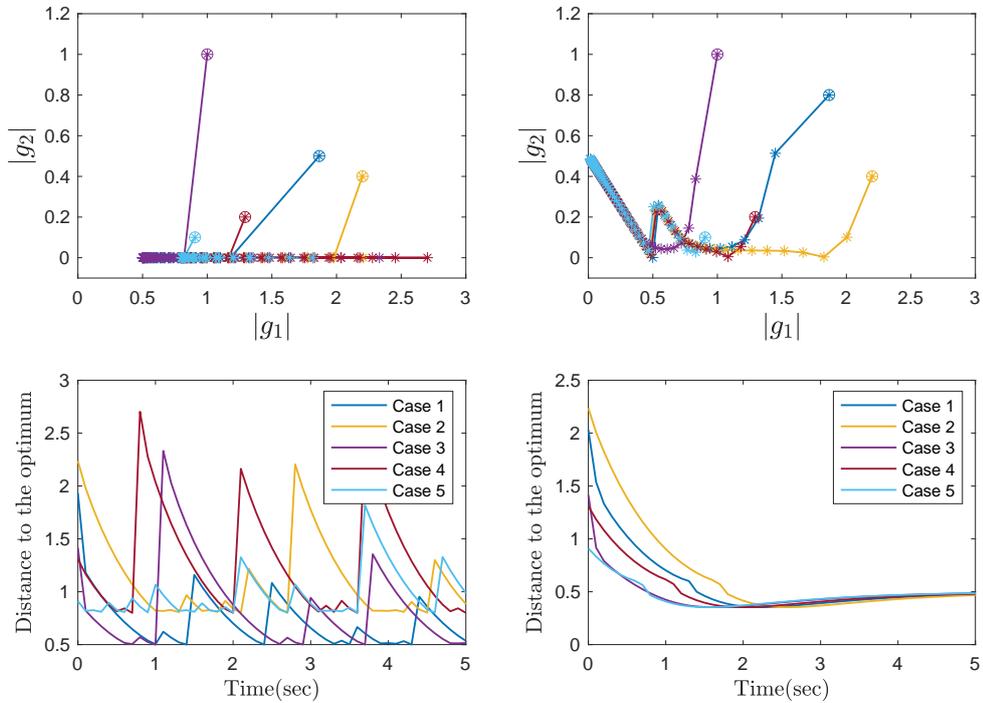


Figure 3.8: Numerical simulation results computed via unregularized HQP (left column) and RHQP (right column). Two tasks cannot be satisfied simultaneously and there is a conflict between two tasks. We compare the results in terms of time history of the magnitude of the task function (top row), distance to the optima (bottom row). As it can be seen, the solution sequences of unregularized HQP cannot converge to the optima while all solutions of RHQP converge to the optima $[|g_1(\mathbf{q}^*)|, |g_2(\mathbf{q}^*)|]^T = [0, 0.5]^T$ respecting hierarchy.

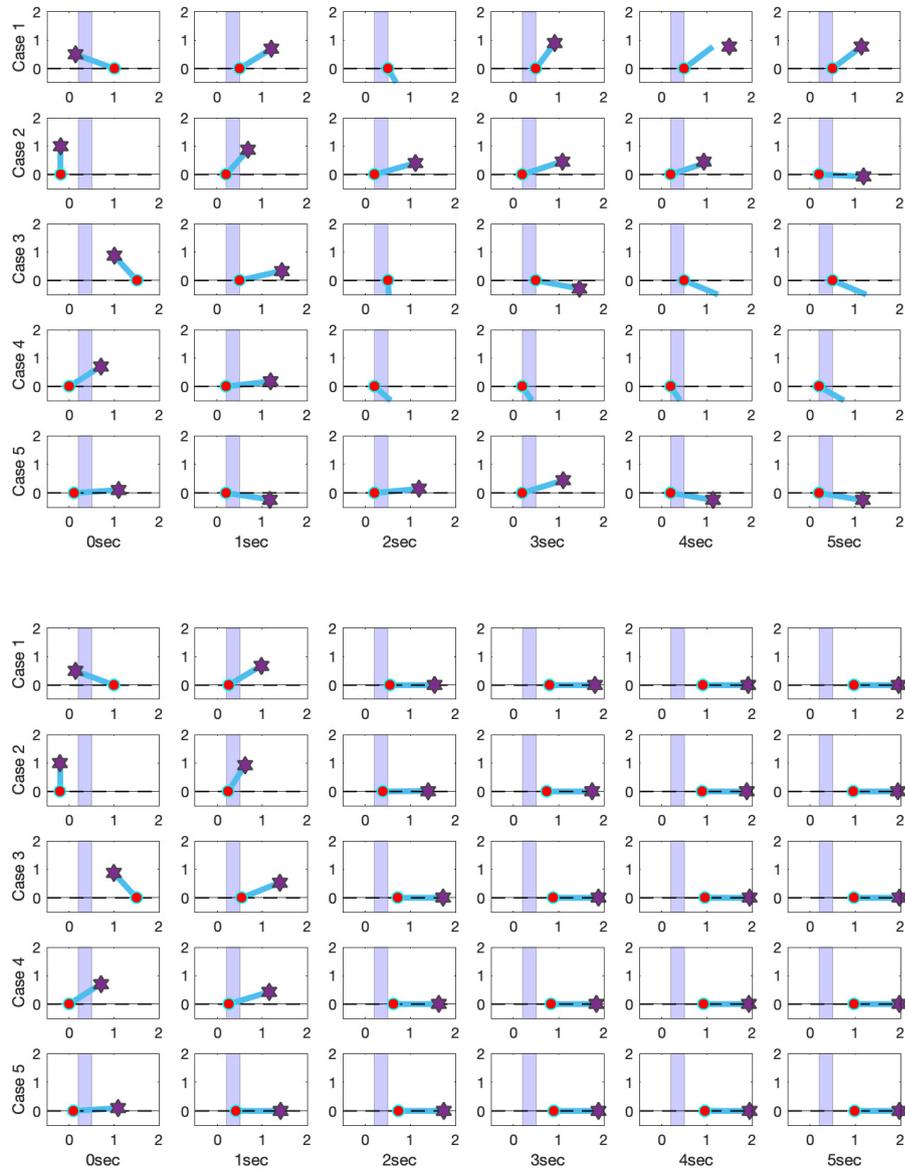


Figure 3.9: Time history of the robot poses computed via unregularized HQP (top row) and RHQP (bottom row) where target point is within the unreachable space. The shaded areas represent the position range constraints of the mobile base.

3.5 DLS-HQP Algorithm

For comparison, the DLS-HQP algorithm [44, 70] is briefly described. Similar to (3.3.30), each k level COD (Complete Orthogonal Decomposition) is computed as

$$\mathbf{J}_k \mathbf{Z}_{k-1} = \begin{bmatrix} \mathbf{U}_k & \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \mathbf{Q}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{Y}}_k^T \\ \tilde{\mathbf{Z}}_k^T \end{bmatrix} \quad (3.5.123)$$

$$= \mathbf{U}_k \mathbf{Q}_k \tilde{\mathbf{Y}}_k^T \quad (3.5.124)$$

where \mathbf{Q}_k is the upper triangular matrix. A singularity robust solution is

$$\mathbf{x}_k^* = \mathbf{x}_{k-1}^* + \mathbf{Y}_k \hat{\mathbf{Q}}_k^{-1} \mathbf{U}_k^T (\mathbf{y}_k - \mathbf{J}_k \mathbf{x}_{k-1}^*) \quad (3.5.125)$$

where

$$\hat{\mathbf{Q}}_k^{-1} := (\mathbf{Q}_k^T \mathbf{Q}_k + \rho_k^2 \mathbf{I}_k)^\dagger \mathbf{Q}_k^T. \quad (3.5.126)$$

With the solution (3.5.125), Lagrange multiplier $\boldsymbol{\mu}_k^*$ (3.4.108) can be obtained numerically with the optimal task error \mathbf{e}_k^* of (3.3.76), but references [44] and [70] do not take into account the regularization induced error \mathbf{u}_k^* in $\boldsymbol{\mu}_k^*$.

The RHQP algorithm based on SVD allows us to acquire the Lagrange multipliers $\boldsymbol{\mu}_k^*$ and $\underline{\boldsymbol{\mu}}_k^*$, however, the upper triangular matrix \mathbf{Q}_k of COD and the square matrix $\hat{\mathbf{Q}}_k^{-1}$ make it difficult to obtain two Lagrange multipliers in the analytical form. The value of \mathbf{u}_k^* in $\boldsymbol{\mu}_k^*$ and $\underline{\boldsymbol{\mu}}_k^*$ is small, but without considering it, the algorithm cannot properly handle inequality constraints. For example, DLS-HQP can incorrectly drop an active constraint. Then, at the next iteration for obtaining the optimal active set, this constraint would be added as a blocking constraint because it is violated by \mathbf{x}_k^* .

This alternating repetition is called *cycling* [16, 41, 55, 82, 83, 84]. In that case, the iteration counts and the computation time increase significantly while

the algorithm returns the same active set and solution as the previous steps without convergence. Moreover, an unnecessary constraint cannot be deactivated at the right time. We are going to show numerically these two inequality constraint treatment problems in the next chapter.

3.6 Concluding Remarks

We develop the fast and robust solver for regularized hierarchical quadratic program without resorting to any off-the-shelf QP solver for the fast translation of the task space objectives into the full-body joint space motion. As a complete reliable and rank revealing SVD centric approach is leveraged, we recursively obtain the singularity robust optimal solution, task error, hierarchical matrix decomposition (HMD), and Lagrange multiplier matrix necessary to solve inequality-constrained optimization.

We show that the optimal solutions of unregularized HQP of [44] can be exactly recovered from the RHQP optimum by setting each level regularization coefficient to zero. Therefore, we provide a generalization of the HQP framework. We mathematically show that regularization requires for us to trade off robustness against performance. The regularization error does not vanish if the regularization coefficients are not zero. Thus the priority task can interfere with all lower level tasks. This small coupling error is the only price we have to pay instead of guaranteeing convergence and robustness of the solution near singular points.

The RHQP solver enables to find the optimal solution very efficiently by considering all level's hierarchical tasks together with only one active set by using the Lagrange multiplier matrix. The multiplier matrix eliminates the iterative process used to find each level optimal active set. All necessary Lagrange multiplier and

optimal solution are acquired in a recursive procedure. Consequently, under the equality and inequality constraints, the proposed algorithm works fast. We modified the active set algorithm with the primal solutions, thereby we do not need a step direction nor a step length usually used in numerical optimization. The algorithm is simple and fast, and moreover it operates regardless of whether the initial point is feasible or not for an initial working set.

We verified the power of the regularization for the hierarchical optimization via the simple mobile manipulator example. The regularization can guarantee the convergence of the solution near singular points while the solution without regularization shows unstable behavior. It is also shown that the hierarchical structure can precisely resolve conflicts by creating a priority in which every task is achieved as much as possible according to the levels. In the next chapter, the effectiveness of the proposed method will be demonstrated through extensive numerical simulations and experimental tests with the real rescue robot, *Hercules*.

4

Rescue Robot Design and Experimental Results

4.1 Introduction

Over the last several years, a new generation of robots has emerged that shows great promise in being able to accomplish complex tasks associated with human behavior. For example, DARPA Robotics Challenge (DRC) has spawned real world humanoid robots and accelerated relevant technology advancement [1, 2, 3, 4, 5, 6]. DRC has a scenario for a large-scale disaster response given the hazard involved, and the quick response needed. Robots can be one of the best solutions in mitigating such events in a safe and timely manner.

However, DRC shows that there are still challenging problems including falls, failed attempts at tasks, long periods of robot inactivity, and operator errors, and it seems that some of problems can be addressed by a novel robot hardware design and a control strategy for real world robotic applications [7]. While most of DRC

teams used bipedal humanoid robots, some pursued statically stable hybrid mobile platform with a combination of legs and wheels. As a statically stable mobile platform provides a large, stable, and flexible support polygon, the hybrid platform is advantageous for movement in various terrains [73, 74, 75, 76, 77]. However, these DRC robots and other rescue robots do not attain all-terrain mobility on the challenging environments and do not have a high load carrying capacity for executing real rescue missions in a highly unstructured environment [78, 79, 80, 81].

The core design concepts of the Hurcules (HUmanoid ResCUe robot caLamity reSponse) rescue robot are firstly presented to realize how a mechanical structure can be developed to have an effect to both manipulation capability and all-terrain mobility. The upper body is designed humanoid in form with replaceable modularized dual arms. The lower body is featured with a hybrid tracked and legged platform to simultaneously acquire versatile manipulability and mobility on difficult terrain. A human like robot posture enables to execute complex manipulation tasks without concerning about balancing control. Yet the robot takes advantage of the stable tank like posture when the versatile mobility is required. As a result, Hurcules can efficiently perform a driving task, dangerous object manipulation, and casualty extraction missions by changing the pose and modularized equipments in an optimized manner.

As the second topic of this chapter, the effectiveness and performance of the proposed solver for regularized hierarchical quadratic program (RHQP) are proved through extensive numerical simulations and experimental tests with the Hurcules robot. We demonstrate that the task regularization and the inequality-constrained optimization are fully integrated via RHQP to guarantee the robustness and convergence of the solution in the face of a singularity. The proposed method can also acquire the optimal solution without showing any cycling and deactivate an

unnecessary constraint at the right time. Especially, we show that, without considering the regularization errors in Lagrange multipliers, the optimization algorithm brings about a cycling in the neighborhood of a singularity, and thus it yields the same solution and working set repeatedly to the earlier iteration. As the RHQP algorithm can impose the top-most priority on the physical inequality constraints such as joint angle ranges, it eliminates the possibility that the motion tasks violate the physical constraints.

The algorithm works fast enough to be used as a practical means of real-time motion control system. The control frequency is set at 1kHz for the Hurcules robot, and the hierarchical optimization algorithm should provide the calculated results within 1 *ms*. The average computation time for the 19-DoF rescue robot is less than 0.1 *ms* on a typical personal computer. Finally, we show that the Hurcules robot can successfully execute a dangerous object disposal and a casualty extraction in a highly unstructured environment.

4.2 Rescue Robot Design

The main question of this section to be answered is: “what is the design concepts for a rescue robot to provide both versatile manipulation capability and all-terrain mobility.” Because dextrous manipulability and reliable mobility are essentially required for a robot to rescue and assist humans in various environments. Although the development of a field robot involves various engineering details, we concentrate on the core design concepts based on the knowledge and experience gained while developing the Hurcules robot [96, 97].

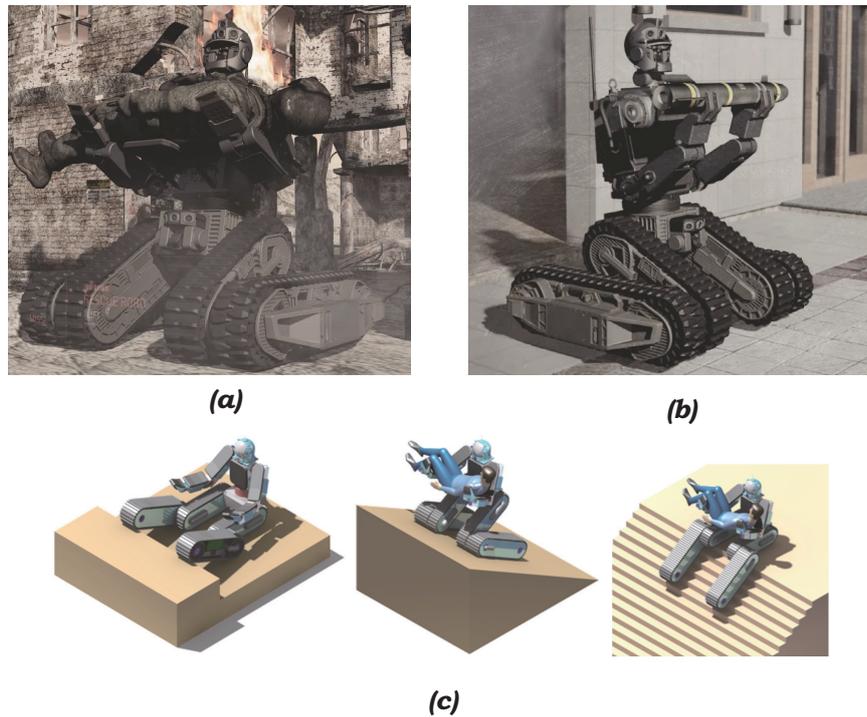


Figure 4.1: Missions of the rescue robot, Hurcules (HUMANoid ResCUE robot caLamity reSponse): (a) casualty extraction (b) explosive ordnance disposal and transportation (c) driving activities on various terrains

4.2.1 System Design

The goal of the Hurcules robot is to extract an impaired person in the battlefields or hazardous environments. Another crucial mission is to dispose a dangerous or explosive object as shown in Figure 4.1. This type of missions asks for two distinct and unique capabilities compared to other conventional humanoid robots. The first requirement is to have a high weight lifting capacity to extract a person over 120kg and the other is to traverse over various kind of difficult terrain. When designing, we set the following priorities in decreasing order of importance, and the main six

Table 4.1: Hurcules hardware specifications for rescue and manipulation missions

	For rescue mission	For manipulation mission
Weight lifting capacity	120kg	30kg per arm
DoF controlled by RHQP	13	19
DoF of a manipulator	4	7
Passive compliance	Elbow	Wrist and finger
Auxiliary equipment	Transfer bed	Finger type grippers
Normal standing height	1.9m	1.9m
Driving posture height	1.678m	1.678m
Weight	680kg	680kg
Maximum velocity	20km/h	20km/h

topics are consisting of:

- To enhance traversability over rough terrains and dexterous manipulability.
- A robust framework for real-time whole-body motion control.
- Maximizing payload capability to lift and transfer an injury over 120kg.
- Replaceable modularized hardware to efficiently perform a casualty extraction and a dangerous object disposal respectively.
- Passive compliance mechanism for a safe interaction with the environment.
- To maximize operation time.

The overall specification and schematic overview of Hurcules are shown in Table 4.1 and Figure 4.2 respectively. The easily replaceable two different lower arms

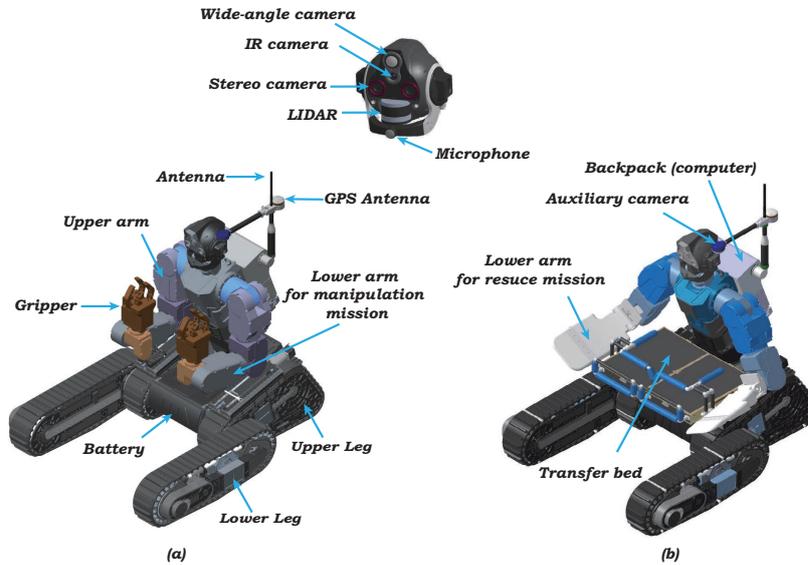


Figure 4.2: Configurations of the rescue robot, Hurcules for manipulation (left) and rescue (right) missions.

are designed. To carry out a manipulation mission, seven-DoF arms and finger type grippers are equipped. On the other hand, light weight four-DoF manipulators and automatically expandable transfer bed are utilized for the rescue mission. Rather than designing the manipulators satisfying all competing requirements at the same time, modular type arms are developed that can be easily replaced depending on the given mission.

As the configuration is depicted on Figure 4.3, the robot can change the pose and modularized equipments to maximize driving and manipulation performance. To manipulate objects in a human centered environment and to handle a heavy object efficiently, Hurcules stands upright to exploit all joints in the lower body platform as well as the upper body. As the robot attains near-human shape, it

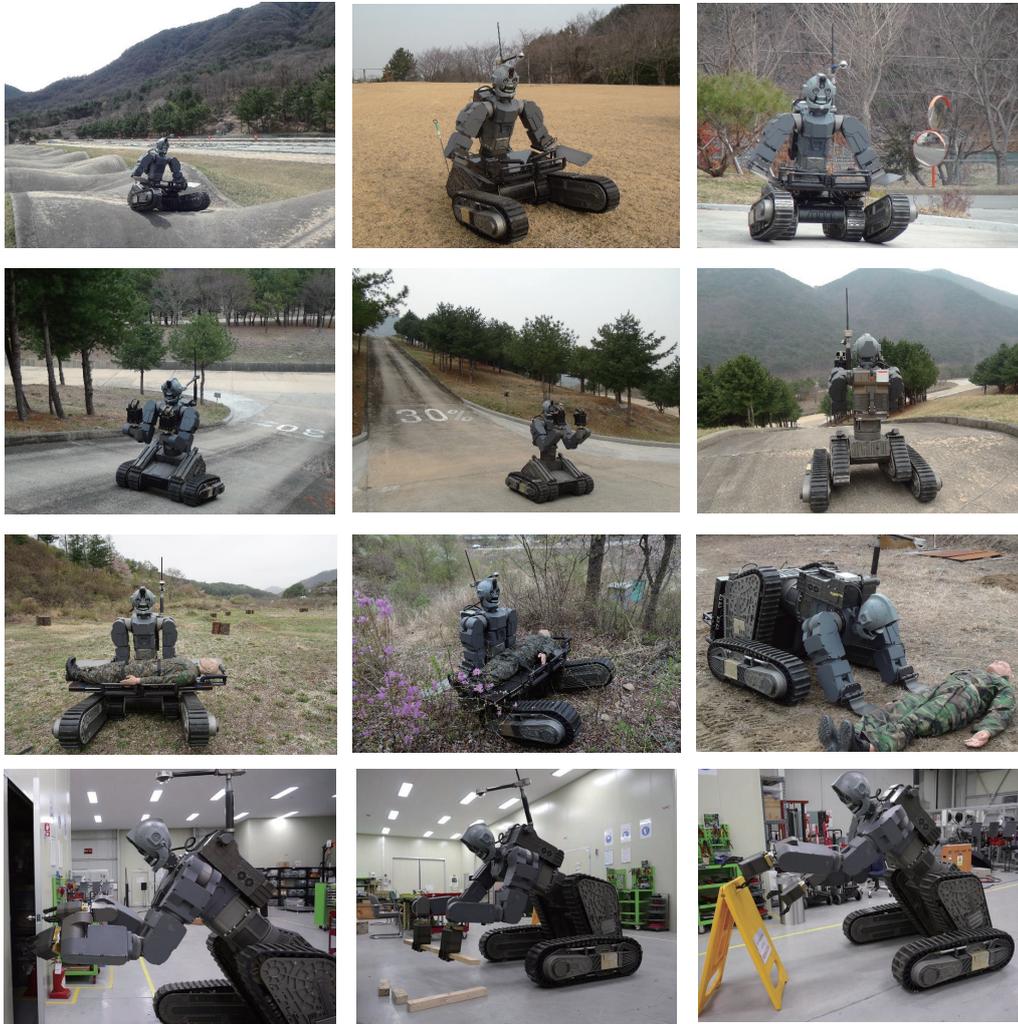


Figure 4.3: Various poses of the rescue robot, Hurcules: drive and task execution postures.

provides very dexterous manipulability and intuitive feedback for a distal operator. Thus, a short training time is usually required to proficiently operate the robot. Dual arm manipulators are useful for lifting and carrying injured as well as dealing with heavy objects. All joints are powered by electric motors. The anthropomorphic upper body for a manipulation mission includes 7 DoF on each arm, 1 for waist, 2 on neck, and 1 on each gripper. The lower body has 2 DoF on knee and hip joints and 1 DoF on each leg for driving. The stable tank like posture is exploited when the mobility is required.

A human-in-the loop operation is the primary working mode for human robot interaction. An operator is receiving visual information and the robot status, performing high level motion planning and control, communicating with the wounded in voice, and quickly intervening whenever critical event or emergency occurs. In order that the operator needs not concentrate on the details of multiple joint control, the user interface enables to control the robot in a natural and intuitive way. As frequently used motions are pre-programmed and optimized beforehand, the user can select and combine these motions to accomplish given tasks. Real-time motion control algorithms are essential to carry out complex or unstructured tasks in the operational space. The bimanual haptic master device (Geomagic Touch X) and joystick allow to intuitively synthesize task space motions of the slave robot.

The maximum weight lifting capacity is demonstrated in Figure 4.4. For the joint space trajectory tracking control of the robot, an inverse optimal robust Proportional Integral Derivative (PID) controller is designed to cope with a wide range of payload variation from 0kg to 120kg without any feedforward compensation [98, 99, 100, 53]. We can obtain not only a control law but also state and control input weighting matrices inversely from \mathcal{H}_∞ optimal control framework. Although



Figure 4.4: Maximum weight lifting capacity: (a) casualty extraction (120kg) (b) explosive ordnance transportation (30kg per arm)

this robust PID trajectory tracking controller cannot guarantee globally asymptotic stability, it promises extended disturbance input-to-state stability. Moreover, it provides two gain tuning laws. Please see Appendix for a detailed explanation.

The adoption of LIDAR (light detection and ranging), infrared and stereo cameras as well as a wide angle camera on the head can provide enhanced situational awareness. Especially, the infrared camera enables us to detect the injured who has body temperature from a far distance and helps to operate the robot at night. The line of sight of head mounted cameras and the auxiliary camera can be adjusted using active joints to monitor the state of the injured and environment. The pose of the target object is estimated by using the stereo cameras. With the help of the pose estimation information of the object, the RHQP control system can automatically plan a joint space trajectory making the grippers in an appropriate grasping configuration. This *look-and-move* approach can assist the robot operator to perform a grasp mission.

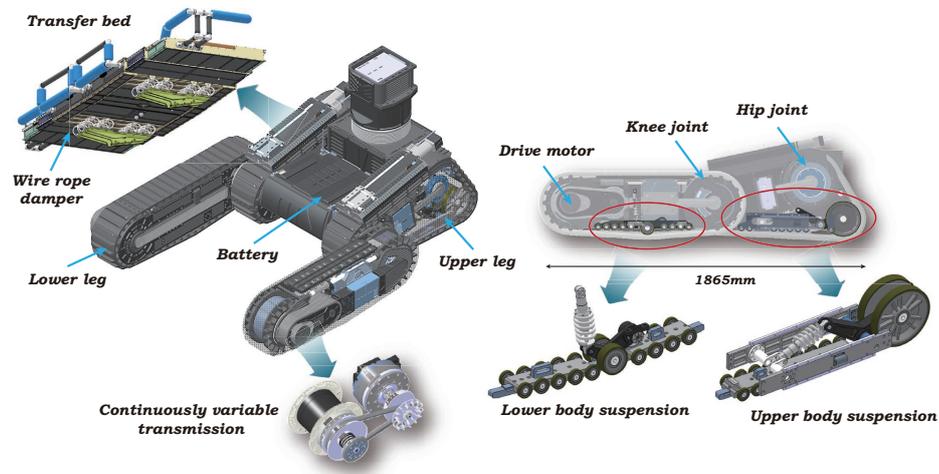


Figure 4.5: Mechanical design of the variable configuration mobile platform.

4.2.2 Variable Configuration Mobile Platform

The wheeled or tracked mobile robots are generally faster and more stable than legged robots. Relative to speed, wheels are the most energy efficient way to travel. The lower body of Hurcules is designed to be stable and sufficiently fast for traversing. At the same time, in order to obtain improved manipulability, the lower body is featured with individually controllable knee, hip, and waist joints as shown in Figure 4.5, 4.6 and 4.7.

As the lower and upper legs have continuous or caterpillar tracks, the mobile platform has better mobility over rough terrain. It can smooth out the bumps, glide over small obstacles and is capable of crossing trenches. In addition, tracks are much less likely to get stuck in soft ground, mud, and snow because they distribute the weight of the robot over a large contact area. This passively stable

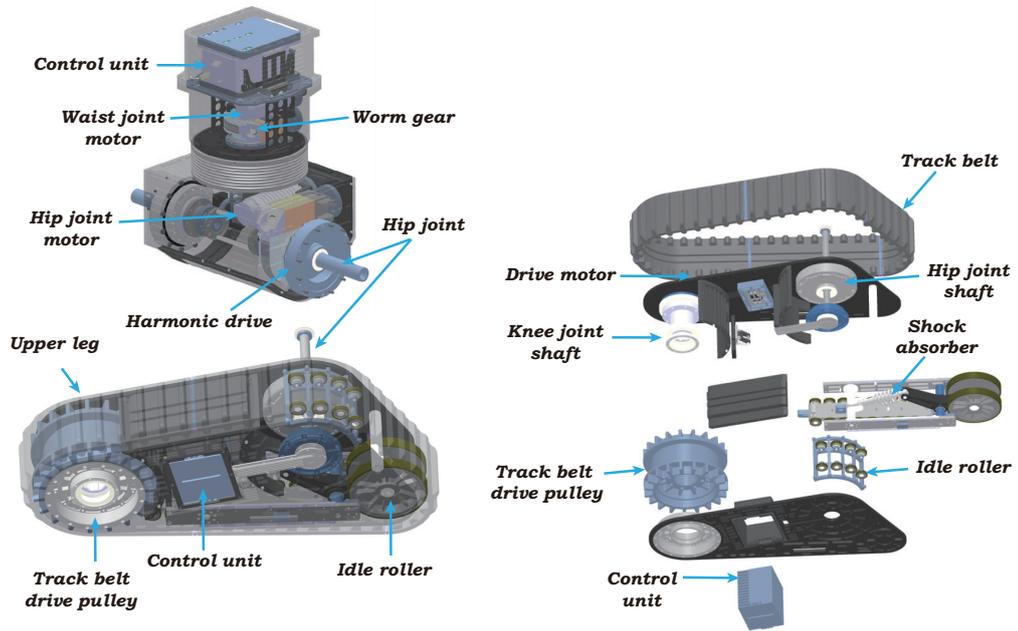


Figure 4.6: Pelvis and upper leg structure of the mobile platform.

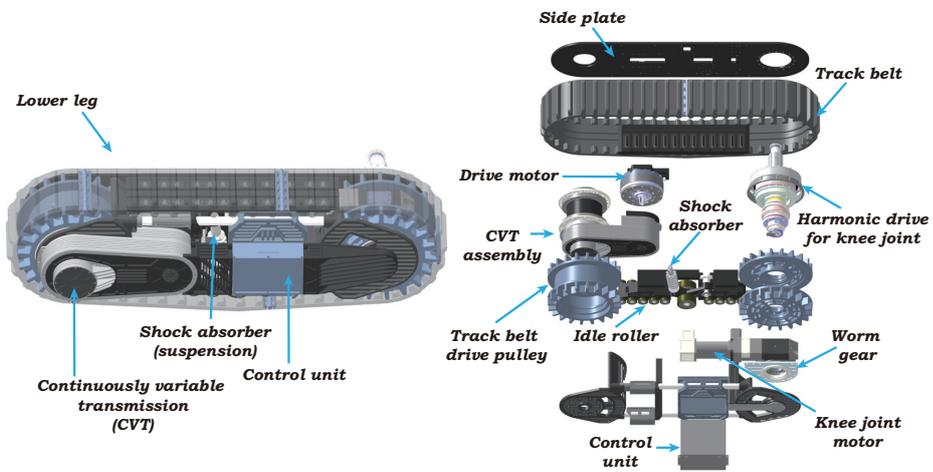


Figure 4.7: Lower leg structure of the mobile platform.

mobile platform has advantage of maintaining balance and handling a heavy object on uneven areas. Finally, skid-steered tracked platform can turn easily in place without forward and backward movement by driving the tracks in opposite directions.

To safely transfer an injured, an automatically expandable transfer bed is applied, and it can be attached or detached depending on the mission. When traveling over rough ground, the robot cannot avoid shock and vibration which inevitably lead to structural fatigue and deteriorate riding quality and steering performance. Multiple springs with shock absorbers are mounted on each legs and the transfer bed to alleviate these effects.

When driving on an unpaved road, a large driving torque is necessary. On the contrary, a fast driving performance is needed on a paved road. As it is difficult to achieve the two competing requirements only with a motor, the vehicle usually uses a transmission. However, it is not suitable for humanoid robots due to the heavy weight and bulky size. To cover wide ranges of driving speed and torque, a small continuously variable transmission (CVT) is adopted which is often used on motorcycles. The gear ratio can be continuously varied by the centrifugal force proportional to the rotational speed of the drive shaft. A large torque is acquired at low speeds with a gear ratio of 3:1 (input:output), a fast maneuver performance is achieved at high speeds with a ratio of 1:1. The continuous gear ratio is provided between low and high speed. Thus, the mobile platform has the ability to move with the maximum velocity of up to 20km/h and to drive on 30 degree slope terrain. Consequently, this variable configuration mobile platform helps us to deploy *Hurcules* in even more challenging outdoor environment.

4.2.3 Dual Arm Manipulators

The powerful electric motors are employed and the actuated torques are delivered by means of mechanical transmissions with large gear reduction ratio as illustrated in Figure 4.8. As the drive trains have power-on-to-disengage magnetic safety breaks, the robot is able to hold the limb at the output side without power consumption. The rescue scenario is composed of several small missions such as standby, driving, acquisition, and returning to the command post. The power of dual arms can be completely removed when there is no upper body movement. Only the acquisition and manipulation missions require the upper body motion and the power of dual arms is completely shut down when driving. This power management strategy enables the robot to be highly energy efficient. Therefore, the continuous operation time is significantly extended up to two hours based on the rescue scenario.

As the environment becomes unstructured, the interaction between the end-effector and environment more complex and uncertain. Moreover, it is very difficult for an operator to acquire accurate information of the robot and environment. The control, planning, and an operator error may cause an excessive contact force during robot-environment interaction. A passive compliance mechanism can effectively mitigate this problem by modifying the end-effector trajectory when the interaction forces arises. This passive approach is simple and the response is much faster than active compliance control. As shown in Figure 4.8, a passive elastic components are applied to the elbow, wrist, and finger joints.

Nevertheless, the passive compliance mechanism can only handle small deviations from the rest position and it is difficult to accurately regulate the interaction force. In order to overcome these drawbacks and to obtain a reasonable force

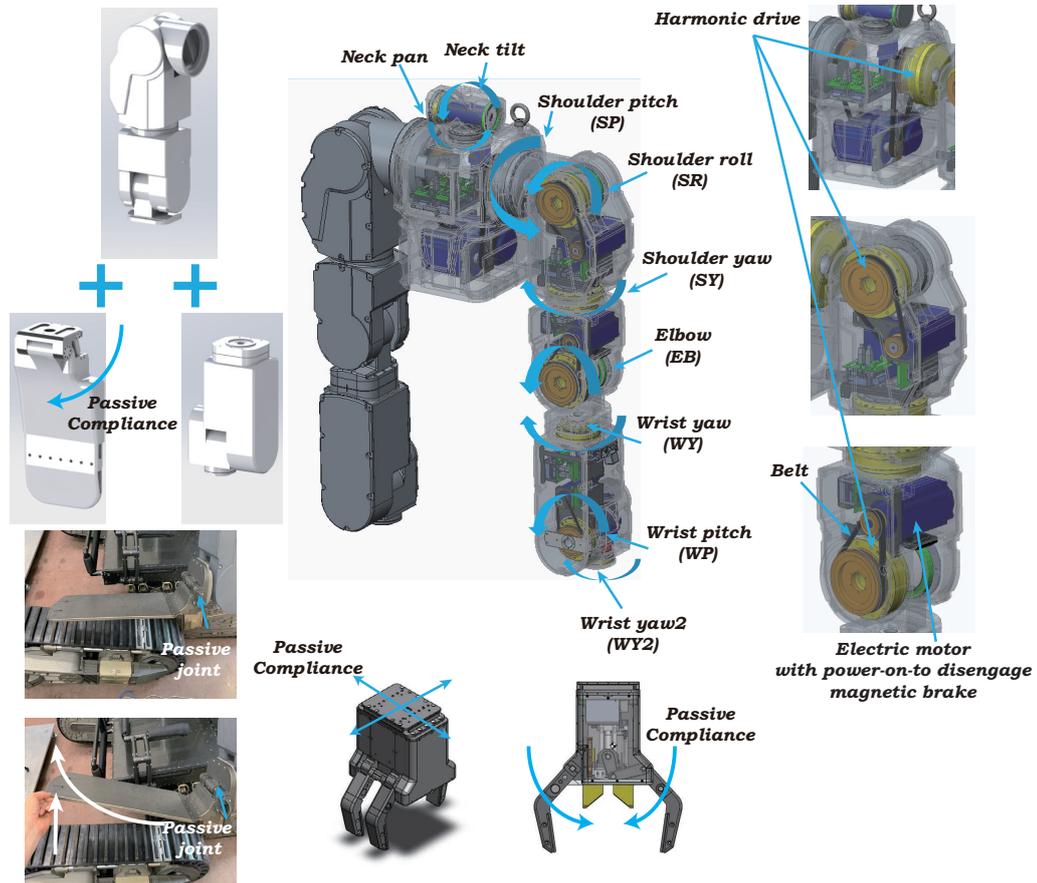


Figure 4.8: Mechanical design of the dual arm manipulators with replaceable modular type arms.

regulation accuracy, direct force control is used in combination with the passive compliance element. The active force control strategy is the topic of Chapter 5.

4.2.4 Software Architecture

The torso and back pack contain three computers, network device, inertial measurement unit, and power electronics. The navigation computer handles GPS and IMU data, and the vision computer processes image data. The motion control computer is allocated for managing whole-body motion control where real time control tasks are executed on the Linux Ubuntu 16.04 operating system. To strength the capability of real time tasks, Xenomai patch is used. The latest industrial EtherCAT field bus provides higher bandwidth to communicate with local joint controllers, and thus 1kHz frequency sensing and control are enabled. The torque reference command resulting from the PID controller is sent to each local joint current controller via EtherCAT field bus.

Software structures are depicted in Figure 4.9. The inter-process communication is working via shared memory, and it has an advantage in data exchange speed and implementation convenience. As we easily select controlled objects either a dynamic simulation model or real robot hardware without any software and hardware modification due to the shared memory structure. The integrated simulation environment helps us to expedite the software development process. Before starting the experiment, all control algorithms are tested in advance through the dynamic model in a virtual environment provided by Gazebo [101].

Socket handler processes all messages coming from an operator, navigation and vision computers, and stores the data in the shared memory. The control data is passed to Message parser and then delivered to Real-time Control (RC) task. In every period, the motion control RC task calculates the joint references with

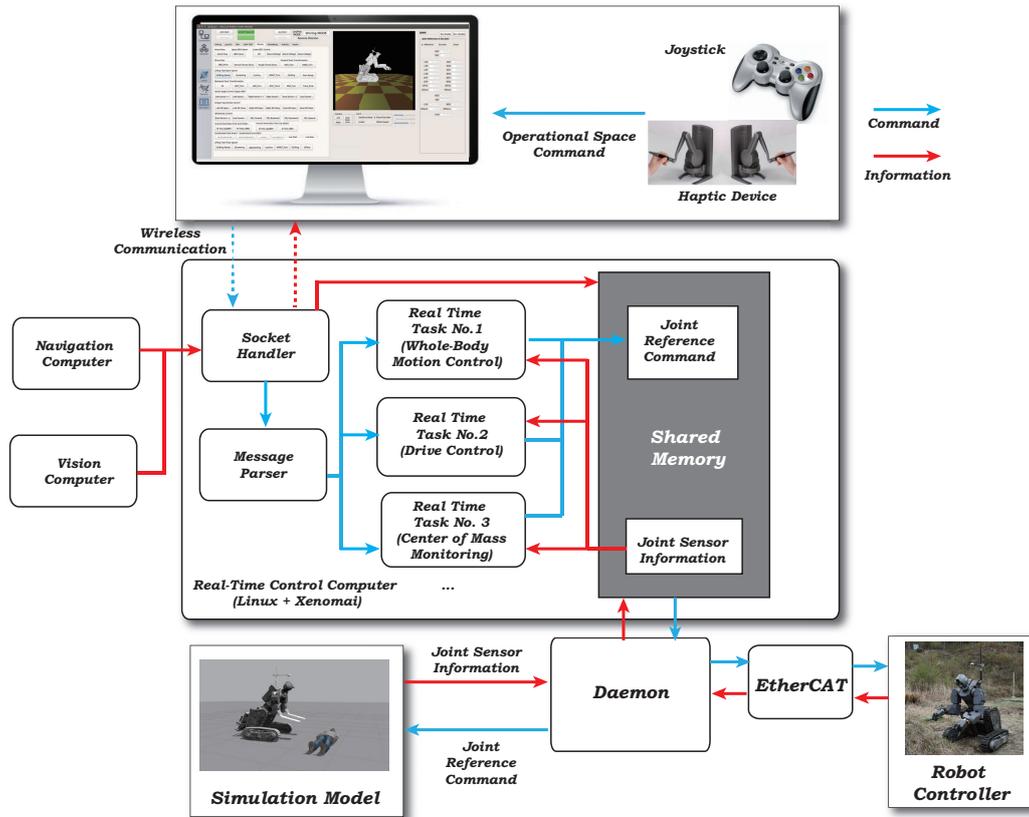


Figure 4.9: Block diagram of software structures

the RHQP algorithm and then writes the commands in the shared memory. The concurrently working multiple RC tasks contain various algorithms such as RHQP, driving control, center of mass (COM) monitoring, and self-collision detection. The Daemon process can directly access the hardware and handles the joint sensor data and command with the shared memory. The control frequency of the thread is set at 1kHz and all algorithms of RC tasks should provide the results within 1 *ms*. Please refer to [76, 102] for a detailed explanation.

4.3 Performance Verification for Hierarchical Motion Control

The RHQP algorithm is applied to the Hurcules robot for motion control. As the configuration is shown in Figure 4.10, a human operator can perform high-level motion planning and control through the bimanual haptic device and joystick while receiving visual information. The weight of the robot is approximately 680 kg, and the height of the posture shown in Figure 4.10 (b) is 1.9 m.

A 19-DoF robot is composed of two 7-axis arms, two axes on the neck, and one on the waist, hip, and knee joints, respectively. The RHQP solver allows the distal operator to intuitively control all level tasks without being overly concerned with the joint range constraints, task conflicts, and singularities. The aim of the solver is to compute the joint space reference angle \mathbf{q}^* and angular velocity $\dot{\mathbf{q}}^*$, as shown in Algorithm 1, and then transmit them to the local joint controllers.

An i7-6700 CPU with a base frequency of 3.4 GHz is used. Numerical computations of the SVD and COD are realized using *JacobiSVD* and *CompleteOrthogonalDecomposition*, respectively, provided by the Eigen C++ library [103].

4.3.1 Real-Time Motion Generation

For considering task hierarchy and singularity robustness, the RHQP framework (3.4.95) can be utilized by defining the following relations:

$$\mathbf{x}_k := \dot{\mathbf{q}}_k \quad (4.3.1)$$

$$\mathbf{y}_k := -\mathbf{K}_k \mathbf{g}_k + {}^d \dot{\mathbf{s}}_k, \quad (4.3.2)$$

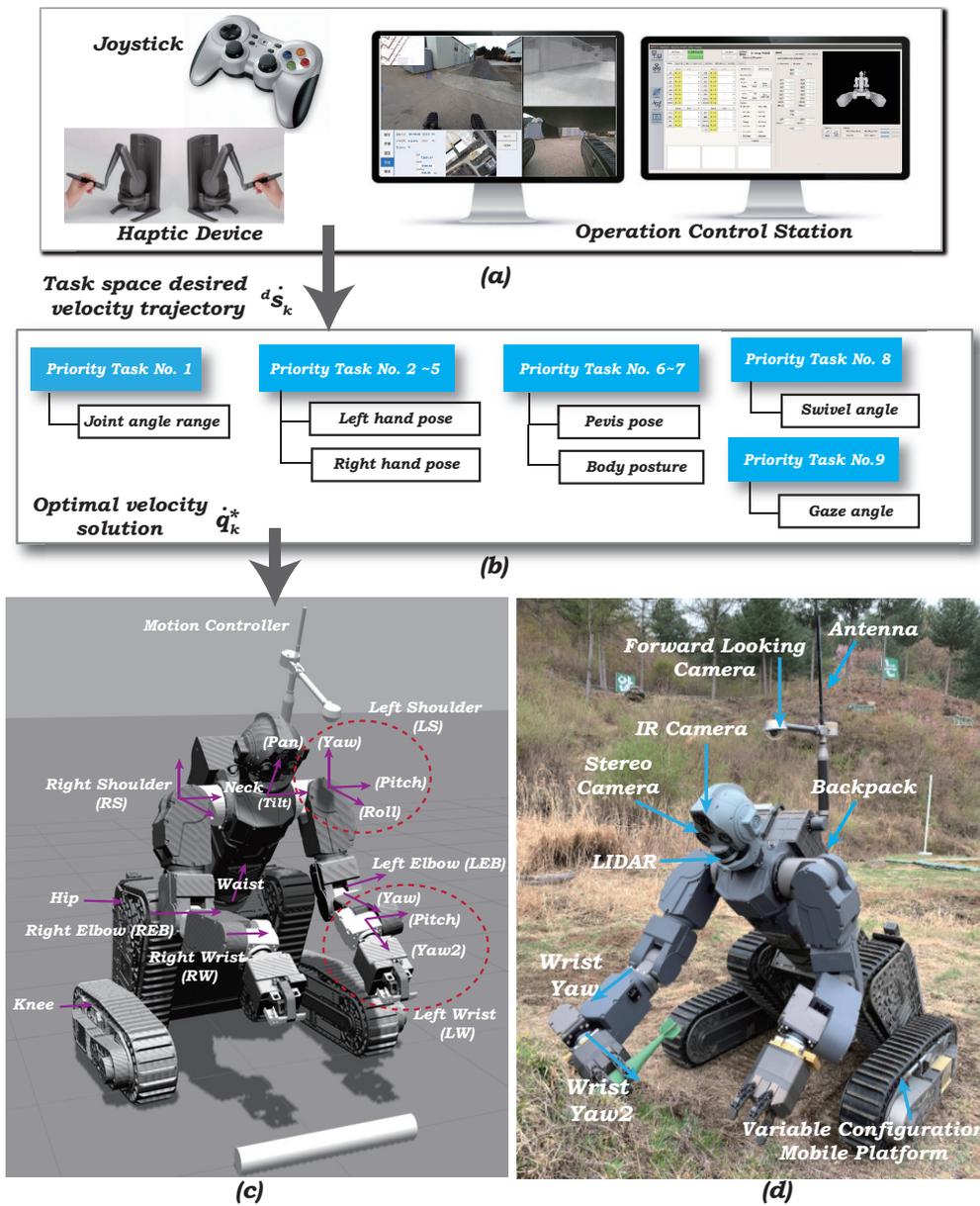


Figure 4.10: Configuration of Hurcules robot: (a) Joystick and haptic device for the task space motion synthesis (b) RHQP algorithm (c) Joint configuration in which the arrows represent the axis of rotation. (d) Visual sensors

where $\mathbf{g}_k := \mathbf{s}_k - {}^d\mathbf{s}_k$. The RHQP problem of (3.4.98) can be rewritten as

$$\min_{\dot{\mathbf{q}}_k, \mathbf{e}_k} \frac{1}{2} \|\mathbf{e}_k\|_2^2 + \frac{1}{2} \rho_k^2 \|\mathbf{Y}_k^T \dot{\mathbf{q}}_k\|_2^2 \quad (4.3.3a)$$

$$\text{subject to } \mathbf{J}_k \dot{\mathbf{q}}_k - \mathbf{y}_k \leq \mathbf{e}_k \quad (4.3.3b)$$

$$\underline{\mathbf{J}}_{k-1} \dot{\mathbf{q}}_k - \underline{\mathbf{y}}_{k-1} \leq \underline{\mathbf{e}}_{k-1}^* \quad (4.3.3c)$$

Similarly, the task error, hierarchical matrix decomposition, Lagrange multipliers, and all other solutions described in Chapter 3 can be obtained.

The motion specifications or constraints can be assigned in the operational space and/or joint space. The task space velocity command ${}^d\dot{\mathbf{s}}_k$ is given by the operator, and the position command ${}^d\mathbf{s}_k$ is computed by numerically integrating ${}^d\dot{\mathbf{s}}_k$ over time. Then the RHQP algorithm is exploited to find joint space reference position \mathbf{q}^* and velocity $\dot{\mathbf{q}}^*$. Therefore, the solver is used for the first-order closed-loop inverse kinematics scheme to generate joint space motions [28, 63, 94, 95, 104, 88].

The optimal solution in (3.3.37) can be rewritten as follows:

$$\dot{\mathbf{q}}_k^* = \sum_{j=1}^k \left(\mathbf{Y}_j \hat{\Sigma}_j^{-1} \mathbf{U}_j^T ({}^d\dot{\mathbf{s}}_k - \mathbf{K}_k \mathbf{g}_k - \mathbf{J}_j \dot{\mathbf{q}}_{j-1}^*) \right) \quad (4.3.4)$$

$$= \dot{\mathbf{q}}_{k-1}^* + \mathbf{Y}_k \hat{\Sigma}_k^{-1} \mathbf{U}_k^T ({}^d\dot{\mathbf{s}}_k - \mathbf{K}_k \mathbf{g}_k - \mathbf{J}_k \dot{\mathbf{q}}_{k-1}^*), \quad (4.3.5)$$

where

$$\hat{\Sigma}_k^{-1} := (\Sigma_k^2 + \rho_k^2 \mathbf{I}_k)^\dagger \Sigma_k. \quad (4.3.6)$$

Because the initial posture $\mathbf{q}(\mathbf{0})$ is known, the reference joint positions can be obtained by integrating velocities $\dot{\mathbf{q}}^*(t_k)$ over time

$$\mathbf{q}^*(t_{k+1}) = \mathbf{q}^*(t_k) + \int_0^{\Delta t} \dot{\mathbf{q}}^*(\zeta) d\zeta, \quad (4.3.7)$$

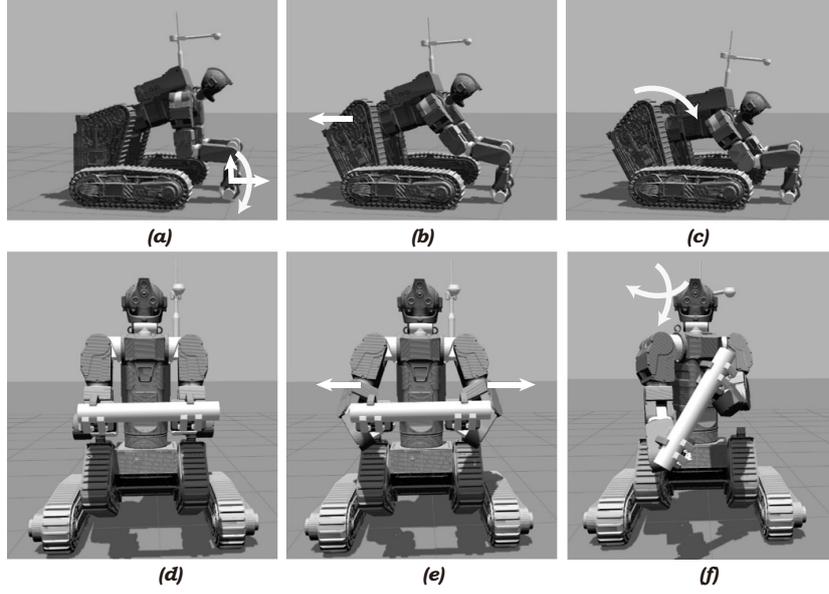


Figure 4.11: Representative tasks for manipulation. (a) position and orientation of hand (b) pelvis traversal position control (c) pelvis pitch angle control (d,e) swivel angle control (f) gaze control.

where Δt is integration interval.

A task specifies a bound on a function relying only on the configuration vector, it can be expressed as follows: \mathbf{q}

$$\mathbf{s}_k^l \leq \mathbf{g}_k(\mathbf{q}) \leq \mathbf{s}_k^u. \quad (4.3.8)$$

The first-order Taylor approximation yields the linear constraint as follows:

$$\mathbf{J}_k(\mathbf{q})\dot{\mathbf{q}} \leq \frac{\mathbf{T}_k}{\Delta t}(\mathbf{s}_k^u - \mathbf{s}_k) \quad (4.3.9)$$

$$-\mathbf{J}_k(\mathbf{q})\dot{\mathbf{q}} \leq -\frac{\mathbf{T}_k}{\Delta t}(\mathbf{s}_k^l - \mathbf{s}_k) \quad (4.3.10)$$

where \mathbf{T}_k is the time horizon and Δt is integration interval.

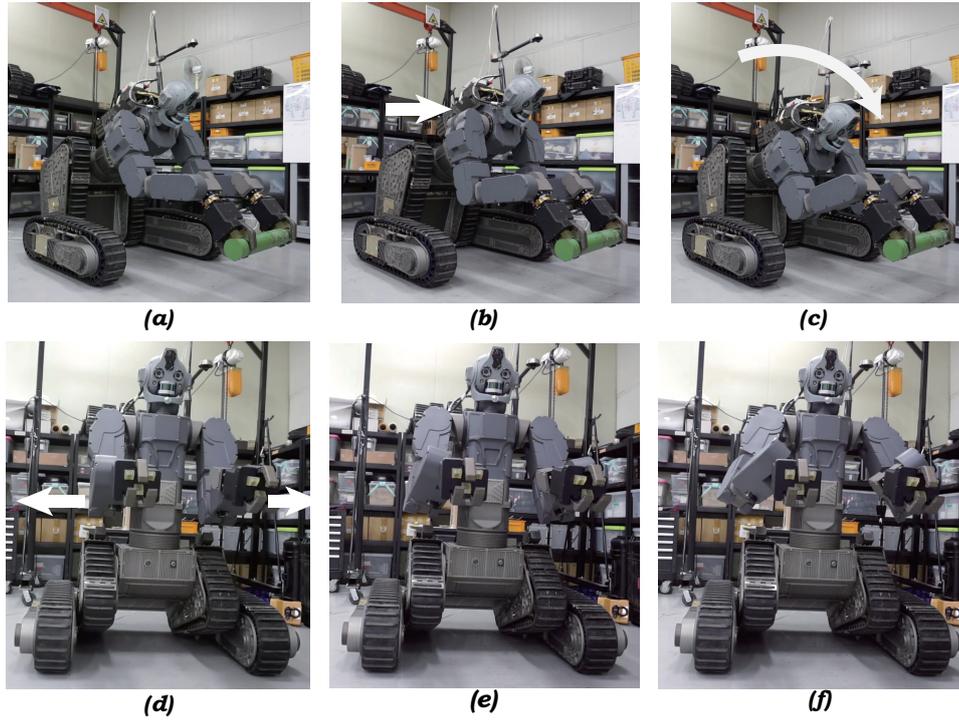


Figure 4.12: Pelvis pose and swivel angle control: (a) ready for manipulation (b) pushing the pelvis forward (15cm) (c) lowering the pelvis down (d, e, f) swivel angle control.

4.3.2 Task Specifications

The tasks are composed of controlling the positions of the left and right hands (\mathbf{g}_{lhpos} , \mathbf{g}_{rhpos}) and orientations (\mathbf{g}_{lhor} , \mathbf{g}_{rhor}) (12-DoF). The others are to control the pelvis traversal position and pitch angle \mathbf{g}_{pel} (2-DoF), body posture \mathbf{g}_{body} (1-DoF), swivel angle \mathbf{g}_{swivel} (2-DoF), and gaze \mathbf{g}_{gaze} (2-DoF). Representative tasks are shown in Figure 4.11 and 4.12.

The swivel angle is defined as an angle between the plane passing through the

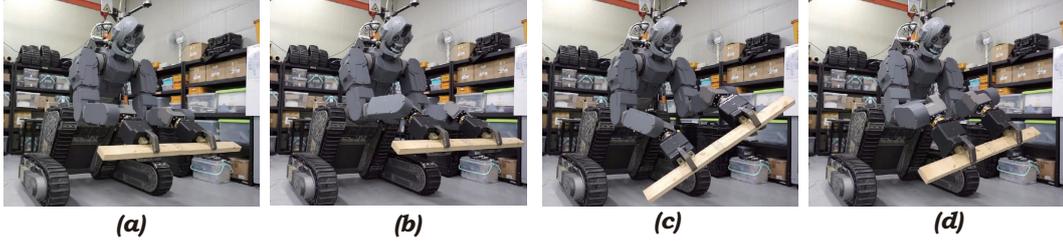


Figure 4.13: Constrained dual arm manipulation: (a) ready for manipulation (b) moving hands sideways (c) rotating hands in roll direction (d) rotating hands in roll and yaw direction.

arm and a reference torso plane [105, 106]. It is useful to avoid self-collision between arms and torso as well as to make a robot to execute a mission with human like posture. The body posture task is to force the torso to look forward as much as possible. The joint operation range \mathbf{g}_{joint} has the top priority among all constraints and the regularization parameter $\rho_k = 0.1$ is used for all level tasks.

The eight levels of tasks are established in decreasing order of importance, i.e., $\mathbf{g}_{joint} \prec \mathbf{g}_{lhpos} \prec \mathbf{g}_{rhpos} \prec \mathbf{g}_{lhor} \prec \mathbf{g}_{rhor} \prec \mathbf{g}_{pel}$, $\mathbf{g}_{body} \prec \mathbf{g}_{swivel} \prec \mathbf{g}_{gaze}$. The notation $\mathbf{g}_A \prec \mathbf{g}_B$ implies that \mathbf{g}_A task has higher priority than \mathbf{g}_B . The joint angle range constraint has top priority because unless the joint angle limits expressed by inequality constraint is explicitly included in the optimization, then the solution is not physically applicable to the real robot. Because the motions of real robots as well as Hurcules are severely restricted by each joint operation range due to collision of mechanical structure and additional equipments such as battery and backpack and the stiffness of internally routed cables as shown in Figure 4.10.

As can be seen in Figure 4.13, in order to handle a heavy object, dual arm manipulation has also been used. Because bimanual manipulation has several advantages over single arm, for example, to provide similarity to human form factor

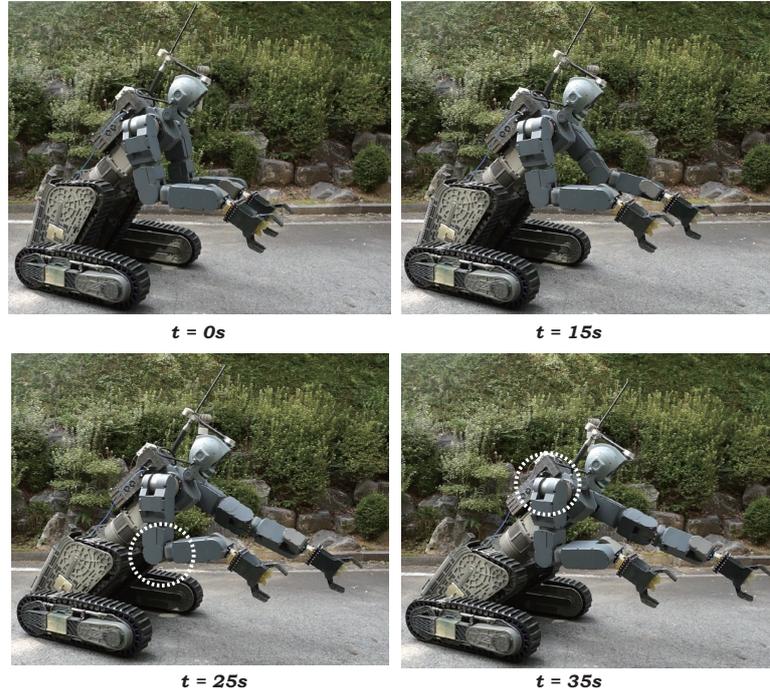


Figure 4.14: Stretching motion of the left arm for grasping a distal object with the RHQP solver. The dashed line represents the activated joint limit constraint at REB (right elbow) and RS (right shoulder) Roll joints.

of a human operator, flexibility, and stiffness [107]. When forming a closed kinematic chain by means of fixed grasps, the high structural stiffness and strength can be acquired at the price of having a reduced workspace. Here, instead of solving loop closure equations, we impose a rigid body kinematic constraint on each individual hand to generate coordinated Cartesian motion.

4.3.3 Singularity Robust Task Priority

To verify the singularity robustness of the proposed solver, we compare the three solutions of RHQP, HQP [44], and DLS-HQP [70]. Recall that DLS-HQP uses the COD and DLS-inverse, as described in (3.5.125), and it does not consider the regularization error \mathbf{u}_k^* in the Lagrange multipliers $\boldsymbol{\mu}_k^*$ in (3.4.108) and $\underline{\boldsymbol{\mu}}_k^*$ in (3.4.109). In particular, the latter $\underline{\boldsymbol{\mu}}_k^*$ is difficult to acquire analytically due to the triangular matrix of the COD. The active set method described in Chapter 3 is applied to all algorithms.

When the left hand reaches out to grasp a distal object, singularity is encountered, as shown in Figure 4.14. Because the left arm is outstretched, the mobility is reduced and the corresponding Jacobian becomes rank deficient. Three numerical solutions are presented in Figure 4.15. From $t = 17$ s, the left arm velocity of RHQP begin to decrease gradually because the regularization in (3.3.39) has a strong impact on the solution as the singular value decreases near singularity as shown in Figure 3.2. However, the HQP solution grows unboundedly, and an excessive joint velocity occurs. Every physical robot has the maximum allowable joint velocity, which prevents the solution from diverging indefinitely, but induces oscillations as depicted in Figure 4.15 (middle). Thus, the state of the robot is completely uncontrollable, and the posture becomes arbitrary.

As can be seen from Figure 4.16, DLS-HQP shows a cycling in the singularity region. The activation of a blocking constraint and its incorrect deactivation are repeated in turn, and thus the number of iterations increases significantly. Here, to prevent this infinite repetition, the maximum number of iterations is limited to 12, and the corresponding constraint is forced to be active at the maximum iteration. If the constraint is not intentionally activated, then the joint angle quickly reaches

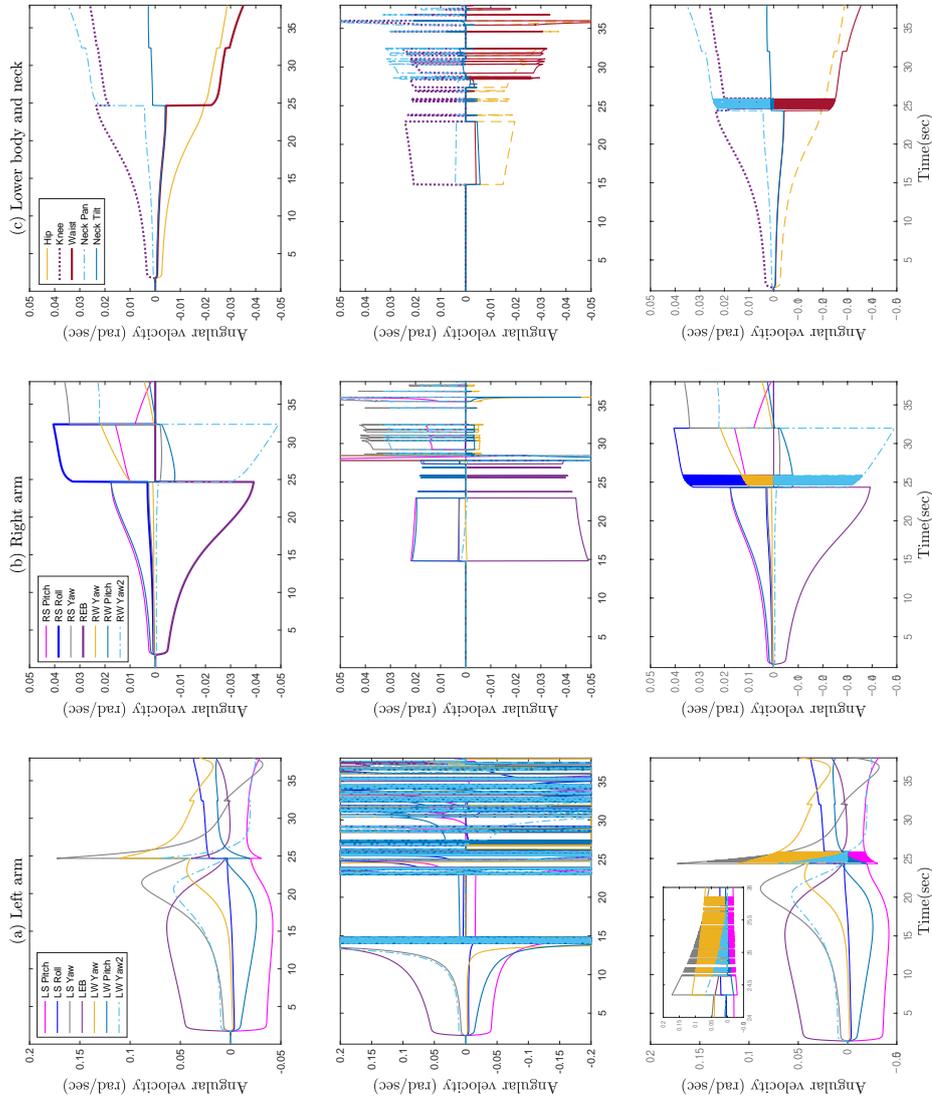


Figure 4.15: Three velocity solution trajectories of RHQP (top row), HQP without regularization (middle row) and DLS-HQP (bottom row): (a) left arm (b) right arm (c) lower body and neck.

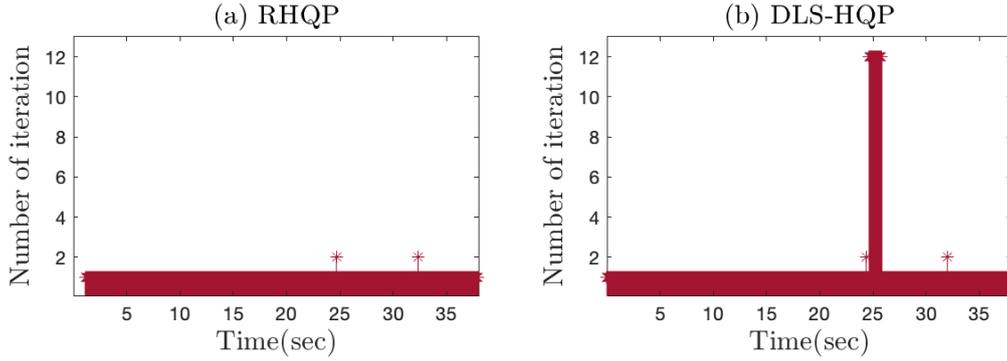


Figure 4.16: Number of iterations of RHQP (Left) and DLS-HQP (Right). The RHQP algorithm can find the optimal active set in 2 iterations. The DLS-HQP solution shows cycling from 24 s to 26 s with the maximum number of iterations.

the limit condition, and the robot enters emergency mode for safety. Even if a constraint is forced into activation, it is incorrectly dropped and the cycling occurs again at the next time step. This cycling significantly increases the computation time, and the solution is not always obtained in 1 *ms*. In this case, the solution is set to zero for safety, and therefore the joint angular velocities often become zero from 24 s to 26 s, as shown in Figure 4.15 (bottom).

It is worth noting that in the absence of a task conflict the RHQP solution trajectories of the right arm, lower body, and neck, shown in Fig. 4.15 (top row (b) and (c)), have nonzero angular velocities for up to 24 s despite the reference movement command being only given by the left hand. It is because of the regularization error. By recalling the optimal task error of (3.3.80), the $k - 1$ level solution \mathbf{x}_{k-1}^* affects all following j levels for $j > k - 1$ through the regularization parameter $\rho_j \neq 0$, and the regularization error \mathbf{u}_j^* occurs. This error is also observed in the DLS-HQP solution shown in Fig. 4.15 (bottom row (b) and (c)). By

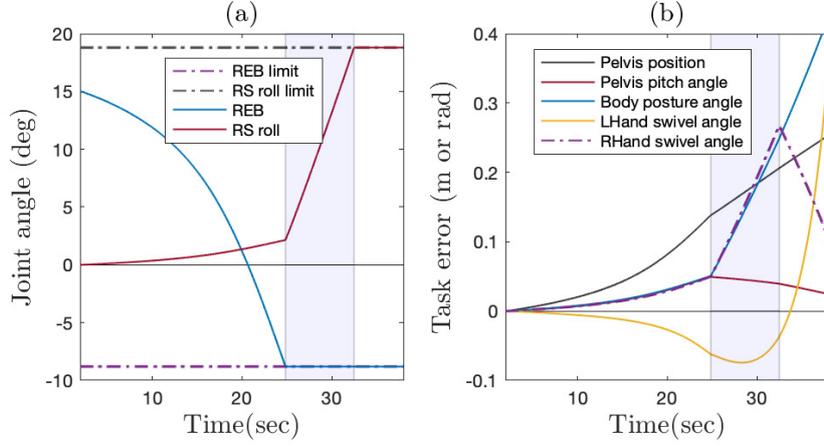


Figure 4.17: Reference joint angle trajectories (left) and task errors (right) of RHQP. (Left) The elbow and shoulder roll joint constraints are activated at $t = 25$ s and $t = 33$ s respectively. (Right) The RHQP solver automatically gives up low-priority tasks, i.e. \mathbf{g}_{body} , \mathbf{g}_{pel} , and \mathbf{g}_{swivel} to satisfy the hand pose tasks. Thus the task errors increase.

contrast, for up to 15 s, the HQP solution trajectories have zero angular velocities except for the left arm, which means that the left-arm tasks are perfectly executed without interfering with any lower-level tasks (middle row (b) and (c)).

The right arm solution of RHQP in Figure 4.15 (top middle) indicates that the angular velocities of the elbow (REB) and shoulder roll (RS Roll) joints become zero at $t = 24$ and $t = 33$ s, respectively. We can see the activated joint angle constraints as shown in Figure 4.17. Each joint angle constraint \mathbf{g}_{joint} is activated to satisfy the joint angle range as shown in Figure 4.14. This sudden activation of a constraint results in a velocity discontinuity for all trajectories, which suddenly increases the joint control errors as shown in Figure 4.18. However, a robust PID joint controller can guarantee the stability and keep the control errors sufficiently

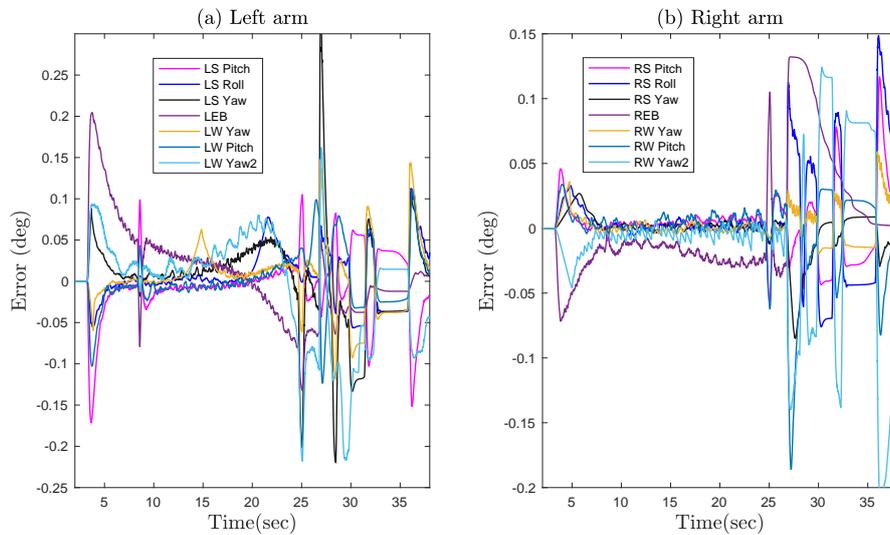


Figure 4.18: Joint space control errors with the RHQP solution: (a) left arm (b) right arm.

small to execute a given reaching task. The velocity discontinuity can be resolved using a smooth task transition method [108, 109, 110]. In summary, the motion tasks cannot violate the physical constraints.

In Figure 4.15 (top right), the velocity of the waist joint increases abruptly at $t = 24$ s. The waist joint is used for positioning the left hand further forward by turning the torso because the left arm is no longer stretched forward on its own. As shown in Figure 4.14 and 4.17 (right), the RHQP solver automatically gives up low-priority tasks, i.e. \mathbf{g}_{body} , \mathbf{g}_{pel} , and \mathbf{g}_{swivel} although these tasks are to maintain initial poses as much as possible. This movement is similar to the way humans try to grab a distant object by sequentially sacrificing less important tasks.

As the SVD-centric approach is leveraged, a wide range of well-known regularization methods and parameter adaptation strategies can be applied to the proposed solver to trade off robustness against performance. According to [62, 63, 65, 68], the regularization parameter is increased in proportion to the inverse of the minimum singular value when the robot configuration is near a singularity. In a singularity free region, the regularization parameter is set to zero, and thus the regularization error can be completely eliminated.

The simulated bimanual manipulation is shown in Figure 4.19. In a bimanual manipulation, since dual arms are physically interacting with the same object, the robot motions are severely restricted by joint operation ranges. Moreover, *internal singularity* can occur inside the reachable workspace. This singularity is more serious than boundary singularity because it occurs in the reachable workspace and it is difficult to notice [104].

In Figure 4.19 (b) we can see internal singularity is encountered by the alignment of two wrist yaw joints when the left forearm is outstretched. As depicted in Figure 4.20, both RHQP and DLS-HQP are singularity robust but DLS-HQP shows cycling. The RHQP algorithm activates the right elbow joint constraint at 5sec due to the restricted operation ranges and then deactivates it at 12 s. However, DLS-HQP cannot remove the right elbow joint constraint around 12 s. Therefore, one more task should be abandoned compared to RHQP in order to satisfy the elbow joint angle constraint.

4.3.4 Inequality Constraint Handling and Computation Time

The main drawback of the clamping method [67] and artificial potential function [20] for treating the inequality constraint is the ambiguity regarding which active constraint should be removed when unnecessary. This problem is addressed by an

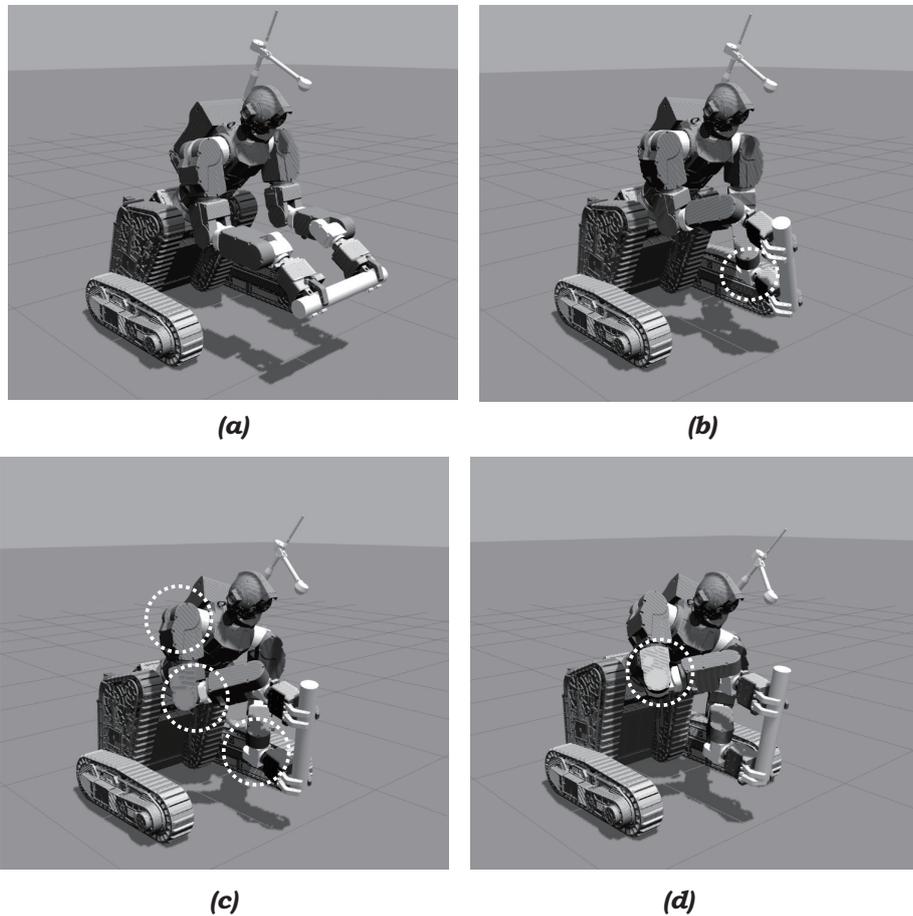


Figure 4.19: Robot poses for bimanual manipulation with the RHQP solver. (a) Initial pose (b) Internal singularity is encountered by the alignment of two wrist yaw joints when the left forearm is outstretched. (c) Right elbow (REB), right shoulder roll (RSR), and left wrist yaw2 (RWY2) joint constraints are activated. (d) Right elbow (REB) joint constraint becomes deactivated.

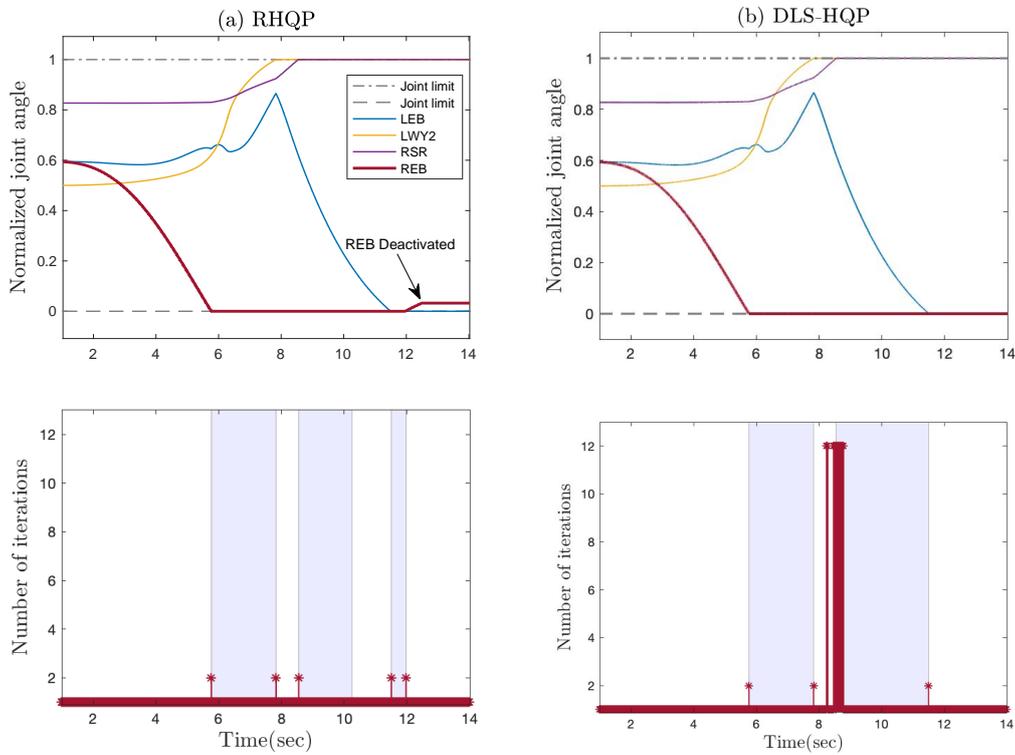


Figure 4.20: Joint angle trajectories (top) and number of iterations (bottom) for bimanual manipulation. (a) Results for RHQP (left). (c) Results for DLS-HQP (right). The RHQP algorithm deactivates the right elbow joint constraint at $t = 12$ s and does not show any cycling. The DLS-HQP solver cannot remove the right elbow constraint at that time and shows cycling with the maximum number of iterations (bottom right). Edges of shaded areas represent the time of activations or deactivations of inequality constraints.

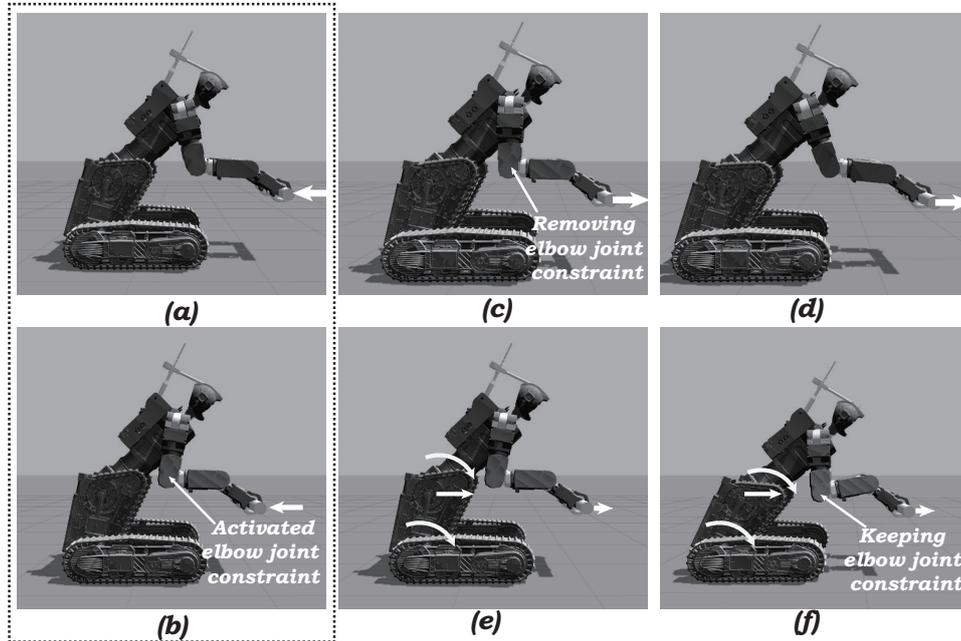


Figure 4.21: Snapshots of bimanual manipulation. The arrows represent the direction of each task. The robot configurations associated with the RHQP solution trajectory: $(a) \rightarrow (b) \rightarrow (c) \rightarrow (d)$ and the DLS-HQP solution trajectory: $(a) \rightarrow (b) \rightarrow (e) \rightarrow (f)$. As the robot pulls its hands toward its body, the elbow joint constraints are activated in (b). The RHQP solver removes the elbow joint constraints as soon as the hand task direction reverses as shown in (c). However, the DLS-HQP solver maintains the elbow joint limit constraint even though the constraint should be removed. Thus, the DLS-HQP gives up the pelvis task by moving the knee and hip joints to perform the hand tasks with priority.

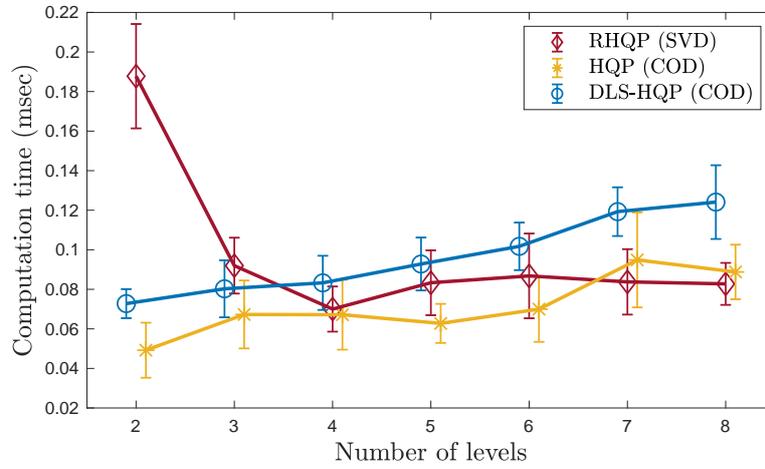


Figure 4.22: Computation time for RHQP, HQP and DLS-HQP with respect to the number of levels. The computation costs of RHQP is nearly equal to the HQP and is cheaper than that of DLS-HQP for $p > 3$.

optimization-based approach, and the RHQP algorithm is also able to properly handle this problem as shown in Figure 4.21. As the robot pulls its hands toward its body, the right and left elbow joint angle constraints are activated (Figure 4.21 (b)). Next, as the robot pushes its hands in the opposite direction, the RHQP solver immediately removes the elbow joint constraints; however, the DLS-HQP cannot deactivate them. Because the DLS-HQP does not involve the regularization error in the Lagrange multipliers for treating inequality constraints. Therefore, the hand task is only achieved by giving up the pelvis task by moving the knee and hip joints since the inequality constraint is not deactivated.

We compare the computation time of RHQP, HQP, and DLS-HQP in the absence of a singularity. The number of level changes from $p = 2$ to $p = 8$ with the same number of DoFs ($n = 19$) and the total task size ($\sum m_k = 19$). The mean

Table 4.2: Computation time (*msec*) for RHQP, HQP and DLS-HQP with respect to the number of levels.

Number of Levels	2	3	4	5	6	7	8
RHQP(mean)	0.1878	0.0920	0.0700	0.0833	0.0868	0.0838	0.0828
RHQP(std)	0.0264	0.0141	0.0114	0.0164	0.0214	0.0166	0.0106
HQP(mean)	0.0492	0.0673	0.0672	0.0628	0.0700	0.0949	0.0888
HQP(std)	0.0139	0.0172	0.0177	0.0099	0.0165	0.0240	0.0138
DLS-HQP(mean)	0.0728	0.0803	0.0833	0.0928	0.1017	0.1193	0.1241
DLS-HQP(std)	0.0073	0.0144	0.0137	0.0134	0.0120	0.0123	0.0186

and standard deviation of the computation time for each algorithm are calculated using 500,000 data points, and the results of which are presented in Figure 4.22 and Table 4.2. These results show that the computation cost of RHQP is nearly equal to HQP and it is cheaper than that of DLS-HQP for $p > 3$. Specifically, when applying the number of levels, $p = 8$, used in all previous results, the matrix sizes $m_k \times o_k$ of each matrix $\mathbf{J}_k \mathbf{Z}_{k-1} \in \mathbb{R}^{m_k \times o_k}$ are 3×19 , 3×16 , 3×13 , 3×10 , 2×7 , 3×5 , and 2×2 according to levels. In this case, interestingly RHQP and HQP have a nearly equal computation time, as shown in Figure 4.22, although it is known that the COD is generally more efficient than SVD. We confirmed that the calculation times of the COD and SVD are nearly equal, if each task size is small, i.e., $m_k \leq 8$ and $o_k \leq 30$, where a typical humanoid robot is applied.

At each level k , DLS-HQP uses one COD to decompose the matrix $\mathbf{J}_k \mathbf{Z}_{k-1}$ in (3.5.124) and computes two inverses of \mathbf{Q}_k^{-1} and $\hat{\mathbf{Q}}_k^{-1}$ necessary to obtain the Lagrange multiplier (3.4.115) and regularized solution (3.5.125), respectively. As a

consequence, DLS-HQP requires a longer computation time than HQP because the DLS-HQP should obtain an additional matrix inverse $\hat{\mathbf{Q}}_k^{-1}$ for each level k . Here, we conducted a forward substitution using the property of the triangular matrix of \mathbf{Q}_k for HQP and DLS-HQP to avoid a matrix inversion of \mathbf{Q}_k . However, we cannot use this technique to acquire $\hat{\mathbf{Q}}_k^{-1}$ because the matrix $\hat{\mathbf{Q}}_k$ is square.

By contrast, RHQP cheaply acquires Σ_k^{-1} and $\hat{\Sigma}_k^{-1}$ by inverting each main diagonal element of Σ_k and $\hat{\Sigma}_k$ respectively. Therefore, the computation time of RHQP is nearly constant regardless of the number of levels for $p > 3$, whereas the cost of DLS-HQP increases as the number of levels increases. However, the smaller the number of levels, such as $p = 2$ or $p = 3$, the closer the Jacobian matrix is to the large square matrix, and the computational complexity of SVD increases significantly. For example, when the number of tasks is $p = 2$, we have the relatively large square matrix $\mathbf{J}_2\mathbf{Z}_1 \in \mathbb{R}^{19 \times 19}$. In this case, the computational cost of the SVD is much higher than that of COD.

In terms of the computational cost, we can see that RHQP is not always disadvantageous compared to HQP if tasks are divided into several hierarchical layers with an appropriate task size m_k for robotic applications. In Figure 4.22, the average computation time of RHQP is within 0.1 *ms*.

4.4 Singularity Robustness and Inequality Handling for Rescue Mission

For a rescue mission, the transfer bed is integrated and the 4 DoF manipulator is equipped as shown in Figure 4.23. The motion tasks are composed of controlling left and right hand positions \mathbf{g}_{hpos} (six DoF) and corresponding pitch angles \mathbf{g}_{hpitch} of the lower arms (two DoF), pelvis traversal position \mathbf{g}_{pel} (single DoF), and gaze

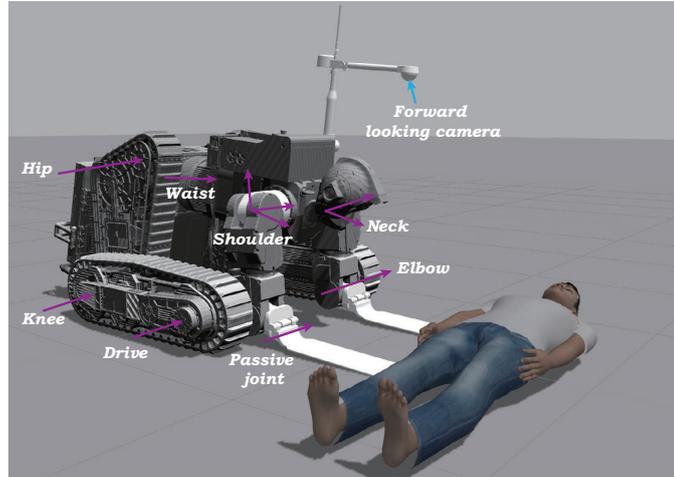


Figure 4.23: Joint configuration of the Hurcules robot for a rescue mission where arrows represent the axis of rotation.

angle \mathbf{g}_{gaze} (two DoF). The motion tasks or motion primitives are shown in Figure 4.24. The purpose of controlling pitch angle is to ensure proper insertion of the lower arms as they enter between the ground and the injured by adjusting the angle. The priority order of five level tasks is then $\mathbf{g}_{joint} \prec \mathbf{g}_{hpos} \prec \mathbf{g}_{hpitch} \prec \mathbf{g}_{pel} \prec \mathbf{g}_{gaze}$, and we use all $\rho_k = 0.1$. The schematic control block diagram for the rescue mission is presented in Figure 2.1.

For the robot to lift the inured, the robot should be able to reach out the lower arms. As can be seen from Figure 4.25 (top left), the battery and transfer bed severely restrict the operation ranges of the hip and knee joints, and the backpack also limits the operation ranges of shoulder joints. When the operator uses a single joystick button to extend the lower arms forward and does not control any other tasks. In that case, the robot postures are presented in Figure 4.25. The joint angle trajectories and control errors with respect to time are shown in Figure 4.26.

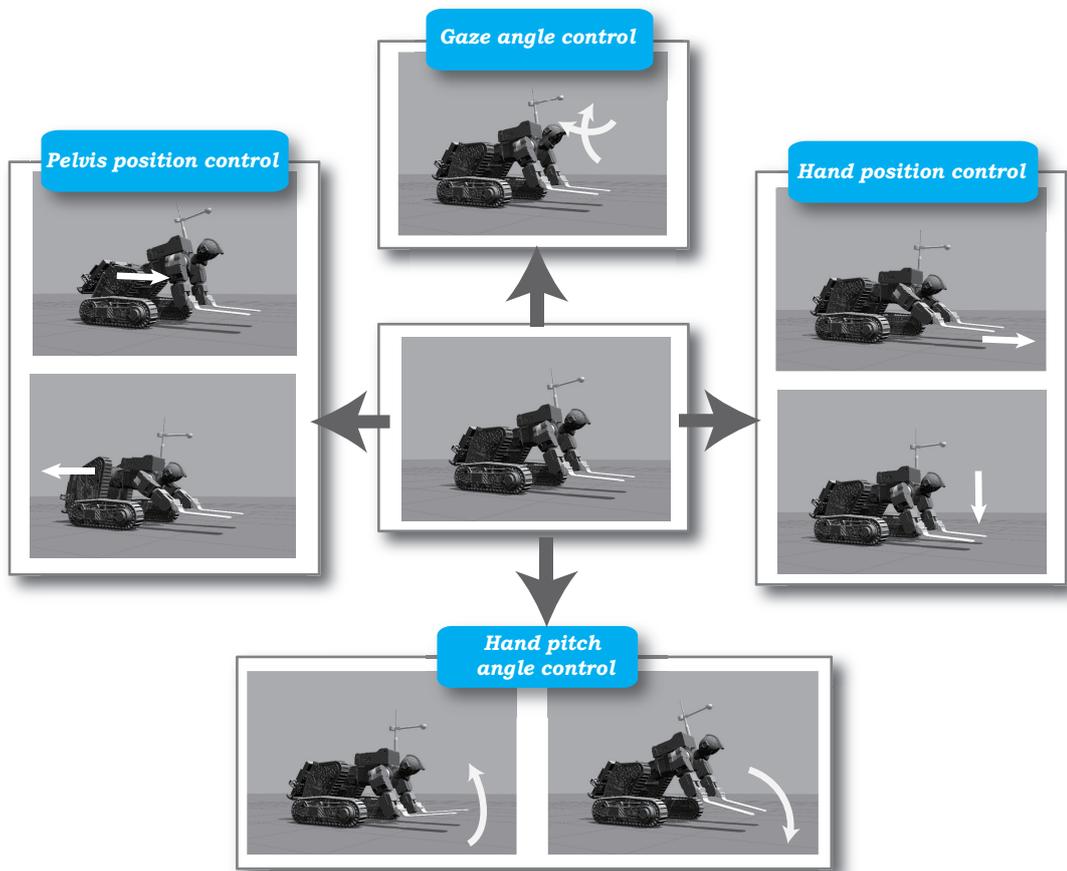


Figure 4.24: Motion tasks for rescue mission. (Left) Pelvis traversal position control (Top) Gaze angle control (Right) Hand position control (Bottom) Hand pitch angle control

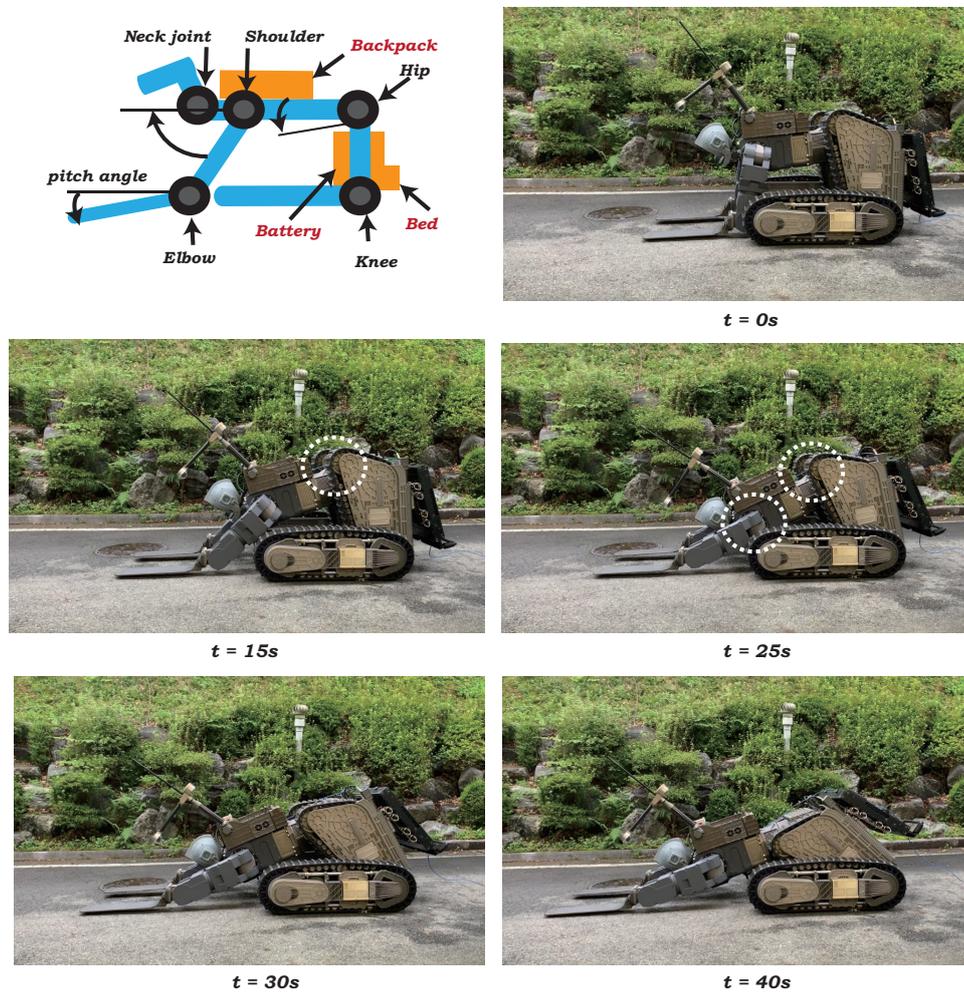


Figure 4.25: Snapshots of stretching motion for the rescue mission. Equipments can severely restrict the joint operation range (top left). The dashed line represents the activated or deactivated joint limit constraint.

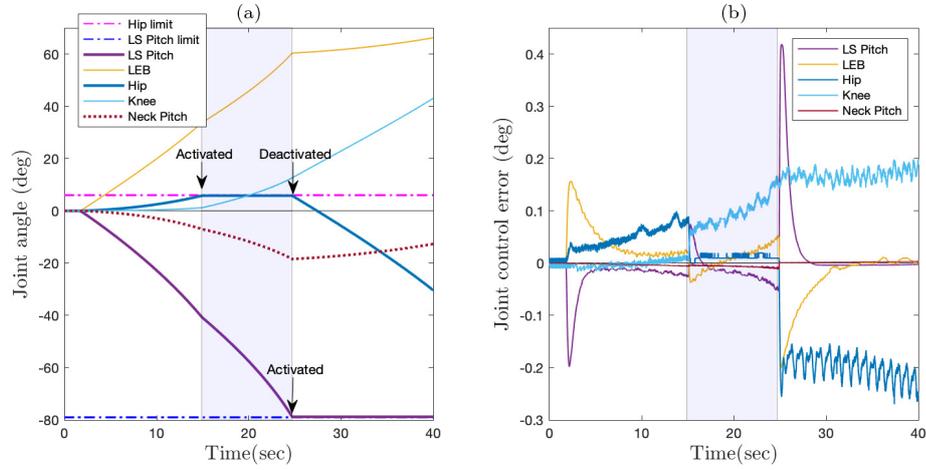


Figure 4.26: Experimental results for RHQP solution: (a) reference joint angular velocities (b) joint space control errors.

Near the time of 15 s, the hip joint angle reaches the limit value established to prevent the battery and torso from colliding. Since the hip joint is locked, the knee joint begins to move in order to satisfy higher level tasks such as \mathbf{g}_{hpos} and \mathbf{g}_{hpitch} by increasing the positional task error of the pelvis task \mathbf{g}_{pel} . At the same time we lose control over \mathbf{g}_{pel} . As the lower arms extend further forward, the shoulder joint reaches its limit value at the time of 25 s, but the hip joint begins to move again since this joint is automatically released from the active constraints due to the knee motion. Here, we confirm that the inequality constraints are well handled by being activated and deactivated at the right time. You can see that the control error increases rapidly when the constraint is suddenly activated or deactivated as illustrated in Figure 4.26 (b). However, the clamping method and artificial potential function have difficulty to decide which active constraint should be deactivated when it is unnecessary.

At all times, the line of sight is maintained to keep the gaze at the injured by adjusting the tilt angle of the neck. At the end of the stretching motion, the pelvis task has the position error of $40cm$ since this task has supported the fulfillment of higher level tasks. We show again that the hierarchical strategy can precisely resolve conflicts by creating a priority in which every task is achieved as much as possible according to the levels.

4.5 Field Tests

Actual field tests for an injured extraction and dangerous object disposal have been conducted at Korean army advanced Combat Training Center (KCTC) and Changwon Proving Ground of Agency for Defense Development. The sequential operations are composed as follows:

- Driving: moving to the destination with the stable tank like posture.
- Turning the waist.
- Folding dual arms to avoid self collision and rotating both lower legs.
- Manipulation ready: standing up and being ready for manipulation.
- Acquisition: task space control for an acquisition of a target object.
- Sitting down, turning the waist, and rotating the right lower leg.
- Turning the waist in the opposite direction and rotating the left lower leg.
- Pointing the torso in the forward direction and being ready to drive.

The injured extraction and explosive ordnance acquisition missions are successfully executed by the sequential pose transformations as shown in Figure 4.27 and 4.28 respectively.

One of the key requirements for the successful execution of the rescue mission is to regulate interaction force between the lower arm and environment because the robot's lower arms should penetrate the gap between the injured and ground as shown in Figure 4.28 (6) and (7). A high value of contact or friction force with the ground prevents the robot to achieve this motion. Moreover, it can stress or break mechanical components or damage electrical parts due to the high current. To make matters worse, as Hurcules adopts position control strategy, the aforementioned situation is more likely to occur. However, a passive compliant element can effectively reduce the excessive contact force. Furthermore, an operator is able to physically feel and monitor the interaction force via the haptic master device and control computer respectively. Because the contact force acting on the lower arms is measured from the angular displacement of the elastic element as shown in Figure 4.8.

In Figure 4.28 (11), the robot automatically expands the transfer bed, and puts the impaired person down to prevent secondary injury to the wounded during transferring.

When driving, the power of dual arms is completely shut down and the posture is maintained by using magnetic breaks in each arm joint. Therefore the robot's continuous operation time is significantly extended. We confirmed that traveling 400m on a round trip, the rescue missions composed of Figure 4.27 and 4.28 are repeatedly performed seven times in two hours without battery replacement.

Although visual images are provided, still it is lacking for an operator to fully recognize the environment and robot status as shown in Figure 4.29. However, the

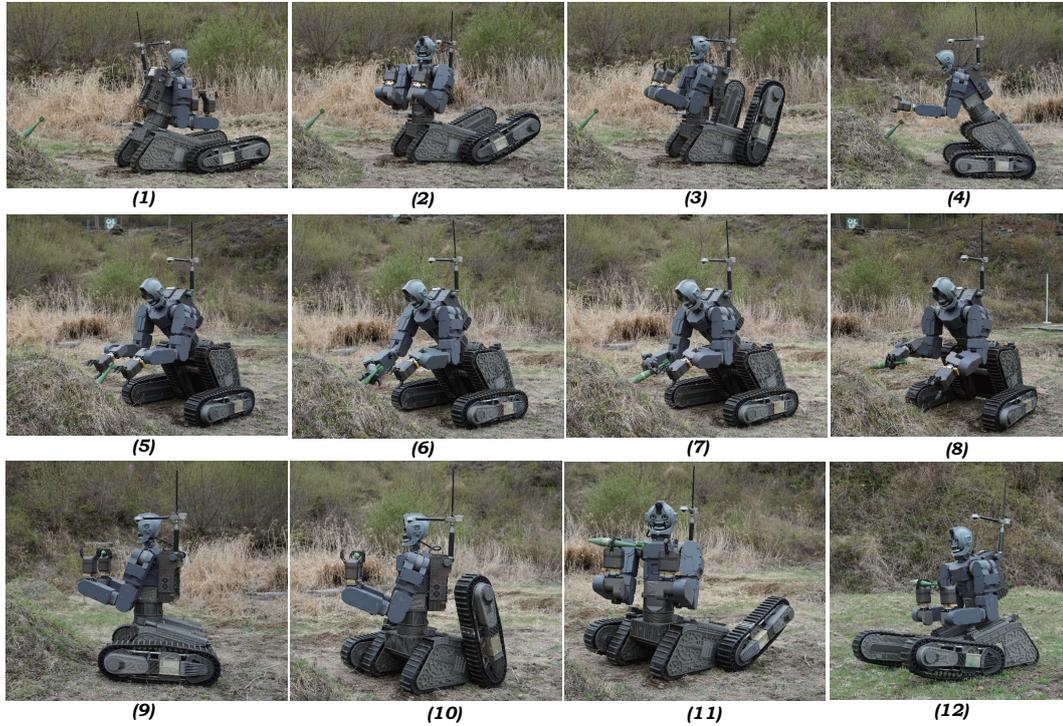


Figure 4.27: Motion sequences of a dangerous object acquisition mission: (1) Driving: moving to the destination (2) turning the waist (3) rotating both lower legs at the same time (4) Manipulation Ready: standing up and being ready for the grasp task (5,6,7,8) Acquisition: task space control via RHQP solver to grasp a dangerous object (9) Lifting and sitting down (10) turning the waist and rotating the right leg (11) turning the waist in the opposite direction and rotating the left lower leg (12) Transfer: pointing the torso in the forward direction and leaning the torso.

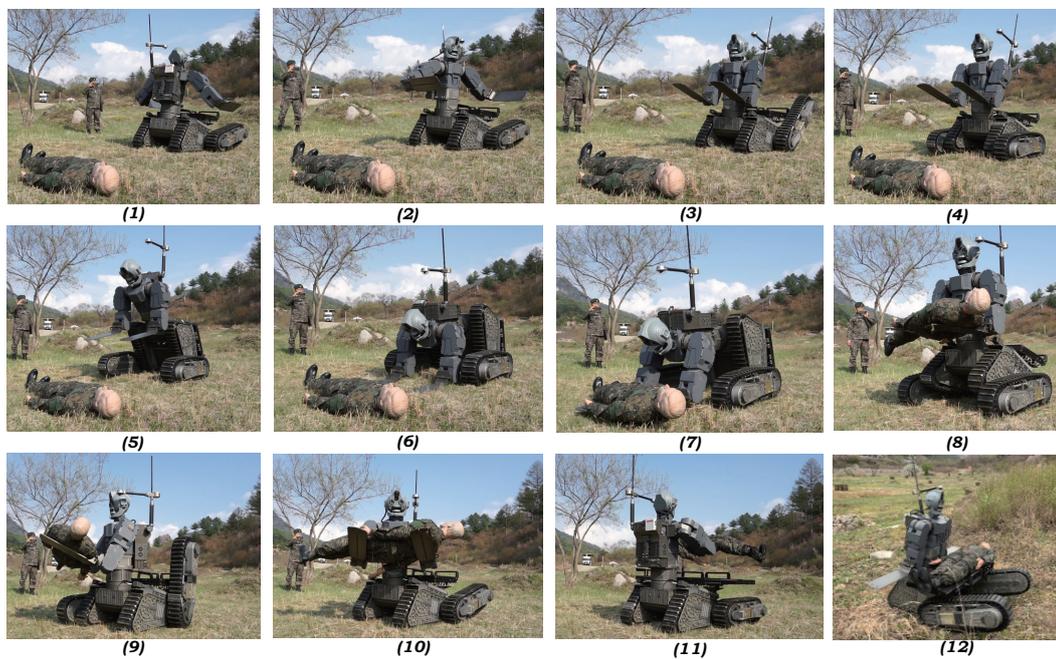


Figure 4.28: Motion sequences of the rescue mission: (1) Driving: moving to the destination (2) turning the waist (3) folding dual arms to avoid self collision and (4) rotating both lower legs at the same time (5) Rescue Ready: standing up and being ready for the rescue task (6,7) Acquisition: task space control via RHQP solver to lift the injured (8) Lifting (9) sitting down, turning the waist, and rotating the right leg (10) turning the waist in the opposite direction, rotating the left lower leg and pointing the torso in the forward direction of driving (11) Putting Down: deploying the transfer bed and putting safely the injured on the bed (12) Transfer: pulling out the arms and being ready to transfer.

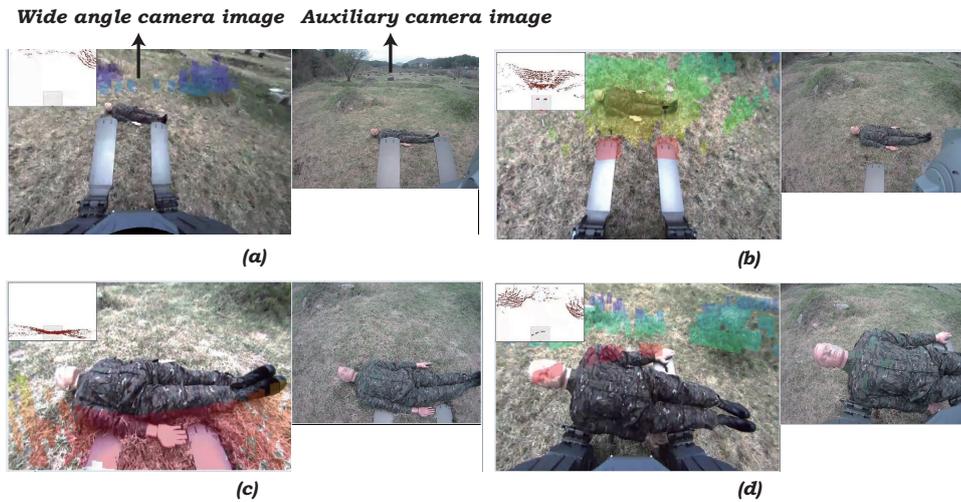


Figure 4.29: Visual information of cameras corresponding to the motion sequences of the rescue mission: (a) Lifting Ready (b,c) Acquisition (d) Lifting.

RHQP motion control algorithm enables the operator to be less concerned with the detailed motion control problems such as task conflict, singularity, and robot's physical constraints.

4.6 Concluding Remarks

We present novel hardware design concepts of the rescue robot, Hurcules and demonstrate the effectiveness of the RHQP algorithm through extensive numerical simulations and experimental tests.

The robot is uniquely designed for mobile manipulation with statically stable platform. The upper body of the robot is designed humanoid in form with replaceable modularized dual arms. The lower body is featured with a hybrid tracked and legged platform to simultaneously acquire versatile manipulability and mobility on

difficult terrain. The robot make use of the stable tank like posture when the versatile mobility is required. As a result, the Hurcules robot can efficiently perform a driving task, dangerous object manipulation, and casualty extraction missions by changing the pose and modularized equipments in an optimized manner.

We demonstrated that the task regularization and the inequality-constrained hierarchical optimization are fully integrated to guarantee the robustness and convergence of the solution in the face of a singularity. The proposed method can also acquire the optimal active set without showing any cycling. The inequality constraints are properly handled by being activated and deactivated at the right time. The RHQP algorithm works fast enough to be used as a practical means of real-time motion control system. The average computation time for a 19-DoF rescue robot is less than 0.1 *ms* on a typical personal computer.

Consequently, the Hurcules robot successfully performs a casualty extraction and explosive ordnance disposal mission in an outdoor environment that none of the existing robots has performed.

5

Hierarchical Motion and Force Control

5.1 Introduction

When the end-effector manipulates an object or executes a work on a surface, the environment imposes constraints on the certain task space direction of geometric path of the end-effector, and a suitable compliant behavior is required. One of the key requirements for the successful execution of the rescue mission is to regulate an interaction force between the lower arm of Hurcules and environment because the lower arms should penetrate the gap between the injured and ground for lifting. However, high value of contact and friction force with the ground prevents the robot to achieve this motion and can stress or break mechanical components or damage electrical parts due to high current. Moreover, as Hurcules adopts the position control strategy in the joint space, the aforementioned situations are more

likely to occur. Otherwise, the lower arms do not contact the ground with appropriate force, and then the injured acquisition mission cannot be accomplished. Because the injured person may be pushed out by the robot's lower arms. Therefore, a direct force control scheme is necessary to precisely regulate the contact force.

While modern hierarchical solvers can handle both motion and force control, the mixed problem of singularity robustness and inequality-constrained hierarchical optimization is not thoroughly investigated [12, 13, 30, 31, 44, 71, 86, 87, 111]. Nearly rank deficient Jacobians coming from kinematic and algorithmic singularities lead to numerical instability, excessive joint accelerations, and large joint torques. These approaches may require that the singularity problem should be solved outside the control loop [104].

In order to address the mixed problem of singularity robustness and inequality-constrained task-priority optimization in a single hierarchical motion and force control loop, the RHQP algorithm is leveraged. While maintaining all of the advantages mentioned in Chapter 3 and 4, the RHQP algorithm is extended to deal with motion and force control. However, instead of concentrating on force control itself, we focus on the integration of RHQP and existing force control laws [104, 112, 113, 114, 115, 116].

We present the resolved acceleration-based motion and force control that can be applied to both torque-controlled and position-controlled robots. The goal of inverse dynamics approach usually used for a torque-controlled robot is the determination of the accelerations or joint torques necessary to generate motions specified in terms of the operational space motion and force [117]. However, it is difficult to use model-based control schemes when a complete dynamic model is unavailable. Usual industrial robots as well as typical field robots are equipped with the joint

space position controllers, and they do not have a joint torque sensor nor direct motor current interfaces.

When the force control is implemented for a position-controlled robot, it is termed to *implicit force control* where the force control loop is closed around the position control loop [112, 118, 119, 120, 121, 122, 123, 124]. The implicit force control schemes exploit a joint space position control loop, which can be the use of PID control or a model-based control laws. The position control loop tends to decouple and linearize the complex robot dynamics.

The behavior of a spring-mass-damper system is obtained by the additional operational space inner position or velocity control loops, and then the force control is conducted according to this dynamic system by closure of the force control loop around the inner loop. As often the force measurement is noisy, it is difficult to directly use a derivative action of the force measurement. Instead, the stabilizing damping effect is obtained through the virtual damper realized by the operational space velocity feedback. Therefore, the force control scheme is realized with a stabilizing PD control action based on the position and velocity measurements as well as force error. If an integral action is applied to the system, zero steady-state force error can be achieved. However, integral action can lead to slow response, wind-up, and reduced stability margin. Thus, instead of resorting to the integral action, we reject steady-state force error by suitably designing the operational space dynamics with the inner position or velocity control loops.

We verify the performance of the RHQP algorithm integrated with motion and force control schemes through numerical simulation results. This control method ensures to obtain a stable optimal solution when facing singularity. The modified

active set method of RHQP correctly activates a blocking constraint and deactivates an unnecessary constraint at the right time. The velocity-based RHQP algorithm is also integrated with the motion and force control, and the effectiveness is demonstrated through experimental tests. As a result, a safe interaction force is guaranteed from the inner position or velocity control loop in conjunction with an outer force feedback controller, and thus the Hercules robot successfully extracts an injured lying on the ground.

5.2 Operational Space Control

The interaction force between robot and environment is naturally described in the operational space. At first, inverse dynamics operational space control is briefly presented. Consider the robot dynamics

$$\mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{c}(\mathbf{q}) = \boldsymbol{\tau} - \mathbf{J}^T(\mathbf{q})\mathbf{h}_e, \quad (5.2.1)$$

where $\mathbf{q} \in \mathbb{R}^n$ is joint angle, $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the symmetric positive definite inertia matrix, $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ denotes Coriolis/centrifugal torque, $\mathbf{c}(\mathbf{q})$ is gravitational torque, $\boldsymbol{\tau}$ is the actuation torque, and \mathbf{h}_e is the vector of contact force.

For multi-level hierarchical control, k level task space dynamics can be written as [20, 85, 117]:

$$\boldsymbol{\Lambda}_k(\mathbf{q})\ddot{\mathbf{s}}_k + \boldsymbol{\nu}_k(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{p}_k(\mathbf{q}) = \mathbf{F}_k - \mathbf{h}_e, \quad (5.2.2)$$

where

$$\Lambda_k(\mathbf{q})^{-1} = \mathbf{J}_k \mathbf{A}_k^{-1} \mathbf{P}_{k-1}^T \mathbf{J}_k^T, \quad (5.2.3)$$

$$\bar{\mathbf{J}}_k^T = \Lambda_k \mathbf{J}_k \mathbf{A}_k^{-1}, \quad (5.2.4)$$

$$\boldsymbol{\nu}_k = \bar{\mathbf{J}}_k^T \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \Lambda_k \dot{\mathbf{J}}_k \dot{\mathbf{q}}, \quad (5.2.5)$$

$$\mathbf{p}_k = \bar{\mathbf{J}}_k^T \mathbf{G}(\mathbf{q}). \quad (5.2.6)$$

The the null space projecting matrix \mathbf{P}_{k-1} in (3.3.42) is used, and the following kinematic relation of the k level is employed

$$\ddot{\mathbf{s}}_k = \mathbf{J}_k \ddot{\mathbf{q}} + \dot{\mathbf{J}}_k \dot{\mathbf{q}}, \quad (5.2.7)$$

where $\ddot{\mathbf{s}}_k$ is the effector's operational space acceleration of the controlled robot. The index k ($1 \leq k \leq p$) refers to the level, where p is the least rank of priority or total number of levels.

We introduce inverse dynamics control law with the force measurement as follows:

$$\mathbf{F}_k = \Lambda_k \mathbf{a}_k + \boldsymbol{\nu}_k(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{p}_k + \mathbf{h}_e, \quad (5.2.8)$$

where \mathbf{a}_k is a new task space control input vector. The nonlinear control law (5.2.8) is denoted by inverse dynamics control because an exact linearization of robot dynamics is realized by a nonlinear state feedback. Substituting the control law (5.2.8) into the task space dynamics (5.2.2) yields

$$\ddot{\mathbf{s}}_k = \mathbf{a}_k, \quad (5.2.9)$$

where \mathbf{a}_k is to be designed for tracking a reference trajectory ${}^d\mathbf{s}_k$ or force command ${}^d\mathbf{f}_k$. The goals of a model-based solution using the inverse dynamics control are linearizing and decoupling the robot dynamics via nonlinear feedback control

(5.2.8). Nonlinear forces acting on the system such as Coriolis/centrifugal torque and gravity force are cancelled by adding the control input, and decoupling is achieved by weighting the control input by the inertia matrix $\mathbf{\Lambda}_k$.

We assume that the linearized and decoupled dynamics holds on for a position-controlled robot, where the position control loop tends to decouple and linearize the complex robot dynamics [118]. The contact force can be compensated for by direct feedback of measured force to the robot actuator or can be neglected. The joint controllers can cancel out the effect of robot dynamics to a large extent while rejecting disturbances acting on the system and make the corresponding joints faithfully follow the reference trajectories. Particularly in the case of slow motion, the behavior of the position-controlled robot can be approximated by [112, 114]

$$\ddot{\mathbf{s}}_k = \mathbf{J}_k \ddot{\mathbf{q}}_k + \dot{\mathbf{J}}_k \dot{\mathbf{q}}_k, \quad (5.2.10)$$

where $\ddot{\mathbf{q}}_k$ is the joint space control signal. The operational space acceleration can be directly controlled by $\ddot{\mathbf{q}}_k$, and this control scheme is termed to *resolved acceleration control* [112].

5.3 Acceleration-Based Hierarchical Motion Control

The task-priority motion control described in Chapter 4 is viewed as first order algorithm, in that it allows the inversion of a task space motion trajectory specified in terms of position and velocities into the equivalent values in the configuration space. This velocity-based resolution has traditionally been preferred in many robotics applications due to the simplicity [12]. Moreover, it is appropriate for the tele-operated velocity control via haptic devices as in our case. Nevertheless, for more smooth motion generation or force control purposes it is necessary to translate a task space motion trajectory specified in terms of position, velocity, and

acceleration into the corresponding configuration space variables. It is termed to *second order algorithm*.

To track the reference command of ${}^d\mathbf{s}_k$, the task function is defined as:

$$\mathbf{g}_k := \mathbf{s}_k - {}^d\mathbf{s}_k \quad (5.3.11)$$

where \mathbf{s}_k is the k level task or pose of an effector. It is a function of the joint variables

$$\mathbf{s}_k = \phi_k(\mathbf{q}), \quad (5.3.12)$$

where ϕ_k a forward mapping from the configuration space to the task space. To bring the task function to zero, the choice of the control law in (5.2.10) is given as follows:

$$\mathbf{J}_k\ddot{\mathbf{q}}_k := -\mathbf{L}_k\dot{\mathbf{g}}_k - \mathbf{K}_k\mathbf{g}_k + {}^d\ddot{\mathbf{s}}_k - \dot{\mathbf{J}}_k\dot{\mathbf{q}}, \quad (5.3.13)$$

where \mathbf{L}_k and \mathbf{K}_k are positive definite gain matrices. The derivative of the task function $\dot{\mathbf{g}}_k$ is

$$\dot{\mathbf{g}}_k = \dot{\mathbf{s}}_k - {}^d\dot{\mathbf{s}}_k, \quad (5.3.14)$$

where $\dot{\mathbf{s}}_k$ is the operational space velocity vector acquired from $\dot{\mathbf{s}}_k = \mathbf{J}_k\dot{\mathbf{q}}$. The position and velocity command, ${}^d\mathbf{s}_k$ and ${}^d\dot{\mathbf{s}}_k$ can be computed by numerically integrating a given ${}^d\ddot{\mathbf{s}}_k$ over time.

Substituting the acceleration control input (5.3.13) into (5.2.10) yields the following operational space dynamics

$$\ddot{\mathbf{g}}_k + \mathbf{L}_k\dot{\mathbf{g}}_k + \mathbf{K}_k\mathbf{g}_k = \mathbf{0}, \quad (5.3.15)$$

which denotes the operational space error dynamics. If k level task is feasible, then the task function (5.3.11) converges to zero along the trajectory (5.3.15), and a

convergence rate depends on the choice of the gain matrices \mathbf{L}_k and \mathbf{K}_k . In other words, once the task space commands ${}^d\mathbf{s}_k$, ${}^d\dot{\mathbf{s}}_k$, and ${}^d\ddot{\mathbf{s}}_k$ are given, then the acceleration solution $\ddot{\mathbf{q}}_k$ of (5.3.13) makes the task function \mathbf{g}_k converge to zero if the corresponding task is feasible.

In order for the acceleration solution to satisfy task hierarchy and guarantee singularity robustness, the RHQP framework (3.4.98) is leveraged by using the following relations:

$$\mathbf{x}_k := \ddot{\mathbf{q}}_k \quad (5.3.16)$$

$$\mathbf{y}_k := -\mathbf{L}_k\dot{\mathbf{g}}_k - \mathbf{K}_k\mathbf{g}_k + {}^d\ddot{\mathbf{s}}_k - \mathbf{J}_k\dot{\mathbf{q}}. \quad (5.3.17)$$

Thus, the resulting regularized hierarchical optimization problem is written as follows:

$$\min_{\ddot{\mathbf{q}}_k, \mathbf{e}_k} \frac{1}{2}\|\mathbf{e}_k\|_2^2 + \frac{1}{2}\rho_k^2\|\mathbf{Y}_k^T\ddot{\mathbf{q}}_k\|_2^2 \quad (5.3.18a)$$

$$\text{subject to } \mathbf{J}_k\ddot{\mathbf{q}}_k - \mathbf{y}_k \leq \mathbf{e}_k \quad (5.3.18b)$$

$$\underline{\mathbf{J}}_{k-1}\ddot{\mathbf{q}}_k - \underline{\mathbf{y}}_{k-1} \leq \underline{\mathbf{e}}_{k-1}^* \quad (5.3.18c)$$

The optimal solution is obtained by using (3.3.37):

$$\ddot{\mathbf{q}}_k^* = \sum_{j=1}^k \left(\mathbf{Y}_j \hat{\Sigma}_j^{-1} \mathbf{U}_j^T (\mathbf{y}_k - \mathbf{J}_j \dot{\mathbf{q}}_{j-1}^*) \right) \quad (5.3.19)$$

$$= \dot{\mathbf{q}}_{k-1}^* + \mathbf{Y}_k \hat{\Sigma}_k^{-1} \mathbf{U}_k^T (\mathbf{y}_k - \mathbf{J}_k \dot{\mathbf{q}}_{k-1}^*), \quad (5.3.20)$$

where

$$\hat{\Sigma}_k^{-1} := (\Sigma_k^2 + \rho_k^2 \mathbf{I}_k)^\dagger \Sigma_k. \quad (5.3.21)$$

The optimal task error is also obtained as:

$$\mathbf{e}_k^* = \mathbf{v}_k^* + \mathbf{u}_k^* \quad (5.3.22)$$

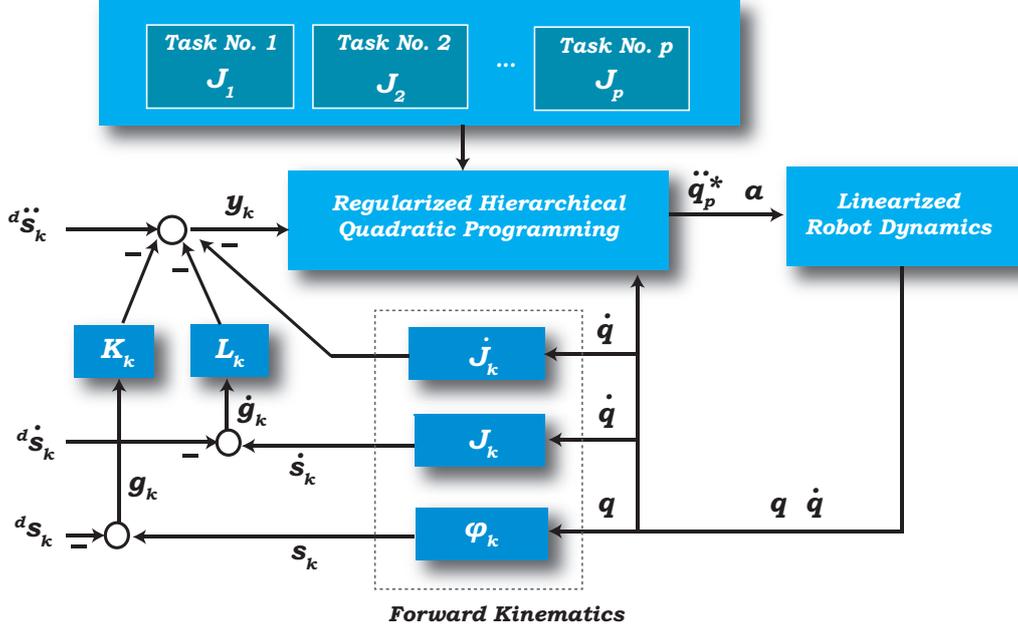


Figure 5.1: Block diagram of operational space motion control with the RHQP solver.

where

$$\mathbf{v}_k^* = \mathbf{V}_k (\mathbf{F}_k \mathbf{Y}_{k-1}^T \ddot{\mathbf{q}}_{k-1}^* - \mathbf{V}_k^T \mathbf{y}_k) \quad (5.3.23)$$

$$= \mathbf{V}_k \mathbf{V}_k^T (\mathbf{J}_k \ddot{\mathbf{q}}_{k-1}^* - \mathbf{y}_k) \quad (5.3.24)$$

$$\mathbf{u}_k^* = \mathbf{U}_k \mathbf{T}_k (\mathbf{E}_k \mathbf{Y}_{k-1}^T \ddot{\mathbf{q}}_{k-1}^* - \mathbf{U}_k^T \mathbf{y}_k) \quad (5.3.25)$$

$$= \mathbf{U}_k \mathbf{T}_k \mathbf{U}_k^T (\mathbf{J}_k \ddot{\mathbf{q}}_{k-1}^* - \mathbf{y}_k). \quad (5.3.26)$$

Similarly hierarchical matrix decomposition, Lagrange multipliers and all other solutions described in Chapter 3 can be obtained.

As a matter of fact, the presence of kinematic and algorithmic singularities influences on the rank of the task Jacobian \mathbf{J}_k , and the effect is difficult to deal with an operational space control [104]. However, the RHQP algorithm allows to seamlessly integrate regularization with inequality-constrained hierarchical optimization in a single loop, and thus the convergence and robustness of the solution are ensured in the face of singularity. The block diagram of operational space motion control (5.3.13) with RHQP (5.3.18) is presented in Figure 5.1. For a position-controlled robot, we assume that the decoupled and linearized dynamics (5.2.10) is satisfied. The p level solutions $\ddot{\mathbf{q}}_p^*$, $\dot{\mathbf{q}}_p^*$, and \mathbf{q}_p^* are directly used as reference trajectories for the joint space position controller. Where p is total number of levels.

5.4 Force Control

Interaction control methodologies can be classified into two groups: indirect force control and direct force control. Indirect force control is working via motion control and does not need a force feedback in principle. There are compliance, impedance, and admittance control [114]. A direct force control is used to regulate the contact force to a desired value via the closure of an outer force feedback loop. Our interest lies on the direct force control because the rescue mission requires to regulate the interaction force between the lower arm and the ground.

Usual industrial robots as well as typical field robots are equipped with the joint space position controllers, and they do not have a joint torque sensor nor direct motor current interfaces. Moreover, the robot joints have transmissions such as harmonic drives, worm, cycloidal, and ball screw, which typically show hard nonlinear characteristics due to friction force. Therefore, it is difficult to obtain an exact dynamic model and hard to design a model-based controller for conventional

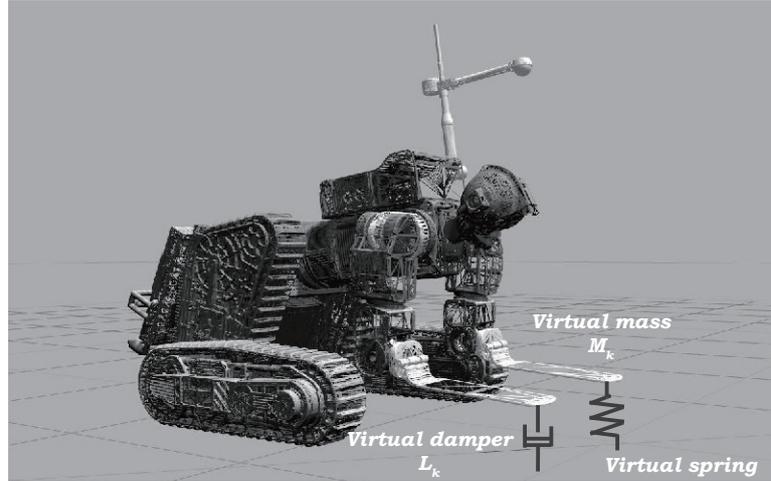


Figure 5.2: Spring mass damper system for force control. The stabilizing damping and spring effects are obtained by the inner operational space position or velocity feedback. Thus, the force control scheme attains a stabilizing PD control action based on velocity and position measurements as well as force error.

industrial robots.

One important requirement of Hurcules is to lift an injury weighing more than 120kg . Thus the robot should attain the ability to cope with large payload variation from 0kg to 120kg without knowing the exact weight of the manipulated object. The joint position controllers are inevitably applied to the Hurcules robot. Therefore, the force control is realized by generating the reference motion command to regulate an interaction force. The joint position controller linearizes the complex robot dynamics and tracks the modified reference by force control to achieve a compliant behavior [112, 118, 119, 120, 121, 122, 123, 124].

The direct force control law is based on the errors between the desired and

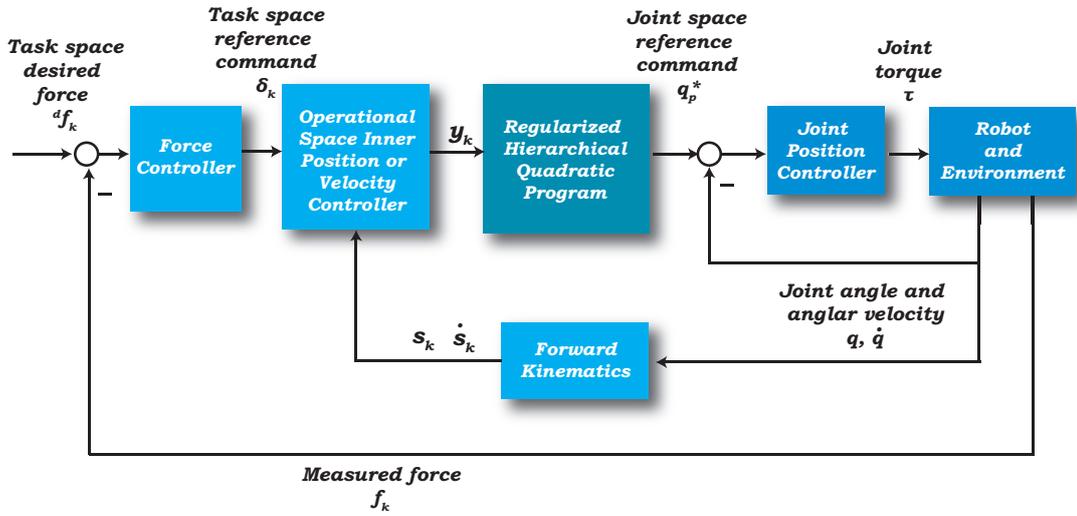


Figure 5.3: Block diagram of force control.

measured contact forces. A spring-mass-damper system is realized by the additional operational space inner position or velocity control laws, and then the force control is performed according to this controlled dynamic system as shown in Figure 5.2. The force measurement is often noisy, and it is difficult to directly employ a derivative action of the force measurement. Instead, the stabilizing damping and spring effects are obtained by the operational space position or velocity feedback. Therefore, the force control scheme requires a stabilizing PD control action based on the position and velocity measurements as well as force error. Moreover, zero steady-state force error is guaranteed by suitably designing the controlled dynamics with the inner control loop. The schematic diagram of force control architecture is depicted in Figure 5.3.

5.4.1 Force Control with Inner Position Loop

Let us assume that the force control space is defined only by position variables and the contact constraint surface is frictionless. For analysis purposes, a point contact is assumed and the generalized spring contact force is used between the end-effector and environment which can be written as

$$\mathbf{f}_k = \mathbf{G}_k (\mathbf{s}_k - \mathbf{r}_k), \quad (5.4.27)$$

where \mathbf{G}_k is a stiffness matrix of the environment, \mathbf{r}_k denotes the environment's rest position, and \mathbf{s}_k is the task space position variable.

In order to obtain a desired operational space dynamics from (5.2.10), the acceleration input $\ddot{\mathbf{q}}_k$ is determined by

$$\mathbf{J}_k \ddot{\mathbf{q}}_k = \mathbf{M}_k^{-1} \left(-\mathbf{L}_k \dot{\mathbf{s}}_k + \mathbf{K}_k (\boldsymbol{\delta}_k - \mathbf{s}_k) - \mathbf{M}_k \dot{\mathbf{J}}_k \dot{\mathbf{q}} \right), \quad (5.4.28)$$

where $\boldsymbol{\delta}_k$ is a reference task space position command to be determined which is related to a force control error. The gains $\mathbf{M}_k \in \mathbb{R}^{m_k \times m_k}$, $\mathbf{L}_k \in \mathbb{R}^{m_k \times m_k}$ and $\mathbf{K}_k \in \mathbb{R}^{m_k \times m_k}$ are positive definite and usually diagonal matrices. The gain matrices \mathbf{M}_k , \mathbf{L}_k and \mathbf{K}_k can be interpreted as the desired mass, damping, and stiffness respectively, and they can be adjusted to suitably determine the dynamic characteristic along the operational space directions.

The vector $\boldsymbol{\delta}_k$ is calculated by the force controller as follows:

$$\boldsymbol{\delta}_k = \mathbf{C}_k ({}^d \mathbf{f}_k - \mathbf{f}_k), \quad (5.4.29)$$

where ${}^d \mathbf{f}_k$ is the desired constant force, \mathbf{f}_k is the measured force of (5.4.27), and \mathbf{C}_k is the diagonal matrix representing compliance. The force controller (5.4.29) has a form of admittance control that takes as an input, the error between a desired force ${}^d \mathbf{f}_k$ and the end-effector's sensed force \mathbf{f}_k , and then it converts the force

error into the position command with the compliance matrix \mathbf{C}_k . Consequently this admittance controller *admits* a certain amount of motion $\boldsymbol{\delta}_k$. The force control algorithm described in (5.4.28) and (5.4.29) is presented in Figure 5.4.

Substituting the force control law (5.4.28) into the linearized and decoupled dynamics (5.2.10) leads to the following operational space dynamics

$$\mathbf{M}_k \ddot{\mathbf{s}}_k + \mathbf{L}_k \dot{\mathbf{s}}_k + \mathbf{K}_k \mathbf{s}_k = \mathbf{K}_k \boldsymbol{\delta}_k. \quad (5.4.30)$$

Note that the control input $\ddot{\mathbf{q}}_k$ in (5.4.28) takes the position \mathbf{s}_k to $\boldsymbol{\delta}_k$. This force control method is termed to the *force control with inner position loop* [104, 115]. With the elastically compliant environment (5.4.27) and the force control law (5.4.29), the operational space dynamics (5.4.30) is rewritten as follows:

$$\mathbf{M}_k \ddot{\mathbf{s}}_k + \mathbf{L}_k \dot{\mathbf{s}}_k + \mathbf{K}_k (\mathbf{I}_k + \mathbf{C}_k \mathbf{G}_k) \mathbf{s}_k = \mathbf{K}_k \mathbf{C}_k ({}^d \mathbf{f}_k + \mathbf{G}_k \mathbf{r}_k), \quad (5.4.31)$$

where $\mathbf{I}_k \in \mathbb{R}^{m_k \times m_k}$ is the identity matrix. This second order system is stable because the matrices \mathbf{M}_k , \mathbf{L}_k , and $\mathbf{K}_k (\mathbf{I}_k + \mathbf{C}_k \mathbf{G}_k)$ are positive definite. If an equilibrium point is reached ($\ddot{\mathbf{s}}_k = \mathbf{0}$, $\dot{\mathbf{s}}_k = \mathbf{0}$), then the point is

$$\mathbf{s}_k = (\mathbf{I}_k + \mathbf{C}_k \mathbf{G}_k)^{-1} \mathbf{C}_k ({}^d \mathbf{f}_k + \mathbf{G}_k \mathbf{r}_k). \quad (5.4.32)$$

The equilibrium point leads to the steady state force in (5.4.27) as follows:

$$\mathbf{f}_k = \mathbf{G}_k \left((\mathbf{I}_k + \mathbf{C}_k \mathbf{G}_k)^{-1} \mathbf{C}_k ({}^d \mathbf{f}_k + \mathbf{G}_k \mathbf{r}_k) - \mathbf{r}_k \right). \quad (5.4.33)$$

Therefore, it is difficult to make \mathbf{f}_k equal to ${}^d \mathbf{f}_k$ because \mathbf{C}_k has pure proportional control action.

An integral control is additionally applied to the force error

$$\boldsymbol{\delta}_k = \mathbf{C}_k ({}^d \mathbf{f}_k - \mathbf{f}_k) + \mathbf{N}_k \int^t ({}^d \mathbf{f}_k - \mathbf{f}_k) d\zeta, \quad (5.4.34)$$

where \mathbf{N}_k is the integral control gain matrix. The resulting dynamics coming from (5.4.30) and (5.4.34) becomes third order, and $\mathbf{f}_k = {}^d\mathbf{f}_k$ can be achieved due to the integral control. Nevertheless, the integral action can guarantee zero steady state force error, it makes the system slow and can reduce the stability margin due to the characteristic of the third order system [125]. Moreover, it may bring about integral induced wind-up. We are going to reject steady-state force error by suitably designing the operational space controlled dynamics.

5.4.2 Force Control with Inner Velocity Loop

Now we focus our attention on accomplishing zero steady state force error. Instead of using the inner position loop in (5.4.28), if we choose the following control law

$$\mathbf{J}_k \ddot{\mathbf{q}}_k = \mathbf{M}_k^{-1} \left(-\mathbf{L}_k \dot{\mathbf{s}}_k + \mathbf{K}_k \boldsymbol{\delta}_k - \mathbf{M}_k \dot{\mathbf{J}}_k \dot{\mathbf{q}} \right), \quad (5.4.35)$$

and then we can obtain the resulting dynamics from the task space dynamics of (5.2.10)

$$\mathbf{M}_k \ddot{\mathbf{s}}_k + \mathbf{L}_k \dot{\mathbf{s}}_k = \mathbf{K}_k \boldsymbol{\delta}_k, \quad (5.4.36)$$

where

$$\boldsymbol{\delta}_k = \mathbf{C}_k ({}^d\mathbf{f}_k - \mathbf{f}_k). \quad (5.4.37)$$

This scheme is called the *force control with inner velocity loop* [104, 115]. In this case, the force error is converted into the velocity command with the compliance matrix \mathbf{C}_k .

The controlled dynamics (5.4.36) can be rewritten by using the elastic model (5.4.27) and the force control law (5.4.37) as follows:

$$\mathbf{M}_k \ddot{\mathbf{s}}_k + \mathbf{L}_k \dot{\mathbf{s}}_k + \mathbf{K}_k \mathbf{C}_k \mathbf{G}_k \mathbf{s}_k = \mathbf{K}_k \mathbf{C}_k \left({}^d\mathbf{f}_k + \mathbf{G}_k \mathbf{r}_k \right). \quad (5.4.38)$$

A stable equilibrium point is reached ($\ddot{\mathbf{s}}_k = \mathbf{0}$, $\dot{\mathbf{s}}_k = \mathbf{0}$), because \mathbf{M}_k , \mathbf{L}_k , and $\mathbf{K}_k \mathbf{C}_k \mathbf{G}_k$ are positive definite matrices. The equilibrium point is

$$\mathbf{s}_k = \mathbf{r}_k + \mathbf{G}_k^{-1} {}^d \mathbf{f}_k. \quad (5.4.39)$$

Substituting the point (5.4.39) into the elastic environment model (5.4.27) yields the measured force as follows:

$$\mathbf{f}_k = {}^d \mathbf{f}_k. \quad (5.4.40)$$

Thus we can obtain $\mathbf{f}_k = {}^d \mathbf{f}_k$ at steady state. This force control scheme in (5.4.35) is shown in Figure 5.4, and it can be also derived from the force control with inner position loop by deleting the position feedback loop (dashed line in Figure 5.4).

5.5 Motion and Force Control

If the motion and force control have different priority, then the task priority strategy allows that every task is achieved as much as possible according to the levels. The lower level tasks do not interfere with the fulfillment of the higher priority tasks. However, as the environment imposes constraints on the end-effector, motion and force control references should be chosen to be compatible with environment features. The hierarchical motion and force control is naturally extended to hybrid motion/force control that provides position and force control with two separate control loops.

The hybrid motion/force control was introduced in [126] and the explicit inclusion of the robot dynamic model is described in [86]. This control problem is a complex, because the robot joints should contribute to both unconstrained motion directions and force directions along the constrained task. For many robotic

systems, an orthogonal reference frame can be specified in terms of natural and artificial constraints of the task or constraint frame.

The natural constraint does not admit translation along a direction or rotation about an axis and the application of force or moment. While the artificial constraint enables us to assign arbitrary reference motion and force values which is not subject to natural constraints. Two sets of constraints are complementary, and selection matrices applied to both the desired and feedback quantities allow to achieve motion and force control for planar contact surfaces [104, 113].

The motion and force control laws integrated with RHQP can be written as

$$\min_{\ddot{\mathbf{q}}_k, \mathbf{e}_k} \quad \frac{1}{2} \|\mathbf{e}_k\|_2^2 + \frac{1}{2} \rho_k^2 \|\mathbf{Y}_k^T \ddot{\mathbf{q}}_k\|_2^2 \quad (5.5.41a)$$

$$\text{subject to} \quad \mathbf{J}_k \ddot{\mathbf{q}}_k - \mathbf{y}_k \leq \mathbf{e}_k \quad (5.5.41b)$$

$$\underline{\mathbf{J}}_{k-1} \ddot{\mathbf{q}}_k - \underline{\mathbf{y}}_{k-1} \leq \underline{\mathbf{e}}_{k-1}^*, \quad (5.5.41c)$$

where the motion control task shown in (5.3.17) is

$$\mathbf{y}_k := -\mathbf{L}_k \dot{\mathbf{g}}_k - \mathbf{K}_k \mathbf{g}_k + {}^d \ddot{\mathbf{s}}_k - \dot{\mathbf{J}}_k \dot{\mathbf{q}}. \quad (5.5.42)$$

The force control tasks are as follows:

- Proportional force control with inner position loop described in (5.4.28) and (5.4.29) is:

$$\mathbf{y}_k := \mathbf{M}_k^{-1} \left(-\mathbf{L}_k \dot{\mathbf{s}}_k + \mathbf{K}_k (\boldsymbol{\delta}_k - \mathbf{s}_k) - \mathbf{M}_k \dot{\mathbf{J}}_k \dot{\mathbf{q}} \right), \quad (5.5.43)$$

where

$$\boldsymbol{\delta}_k = \mathbf{C}_k ({}^d \mathbf{f}_k - \mathbf{f}_k). \quad (5.5.44)$$

- Proportional-integral force control with inner position loop shown in (5.4.28) and (5.4.34) is:

$$\mathbf{y}_k := \mathbf{M}_k^{-1} \left(-\mathbf{L}_k \dot{\mathbf{s}}_k + \mathbf{K}_k (\boldsymbol{\delta}_k - \mathbf{s}_k) - \mathbf{M}_k \dot{\mathbf{J}}_k \dot{\mathbf{q}} \right), \quad (5.5.45)$$

where

$$\boldsymbol{\delta}_k = \mathbf{C}_k ({}^d \mathbf{f}_k - \mathbf{f}_k) + \mathbf{N}_k \int^t ({}^d \mathbf{f}_k - \mathbf{f}_k) d\zeta. \quad (5.5.46)$$

- Proportional force control with inner velocity loop presented in (5.4.35) and (5.4.37) is:

$$\mathbf{y}_k := \mathbf{M}_k^{-1} \left(-\mathbf{L}_k \dot{\mathbf{s}}_k + \mathbf{K}_k \boldsymbol{\delta}_k - \mathbf{M}_k \dot{\mathbf{J}}_k \dot{\mathbf{q}} \right), \quad (5.5.47)$$

where

$$\boldsymbol{\delta}_k = \mathbf{C}_k ({}^d \mathbf{f}_k - \mathbf{f}_k). \quad (5.5.48)$$

The block diagram of the k level motion and force control framework is shown in Figure 5.5, where the motion and force control references are chosen to be compatible with environment features.

The p level optimal acceleration solution of RHQP in (5.5.41) is expressed as $\ddot{\mathbf{q}}_p^*$, and reference joint angles and angular velocities are obtained by integrating the solution $\ddot{\mathbf{q}}_p^*$ and $\dot{\mathbf{q}}_p^*$ over time. Where p is total number of levels. They are used for the joint space position controller as commands. As Hurcules adopts the inverse optimal PID controllers, these references are given as follows:

$$\mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{c}(\mathbf{q}) + \mathbf{d} = -\left(\mathbf{K} + \frac{1}{\gamma^2} \mathbf{I} \right) \left(\dot{\boldsymbol{\epsilon}} + \mathbf{K}_P \boldsymbol{\epsilon} + \mathbf{K}_I \int \boldsymbol{\epsilon} \right), \quad (5.5.49)$$

where $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the the symmetric positive definite inertia matrix, and $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ denotes the nonlinear forces including Coriolis and centrifugal torque, $\mathbf{c}(\mathbf{q})$

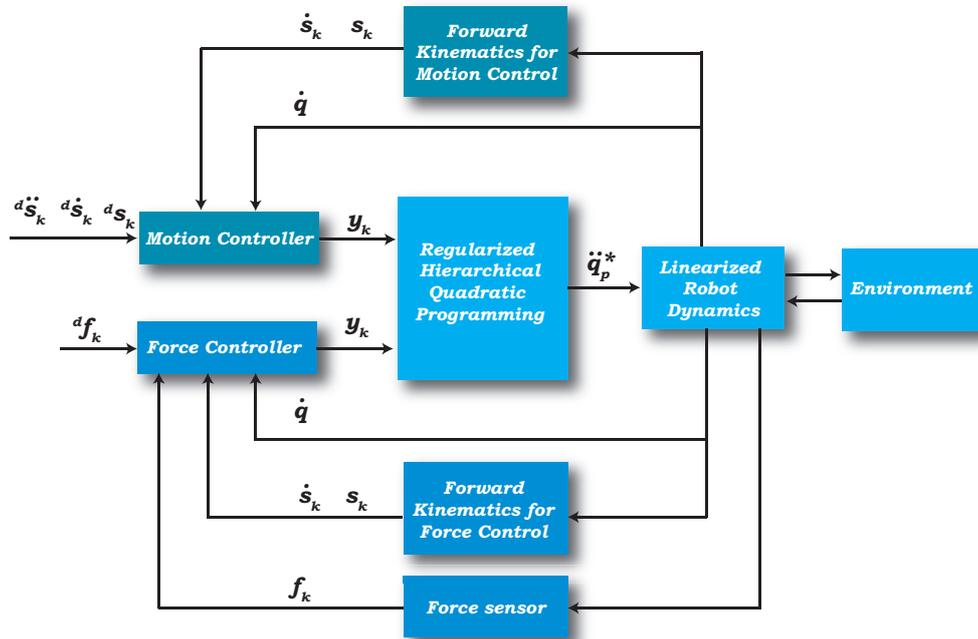


Figure 5.5: Block diagram of acceleration-based motion and force control

is gravitational torque. The vector $\epsilon = q_p^* - q$ is a control error and d is an unknown bounded disturbance input. The matrices \mathbf{K} , \mathbf{K}_p and \mathbf{K}_I are constant diagonal matrices and γ is a scalar \mathcal{L}_2 gain. Please see the Appendix for a detailed explanation.

5.6 Numerical Results for Acceleration-Based Motion and Force Control

We verify the performance of the acceleration-based RHQP algorithm integrated with motion and force control schemes.

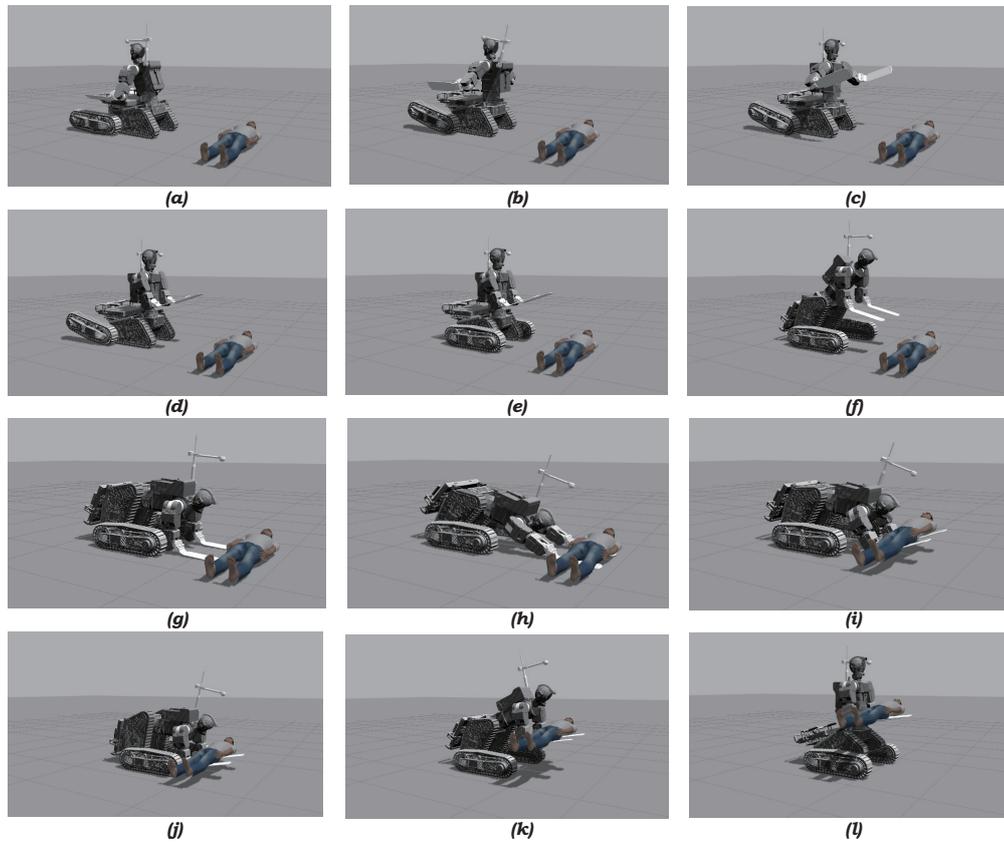


Figure 5.6: Injured acquisition movement sequences. (a) Driving: arriving at the destination (b) open the arms to avoid self-collision (c) turning the waist (d) folding dual arms and (e) rotating both lower legs at the same time (f) Rescue Ready: standing up and being ready for the rescue task (g) Lowering: The lowering motion phase is executed under acceleration-based motion control at first. If the contact force exceeds a specified threshold, the force control begins to work. (h) Acquisition: task space control via motion and force control for the acquisition of an injured lying on the ground (i) Confirm: To move out of contact is executed by only motion control scheme. (j) Embracing (k-l) Lifting.

5.6.1 Task Specifications

The motion sequences for an injured acquisition mission are presented in Figure 5.6, where the pose transformation is used to obtain enhanced manipulability by using the knee, hip, and waist joints of the mobile platform. An interaction force regulation is required in the lowering and acquisition motions as shown in Figure 5.6 (g) and (h). The robot's lower arms should contact with an appropriate magnitude of force to penetrate the gap between the injured and ground for lifting. For successful rescue mission, singularity robustness and the physical constraints satisfaction should be ensured.

The main tasks are composed of controlling left and right hand positions \mathbf{g}_{hpos} , interaction forces \mathbf{g}_{force} , pitch angles \mathbf{g}_{hpitch} of the lower arms, pelvis traversal position \mathbf{g}_{pel} , and gaze angle \mathbf{g}_{gaze} as shown in Figure 4.24. The priority order is specified as $\mathbf{g}_{joint} \prec \mathbf{g}_{hpos}$, $\mathbf{g}_{force} \prec \mathbf{g}_{hpitch} \prec \mathbf{g}_{pel} \prec \mathbf{g}_{gaze}$, and $\rho_k = 0.1$ is used for all task regularization coefficients. We impose the top-most priority on the physical constraints of the joint angle ranges \mathbf{g}_{joint} to eliminate the possibility that the motion tasks violate the physical constraints. The task \mathbf{g}_{force} denotes force control to regulate the contact force between the lower arms and the ground. The operational space command \mathbf{y}_k based on sensed interaction forces is provided to the RHQP algorithm for fast transcription of the objectives into the configuration space motion as shown in Figure 5.5.

The joint angle ranges are expressed by inequality constraints with the highest priority. If the joint angle limits are not explicitly included in the optimization, then the solution is not physically applicable to the real robot. Since the motions of real robots as well as the Hircules are severely restricted by each joint operation range due to collision of mechanical structure, the stiffness of internally routed

cables, and additional equipments such as battery, backpack, and transfer bed as shown in Figure 4.25. Note that the battery and transfer bed severely restrict the operation ranges of the hip and knee joints, and the backpack limits the operation ranges of shoulder joints.

The joint angle constraints \mathbf{g}_{joint} can be specified in terms of accelerations [127]. Each joint has an admissible maximum angle q_u and minimum value q_l , and the constraints are expressed by two inequality as $q \leq q_u$ and $-q \leq -q_l$ where $q := q(t_k)$ is the joint angle at the time t_k . Then the angle constraint is represented as:

$$q_l \leq q + \dot{q}\Delta t + \frac{1}{2}\ddot{q}\Delta t^2 \leq q_u, \quad (5.6.50)$$

where Δt is integration interval. Thus each constraint is described by the acceleration

$$\ddot{q}_l \leq \ddot{q} \leq \ddot{q}_u \quad (5.6.51)$$

where

$$\ddot{q}_l = \frac{2}{\Delta t^2} (q_l - q - \dot{q}\Delta t) \quad (5.6.52)$$

$$\ddot{q}_u = \frac{2}{\Delta t^2} (q_u - q - \dot{q}\Delta t). \quad (5.6.53)$$

The inequality constraint (5.6.51) can be written as the form of $\mathbf{J}_k \ddot{\mathbf{q}}_k \leq \mathbf{y}_k$ where $\mathbf{J}_k \in \mathbb{R}^{2n \times 2n}$ represents all joints having joint angle limit constraints, and \mathbf{y}_k is the augmented vectors of \ddot{q}_l and \ddot{q}_u .

5.6.2 Force Control Performance

The Hurcules robot is equipped with a spring element near the elbow joints as shown in Figure 4.8, but the numerical simulation models do not consider this passive compliance element. Therefore, the simulation condition is more severe than

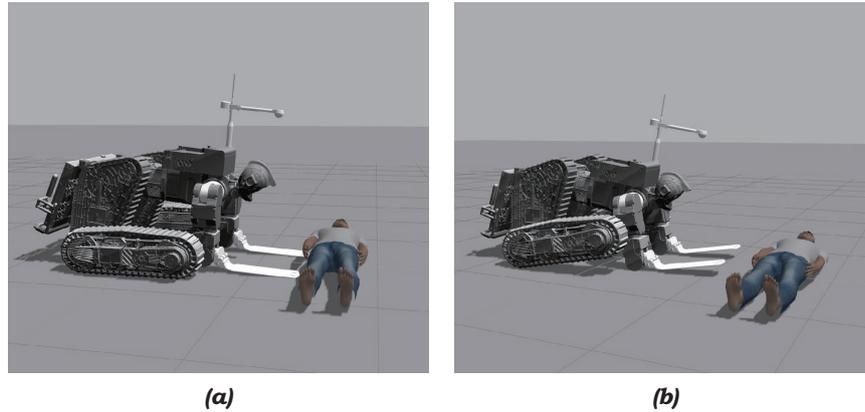
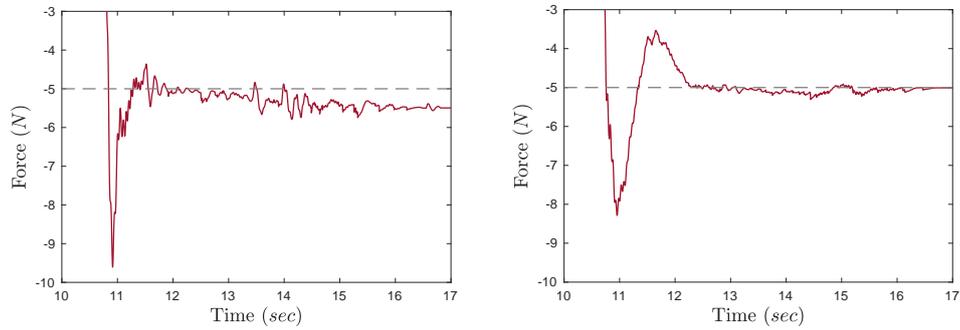


Figure 5.7: Lowering motions: (a) with force control laws (b) without force control laws.

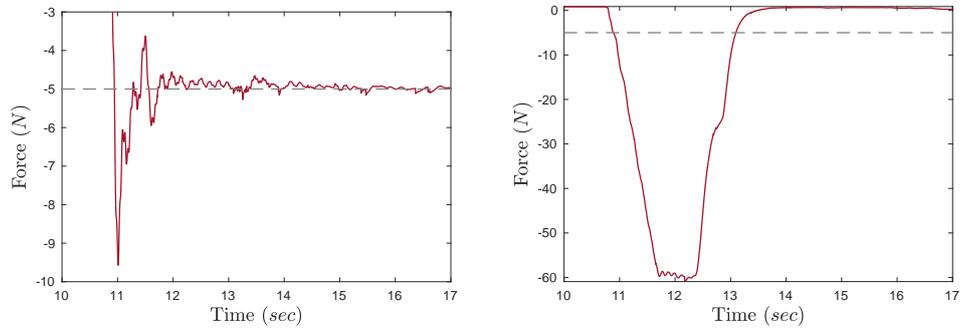
the actual experimental condition because the force controller can result in instability when interacting with the stiff environment.

When lowering motion is performed, the numerical simulation results are depicted in Figure 5.7, 5.8 and 5.9, in terms of the robot poses, measured force, and solution trajectories respectively. As a distal human operator is difficult to obtain accurate information of the robot and its environment, the planning and control errors may lead the robot pose with the high contact force as shown in Figure 5.7 (b) and 5.8 (d). The reference trajectories do not change as depicted in Figure 5.9 (b).

If the force control laws are applied, the interaction force can be regulated around the desired value. As can be seen in Figure 5.8, the measured force \mathbf{f}_k of the proportional force control with inner position loop cannot reach the desired force ${}^d\mathbf{f}_k$ at steady state. Because the force controller has pure proportional control action with \mathbf{C}_k . The proportional-integral force control with inner position loop can reject the steady state force error, however the response is slow due

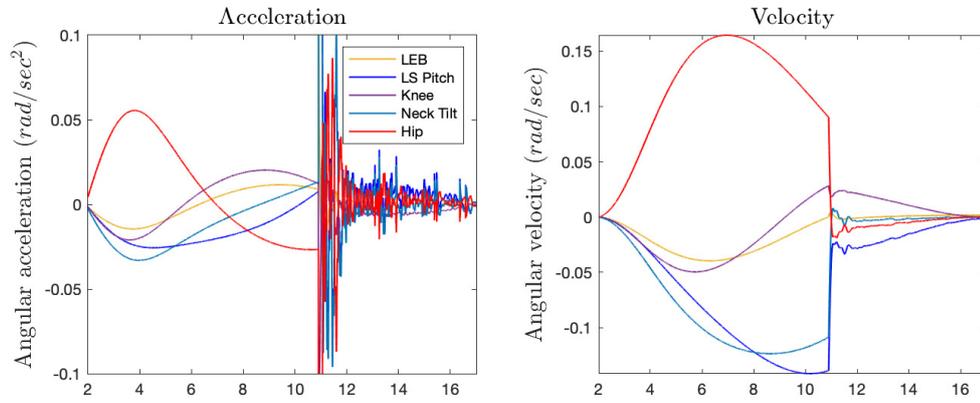


(a) Force control with inner position loop (b) Force control with inner position loop and integral action

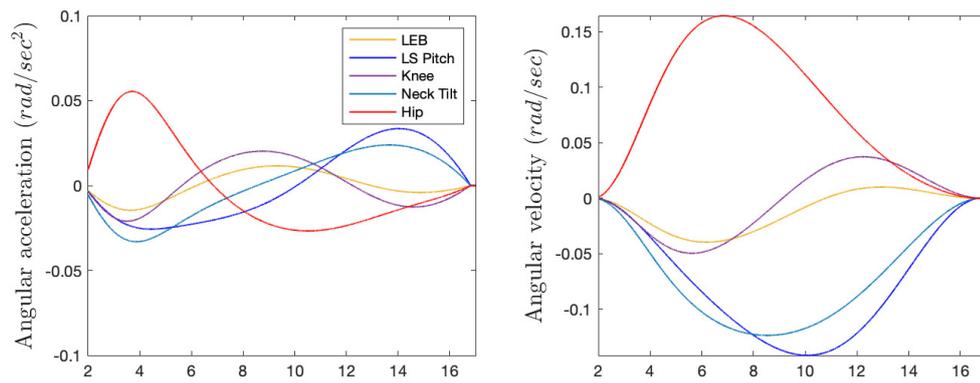


(c) Force control with inner velocity loop (d) Without force control

Figure 5.8: Time histories of the measured force when three force control laws are applied. (a) Measured force for the proportional force control with inner position loop cannot reach the desired force at steady state because C_k is pure proportional control action. (b) The proportional-integral force control with inner position loop can reject the steady state force error, however the response is slow due to the third order system characteristics with the reduced the stability margin. (c) The proportional force control with inner velocity loop shows that the desired contact force $-5N$ is achieved. (d) Excessive contact force arises when only motion control is applied.



(a) Force control with inner velocity loop



(b) Without force control

Figure 5.9: Lowering motions with and without force control laws. Two solution trajectories (angular acceleration, angular velocity) with force control law (a) and without force control law (b).

to the third order system dynamics. Although the integral action guarantees zero steady-state force error, it can lead to stability problems due to the reduced stability margin and integral induced wind-up.

In Figure 5.8 (c), the time history of the measured force for force control with inner velocity loop shows that the contact force is regulated around the desired force $-5N$. It should be noticed that there is no significant force error at steady state, and it verifies that the controlled dynamics stated in (5.4.38) is valid with the equilibrium point (5.4.39). Thus we can obtain $\mathbf{f}_k = {}^d\mathbf{f}_k$ at steady state as predicted in (5.4.40). The reference joint trajectories are modified by the force controller (5.5.47) to regulate the interaction force which are reported in Figure 5.9 (a).

5.6.3 Singularity Robustness and Inequality Constraint Handling

After the lower arms contact with the ground, the robot should acquire an injury as depicted in Figure 5.6 (third row). Now the force control with inner velocity loop is employed because the desired contact force is achieved with the second order operational space dynamics as demonstrated in the previous Section 5.6.2.

Let us consider that both hands of the robot reach out to extract a person and other tasks are to maintain the initial poses as close as possible. In this case, a singularity is encountered as shown in Figure 5.10. Because the arms are outstretched, the mobility is reduced, and the corresponding Jacobian becomes rank deficient. Numerical acceleration and velocity solutions are presented in Figure 5.11. From $t = 12$ s, the angular accelerations and velocities begin to decrease gradually because the regularization in (3.3.39) has a strong impact on the solution as the singular value decreases as shown in Figure 3.2. Thus singularity robustness is ensured.

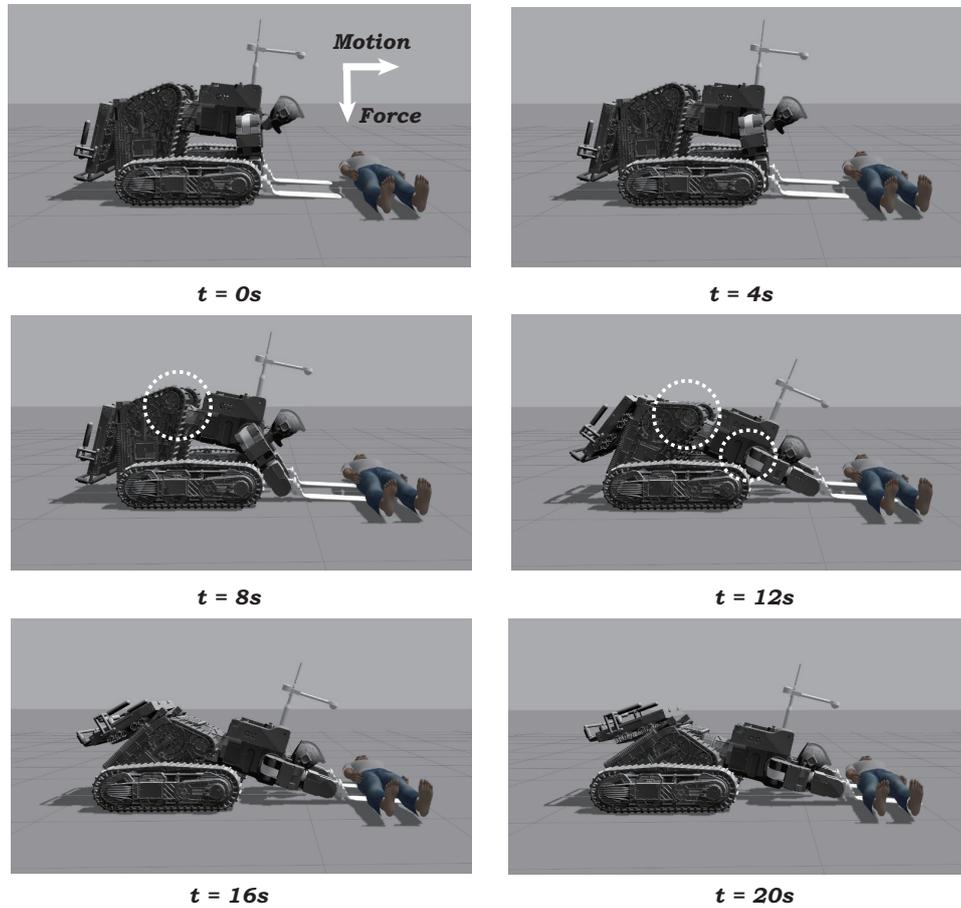
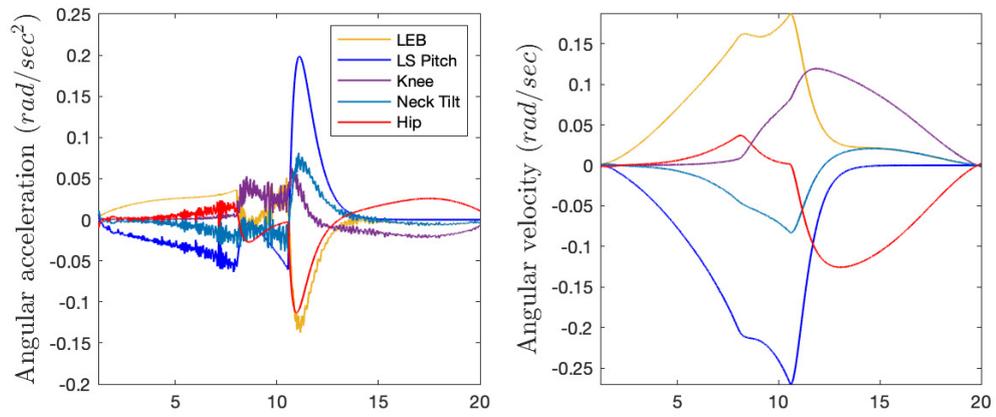
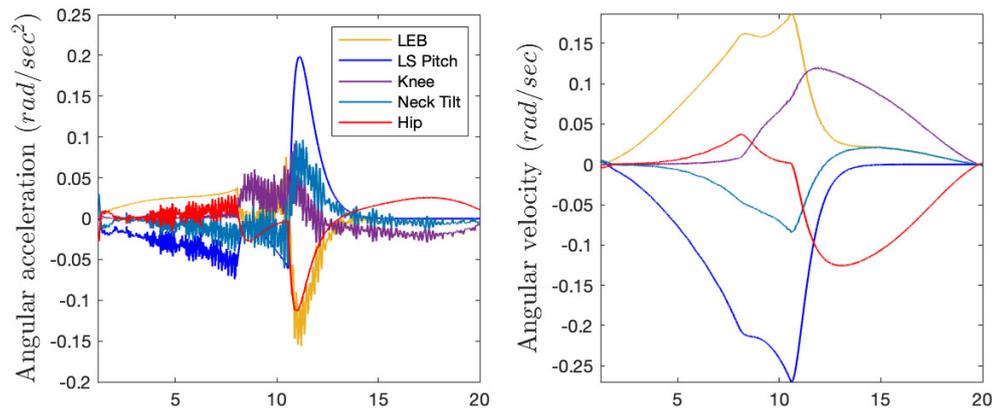


Figure 5.10: Snapshots of an injured acquisition motion with the motion and force control laws. The dashed lines represent the activated or deactivated joint limit constraints. The inequality constraint of the hip joint is activated at $t = 8$ s and deactivated $t = 11$ s. The RSP (right shoulder pitch) and LSP (left shoulder pitch) joint limit constraints are activated at 11 s. The arrows (top left corner) represent the directions of motion and force control where an orthogonal reference frame is specified.



(a) Force control with $C_k = 0.05$



(b) Force control with $C_k = 0.10$

Figure 5.11: Solution trajectories for hierarchical motion and force control with inner velocity loop for two compliance gains: (Left) angular acceleration (Right) angular velocity.

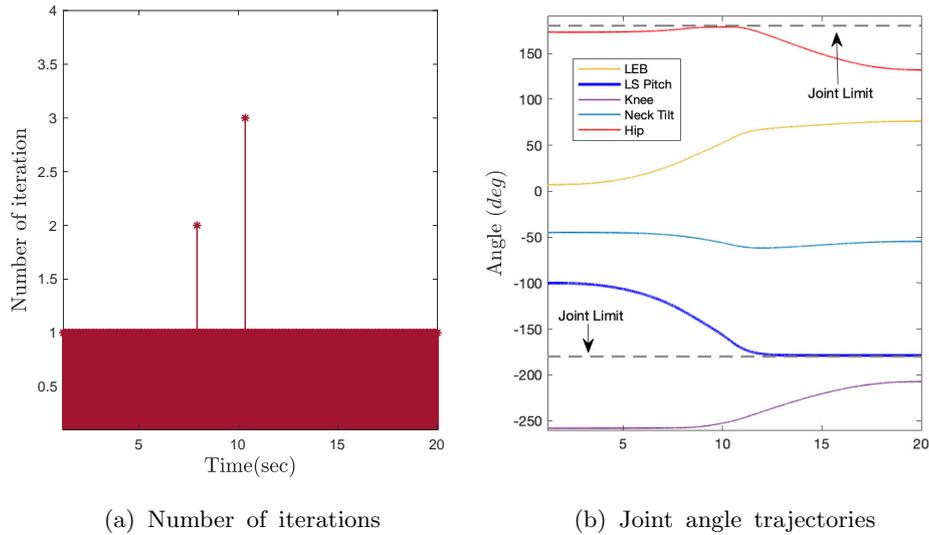


Figure 5.12: Number of iterations and joint angle trajectories. (a) The acceleration-based RHQP algorithm activates and deactivates the hip joint angle constraint at $t = 8$ s and $t = 11$ s respectively. The number of iteration is 2 at $t = 8$ s to activate the hip joint constraint. The number of iteration becomes 3 at $t = 11$ s, because the left and right shoulder joint constraints are activated and the hip joint constraint is deactivated. (b) Hip and shoulder pitch joint angle constraints are satisfied for stretching motion.

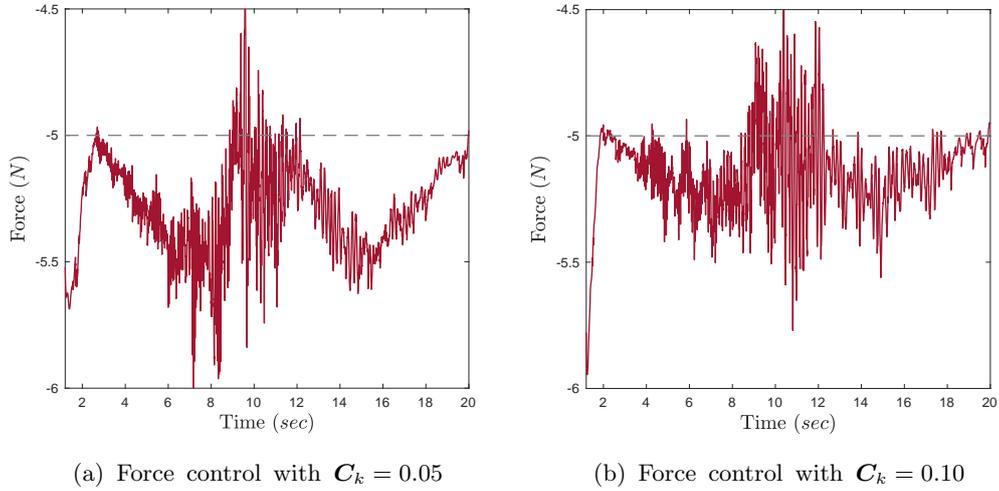


Figure 5.13: Numerical simulation results for measured force with hierarchical motion and force control laws.

The number of iteration for controlling inequality constraints and joint angle trajectories are presented in Figure 5.12. The number of iteration is 2 at $t = 8$ s because the inequality constraint of the hip joint becomes the active constraint. The hip joint angle is expected to reach the limit value imposed to prevent the battery and torso from colliding. Thus, the hip velocity starts to decrease as can be seen in Figure 5.11 (right). At the same time, the knee joint moves in order to execute the higher level tasks such as \mathbf{g}_{hpos} , \mathbf{g}_{force} and \mathbf{g}_{hpitch} . We can also confirm it through the robot poses and joint angle trajectories reported in Figure 5.10 and 5.12 (b) respectively. Thus, from $t = 8$ s, the position of the pelvis is moving by increasing the positional error of the pelvis task, and therefore we lose control over \mathbf{g}_{pel} .

In Figure 5.10, at the time of $t = 11$ s, as the lower arms extends further

forward, the shoulder pitch joints are expected to reach the limit values. Thus, corresponding constraints are activated and the joint velocity begins to decrease significantly as shown in Figure 5.11 (right). As the knee joint keeps moving, the hip joint begins to move again by being deactivated from the active constraint due to the knee motion. As the left and right shoulder joint constraints become active and the hip joint constraint is deactivated, and therefore the number of iteration is 3 at $t = 11$ s. Figure 5.12 clearly shows the proper inequality-constraint handling.

The time histories of the contact force are presented in Figure 5.13, where the contact force is regulated around desired value $-5N$, but a certain amount of force error exists for the compliance gain $C_k = 0.05$. To achieve a fast and accurate force regulation performance, a large magnitude of C_k and \dot{q}_k is required, but the motion controller with the limited control bandwidth makes it difficult to accomplish. Because the bandwidth of the joint space position controller should be higher than the bandwidth of the outer operational motion and force control loops for closed-loop stability [112, 124]. Thus the position controller makes the system stiff, and the outer force loop render it soft. If the compliance gain matrix C_k is too small to prevent a large value of \dot{q}_k , contact instability may occur and force control performance is degraded.

5.7 Velocity Resolved Motion and Force Control

We integrate the velocity-based RHQP algorithm with the motion and force control and then demonstrate the performance through experimental results

5.7.1 Velocity-Based Motion and Force Control

The velocity-resolved control is suitable for a tele-operation using the joystick or haptic device as depicted in Figure 4.10. The force control scheme applied at the velocity level is called *damping control* [128]. If the joint controllers can cancel out the effect of robot dynamics to a large extent, then the position controlled robot shows kinematic behavior expressed as

$$\dot{\mathbf{s}}_k = \mathbf{J}_k \dot{\mathbf{q}}_k, \quad (5.7.54)$$

where $\dot{\mathbf{s}}_k$ is the effector's operational space velocity and $\dot{\mathbf{q}}_k$ is the control input to be determined from RHQP.

Let us consider an elementary displacement between the environment rest position \mathbf{r}_k and the lower arm's position \mathbf{s}_k respectively. The corresponding elastic force \mathbf{f}_k is

$$\mathbf{f}_k = \mathbf{G}_k(\mathbf{s}_k - \mathbf{r}_k). \quad (5.7.55)$$

where \mathbf{G}_k is a stiffness matrix of the environment. The reference velocity input $\dot{\mathbf{q}}_k$ for force control is chosen with an active compliance gain matrix \mathbf{C}_k as follows:

$$\mathbf{J}_k \dot{\mathbf{q}}_k := \mathbf{C}_k({}^d \mathbf{f}_k - \mathbf{f}_k), \quad (5.7.56)$$

where ${}^d \mathbf{f}_k$ is the constant desired force and \mathbf{f}_k is the measured force. Therefore, the force error is converted into the velocity command with the compliance matrix \mathbf{C}_k .

With the elastically compliant environment (5.7.55) and the force control law (5.7.56), the operational space dynamics (5.7.54) is rewritten as follows:

$$\dot{\mathbf{s}}_k + \mathbf{C}_k \mathbf{G}_k \mathbf{s}_k = \mathbf{C}_k {}^d \mathbf{f}_k + \mathbf{C}_k \mathbf{G}_k \mathbf{r}_k. \quad (5.7.57)$$

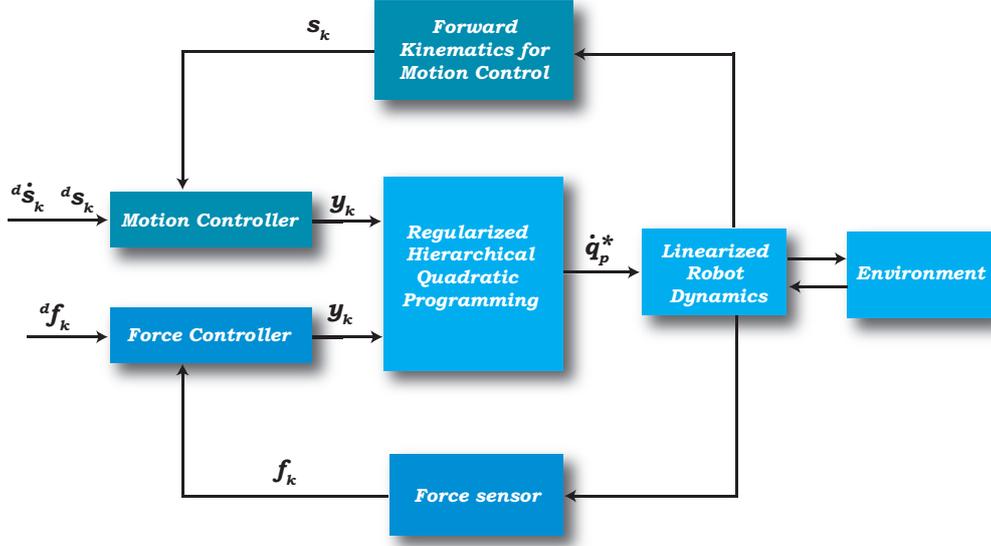


Figure 5.14: Block diagram of velocity-based motion and force control

A stable equilibrium point is reached since $\mathbf{C}_k \mathbf{G}_k$ is positive definite matrix. The equilibrium point is

$$\mathbf{s}_k = \mathbf{r}_k + \mathbf{G}_k^{-1} {}^d \mathbf{f}_k. \quad (5.7.58)$$

From the elastically compliant environment of (5.7.55), the equilibrium point leads to

$$\mathbf{f}_k = {}^d \mathbf{f}_k. \quad (5.7.59)$$

Thus we can obtain $\mathbf{f}_k = {}^d \mathbf{f}_k$ at steady state.

The force control law (5.7.56) can be integrated with the RHQP formulation

of (3.4.98) as follows:

$$\min_{\dot{\mathbf{q}}_k, \mathbf{e}_k} \frac{1}{2} \|\mathbf{e}_k\|_2^2 + \frac{1}{2} \rho_k^2 \|\mathbf{Y}_k^T \dot{\mathbf{q}}_k\|_2^2 \quad (5.7.60a)$$

$$\text{subject to } \mathbf{J}_k \dot{\mathbf{q}}_k - \mathbf{y}_k \leq \mathbf{e}_k \quad (5.7.60b)$$

$$\underline{\mathbf{J}}_{k-1} \dot{\mathbf{q}}_k - \underline{\mathbf{y}}_{k-1} \leq \underline{\mathbf{e}}_{k-1}^* \quad (5.7.60c)$$

where the tasks \mathbf{y}_k are given for motion and force control respectively

$$\mathbf{y}_k := -\mathbf{K}_k \mathbf{g}_k + {}^d \dot{\mathbf{s}}_k, \quad (5.7.61)$$

$$\mathbf{y}_k := \mathbf{C}_k ({}^d \mathbf{f}_k - \mathbf{f}_k). \quad (5.7.62)$$

The optimal solution for force control is obtained by using (3.3.37)

$$\dot{\mathbf{q}}_k^* = \dot{\mathbf{q}}_{k-1}^* + \mathbf{Y}_k \hat{\Sigma}_k^{-1} \mathbf{U}_k^T (\mathbf{y}_k - \mathbf{J}_k \dot{\mathbf{q}}_{k-1}^*), \quad (5.7.63)$$

where $\hat{\Sigma}_k^{-1} := (\Sigma_k^2 + \rho_k^2 \mathbf{I}_k)^\dagger \Sigma_k$.

The angular velocity solution $\dot{\mathbf{q}}_p^*$ and angle solution \mathbf{q}_p^* are directly used as reference trajectories for the position controller in (5.5.49). Figure 5.14 shows the block diagram of the k level motion and force control framework. The control structure of the velocity-based approach is simpler than that of acceleration-based force control presented in Figure 5.5. However, the second order controlled dynamics can be obtained from the acceleration-based force control.

5.7.2 Experimental Results

The experimental results for the injured acquisition movements are depicted in Figure 5.15 and 5.16 in terms of robot poses and time history of contact force respectively. From Figure 5.16, we can see that the force controller regulates the contact force to the desired value $-5N$. Note that there is no significant force error

at steady state, and it verifies that the controlled dynamics stated in (5.7.57) is valid with the equilibrium point (5.7.58), and therefore $\mathbf{f}_k = {}^d\mathbf{f}_k$ is accomplished at steady state.

As the compliance gain increases, the faster response is obtained. In fact, when the lower arm contacts with the stiff environment, the compliance gain matrix \mathbf{C}_k in (5.7.56) should be large to prevent excessive contact forces. In this case, although a large magnitude of $\dot{\mathbf{q}}_k$ is also required, the joint position controller with a limited control bandwidth make it difficult to achieve fast force regulation performance. As the compliance gain \mathbf{C}_k increases, higher bandwidth of position control loop is required. This is one of the limitations of the implicit force control when a low impedance or high gain \mathbf{C}_k is desired. Consequently, admittance control shows high performance in non-contact tasks but it can result in instability when interacting with the stiff environment. On the contrary, small values of \mathbf{C}_k can provide small damped movement, and it causes contact instability [116, 129]. The instability problem arises when the robot comes into contact with the stiff environment [130].

The impact induced instability or transition phase problem occurring from free to constrained motion is mitigated by a passive compliance mechanism. The impact force is observed from the numerical simulation result as shown in Figure 5.8, where the passive compliance mechanism is not considered in the numerical model. However, experimental result, Figure 5.16 shows that smooth transition is achieved. Because the Hurcules robot is equipped with the spring element near the elbow joints as depicted in Figure 4.8.

Generally, the response of the passive compliance control is much faster than the active force control algorithm. Moreover, the passive approach is simple and cheap. However, the passive compliance control can only handle small deviations

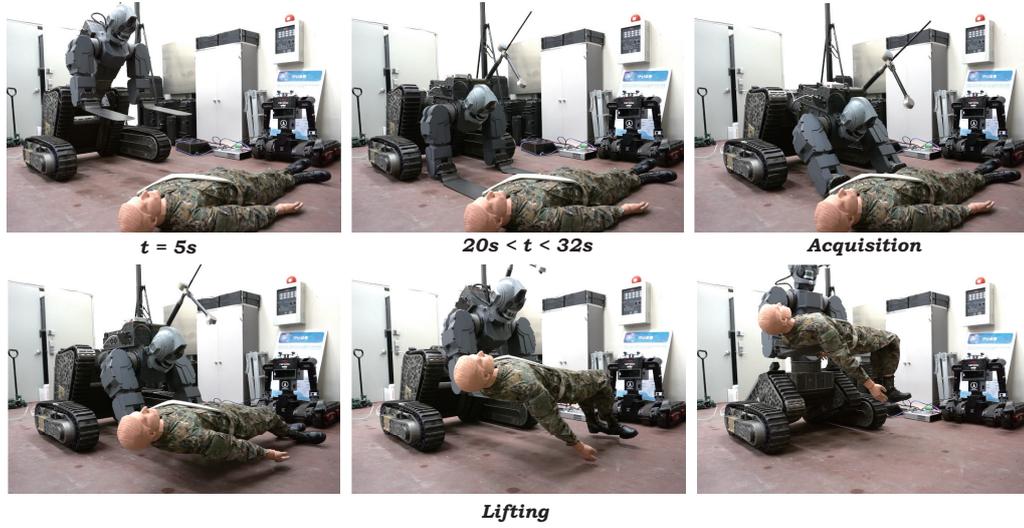


Figure 5.15: Injured acquisition movements with the velocity-based of RHQP. The force control gain $C_k = 0.05$ is applied.

from the rest position, and it is difficult to accurately regulate the interaction force. In order to overcome these drawbacks and to obtain a reasonable force regulation accuracy and stability, active force control is exploited in combination with the passive compliance mechanism.

For safety, the value of measured force $f_{k,r}$ is restricted between

$$|f_{k,r}| \in [f_{min}, f_{max}], \quad (5.7.64)$$

where k, r denotes the r^{th} force component of the level k . If the magnitude of the measured force $|f_{k,r}|$ exceeds $f_{min} = 2.5N$, then force controller begins to operate as shown in Figure 5.16. The threshold f_{min} prevents the end-effector to move when it has not yet been in contact with the ground ($f_{k,r} = 0$). If the magnitude of the measured force exceeds f_{max} , the robot enters emergency mode for safety.

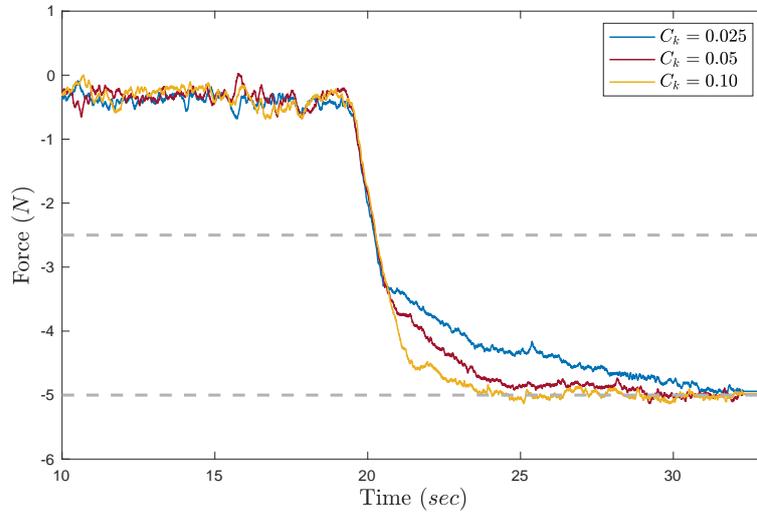


Figure 5.16: Experimental results for velocity-based motion and force control. Time history of the estimated contact force for various compliance gains. As the compliance gain increases, the faster response is obtained.

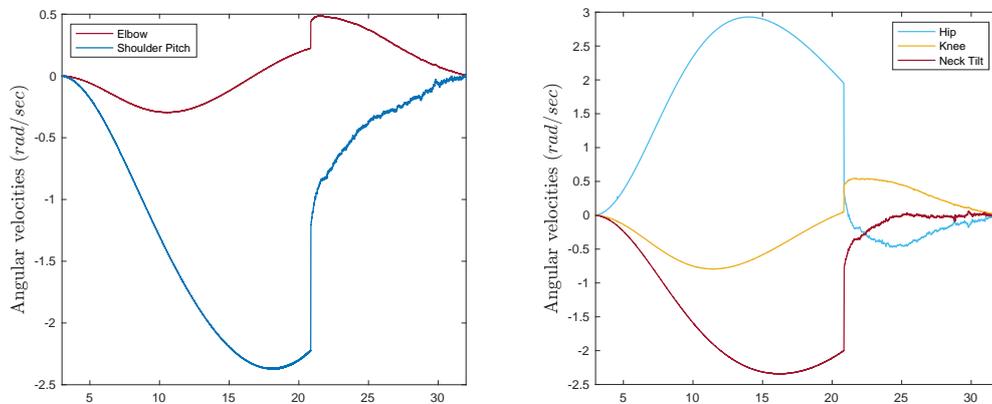


Figure 5.17: Joint velocity solutions for motion and force control. The force control gain $C_k = 0.05$ is applied.

Figure 5.17 shows the experimental results in terms of velocity solution trajectories. Near the time of $t = 21$ s, the angular velocities of all joints suddenly change to accommodate to the interaction force after $|f_{k,r}|$ reaches the threshold f_{min} . As can be seen in Figure 5.15, the robot movements with force control law demonstrate that the performance of the force control is good enough to achieve the lower arms' successful penetration motion even on the stiff ground. High value of contact and friction force with the ground prevents the robot to achieve this motion. Otherwise, if the lower arms do not contact the ground with appropriate force, the injured person is pushed out by the lower arms.

5.8 Concluding Remarks

The RHQP algorithm is extended to hierarchical task-priority motion and force control to ensure singularity robustness and proper inequality-constraint handling in a single hierarchical control loop. We demonstrated that the proposed RHQP solver can obtain the robust optimal solution in the face of singularity. Moreover, the algorithm activates a blocking constraint and deactivates an unnecessary constraint at the right time in the acceleration level. The velocity-based RHQP algorithm is also integrated with the motion and force control schemes.

For acceleration-based force control, a spring-mass-damper system is realized with the additional operational space inner position or velocity control laws, and then the force control is performed by closure of the force control loop around this inner loop. Instead of using a derivative action of noisy force measurement, the stabilizing damping and spring effects are obtained by the operational space position or velocity feedback. Therefore, the force control scheme is realized by a stabilizing PD control action based on velocity and position measurements as well

as force error. Moreover, steady-state force error is rejected by suitably designing the controlled dynamics.

We show that a safe interaction force is guaranteed from the motion control loop in conjunction with an outer force feedback loop through numerical and experimental results. Finally, the Hurcules robot successfully extracts an injured lying on the ground.

6

Conclusion

6.1 Summary

This dissertation proposed a novel hardware design concept and hierarchical task-priority based motion planning and control methodologies for a rescue robot to operate in a highly unstructured environment. In summary, the main contributions are described as follows:

- **Robust Hierarchical Task-Priority Control**

The main objective is to find the optimal and robust solution for online hierarchical optimization for whole-body robotic manipulation subject to both equality and inequality tasks or constraints. A generic solver is developed for regularized hierarchical quadratic program (RHQP) without resorting to any off-the-shelf QP solver for fast and efficient transcription of the objectives into the configuration space motion. We concentrate on the reasoning about the task regularization to ensure the convergence and robustness of

the solution in the face of singularity. The regularization is fully integrated into the inequality-constrained hierarchical optimization by considering regularization from the very beginning of the problem.

The robust optimal solution is obtained from RHQP while resolving regularization induced problems such as a non-convexity, task errors, difficulty to obtain Lagrange multipliers, and not leaving redundant resources from higher to lower levels. As a singular value decomposition centric approach is leveraged, all hierarchical solutions and Lagrange multipliers are analytically obtained while considering regularization induced errors. We mathematically show that regularization requires for us to trade off robustness against performance. The small coupling error induced by regularization is the only price we have to pay instead of guaranteeing convergence and robustness of the solution in the vicinity of singular points.

The RHQP solver is able to find the optimal solution efficiently by considering all level's hierarchical tasks together with only one active set. The Lagrange multiplier matrix eliminates the iterative process used to find each level optimal active set. All necessary Lagrange multiplier and optimal solution are obtained in a recursive procedure. As the modified active set algorithm does not need a step direction nor a step length usually used in numerical optimization, it is simple and fast. Moreover, it operates regardless of whether the initial point is feasible or not for an initial working set. Consequently, under the equality and inequality constraints, the proposed algorithm works fast.

As a result, we thoroughly investigated the underlying principles of regularized hierarchical optimization for whole-body robot motion control.

- **Rescue Robot Design and Experimental Results**

We propose a new hardware design concept for a rescue robot to safely extract a patient and to dispose a dangerous object instead of humans. The robot is designed to realize how a mechanical structure is developed to simultaneously obtain both manipulation capability and all-terrain mobility. The upper body has humanoid in form with replaceable dual arms, and the lower body is featured with the hybrid tracked and legged mobile platform. Thus, the robot can execute a driving task, dangerous object manipulation, and casualty extraction missions successfully over various kinds of difficult terrain by changing the pose and replaceable modularized equipments in an optimized manner.

The effectiveness and performance of the RHQP solver are demonstrated through extensive numerical simulations and experimental tests using the Hurcules robot. We show that the task regularization and the inequality-constrained optimization are fully integrated via regularized hierarchical optimization to guarantee the robustness and convergence of the solution in the face of singularity. The inequality constraints are also properly handled without showing any cycling. Particularly, without considering the regularization errors in Lagrange multipliers, the optimization algorithm can bring about a cycling in the neighborhood of a singularity, and thus it yields the same solution and working set repeatedly to an earlier iteration.

Due to the recursive structure of the solutions and the modified active set method, the RHQP algorithm operates fast enough for use in real-time motion planning and control systems. The average computation time for the 19-DoF rescue robot is less than 0.1 *ms* on a typical personal computer.

- **Hierarchical Motion and Force Control**

While maintaining all of the strength of RHQP, the algorithm is extended to motion and force control. The acceleration-based RHQP method is also able to acquire the robust optimal solution when facing singularity. Moreover, the method activates a blocking constraint and deactivates an unnecessary constraint at the right time.

For acceleration-based force control, a spring-mass-damper system is realized with the additional operational space inner position or velocity controllers, and then the force control is performed according to this system. Instead of using a derivative action of the noisy force measurement, the stabilizing damping and spring effects are obtained by the operational space position or velocity feedback. Therefore, the force control scheme is composed of a stabilizing PD control action based on velocity and position measurements as well as the outer force feedback loop. Moreover, steady-state force error is also rejected by suitably designing the controlled dynamics with the task space velocity feedback. The first order velocity-based RHQP algorithm is also integrated with the motion and force control.

As a result, we show that a safe interaction force is guaranteed and the Hurcules robot can extract an injured lying on the ground.

Throughout the dissertation, all proposed design methods and online optimization-based control algorithms are validated via extensive numerical simulations and experimental tests. We show that the Hurcules robot can successfully execute injured extraction and explosive ordnance disposal missions in both indoor and outdoor environment that none of the existing robots has performed.

6.2 Concluding Remarks

The RHQP algorithm would be applied to any other regularized hierarchical optimization problems such as motion generation [38, 44], parameter estimation [131], operational space inverse dynamics control [132, 133, 134, 135, 136]. Online optimization based control in robotics are becoming powerful warehouse for ever more complex robotic systems such as humanoids and unmanned autonomous vehicles [137, 138, 139]. Recently the roles of robots are no longer limited to the automation in the industrial process but are changing into explorers for a wide range of unstructured environment. To more safely interact with the complex environment while achieving high performance, model based control strategies [87, 140, 141] and learning-based approaches [58, 142, 143] should be tightly integrated.

The model based control can face difficult problems in the real world if a dynamic model is unavailable or uncertain. Then the performance degradation and unstable behaviors can occur due to the unmodelled dynamics and nonlinearities [50]. Nowadays a grey box model becomes more popular, and there are a nonlinear model with a partial theoretical structure and some unknown parameters or models estimated from data [59, 60, 144].

The tremendous progress in machine learning over the last decades offers us the promise of less human driven approaches to motor skill acquisition [51, 145, 146, 147, 148, 149]. Especially, reinforcement learning is closely related to the optimal control theory and dynamic programming that most robotics and control engineers are very familiar with. Both reinforcement learning and optimal control have the same goal to find an optimal control or policy that optimizes an objective function. If a robot task can be specified as an optimization problem with temporal structure, reinforcement learning can be leveraged to find the optimal

policy [150, 151]. The RHQP algorithm is also interpreted as the model-based reinforcement learning problem that maximizes an immediate reward. A regularized hierarchical differential dynamic programming can solve the hierarchical optimization problem for robot motion control [47]. At each level, the algorithm involves iterating forward pass computing the cost of all tasks for a given locally optimal control policy, and it is followed by a backward pass which finds a locally optimal solution.

However, the reinforcement learning method can be very complex problem by requiring interaction with the environment, sequential prediction, handling curse of dimensionality due to the high dimensional continuous robot states, and complex reward structure [152]. Moreover, the learning methods should be safe, data efficient, and reliable because a robot interacting with the physical world suffers from cost, physical labor, long period of experiment for learning, and safety critical issues. Thus, some problems of the reinforcement learning approach can be addressed by the optimal control strategies, and vice versa [153, 154, 155, 156, 157, 158, 159, 160].

We expect a robot to be safer, more dynamic and agile than the current state-of-the-art. We hope that the real-time hierarchical optimization-based control method proposed in this dissertation is extended to synthesize human like optimal robotic behaviors at runtime with the model and data driven approaches, and further extended in several ways to allow robots to perform more dynamic, robust, and complex tasks in a human centered environment.

A

Appendix

A.1 Introduction to PID Control

Even though the control theory has experienced a striking development in the last 50 years [161], PID controller still strives to survive today. More than 90% of industrial controllers are still implemented based on PID control or its variations [162]. The three actions of a PID controller execute three common requirements of most control problems. Each term in the PID controller is reasonably intuitive, and it allows control engineers to intuitively understand the essential functionality of the controller's action [163, 164, 165, 166].

As the integral action is able to track a set point value, zero steady-state error can be obtained and constant disturbance rejection is also ensured. While integral control filters higher frequency sensor noise, it is slow in response to the current error. However, the proportional action responds immediately to the current error, yet typically cannot achieve the fast response to reach the desired set point without an unacceptably large gain. There may be large transient errors when PI

control is used. Derivative action resolves this problem by using a prediction of future error.

The \mathcal{H}_∞ control is a suitable design framework to accomplish both stability and robustness in the face of uncertainty, and it proves to be able to provide promising solution for linear and nonlinear systems [167, 168, 169, 170, 171]. However, nonlinear \mathcal{H}_∞ control methods have not been widely used because the complex partial differential Hamilton-Jacobi-Isaacs (HJI) equation should be solved [172, 173]. At this point, it is worthwhile to note the result that extends the theoretical basis of PID control to robust \mathcal{H}_∞ control theory. The \mathcal{H}_∞ optimality of PID control is satisfied for trajectory tracking control of Lagrangian system using inverse optimal control technique [53, 98, 99]. For the inverse optimal control problems, we are searching for not only a control law but also state weighting $\mathbf{Q}(\mathbf{x})$ and control input weighting matrix $\mathbf{R}(\mathbf{x})$. The inverse problem is more tractable than the direct one in which $\mathbf{Q}(\mathbf{x})$ and $\mathbf{R}(\mathbf{x})$ are given and then the HJI partial differential equation is solved.

The maximum weight lifting capacity of Hurcules is shown in Figure 4.4. The robot should attain an ability to cope with large payload variation from 0kg to 120kg, and hence robust joint controllers are required. Here, we briefly describes the design strategies of robust PID controllers by reorganizing and summarizing the method proposed by [98]. The effectiveness of the inverse optimal PID controller is demonstrated through experimental results with the Hurcules robot.

A.2 Inverse Optimal Control

Consider the following Lagrangian equation of motions

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{d} = \boldsymbol{\tau}, \quad (\text{A.2.1})$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the symmetric positive definite inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ denotes the nonlinear forces including Coriolis/centrifugal torque, $\mathbf{g}(\mathbf{q})$ is gravitational torque, $\boldsymbol{\tau}$ is the actuation torques, and \mathbf{d} is the external disturbance. The extended disturbance is defined for the trajectory tracking control system as:

$$\begin{aligned} \mathbf{w}\left(t, \dot{\mathbf{e}}, \mathbf{e}, \int \mathbf{e}\right) &:= \mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_d + \mathbf{K}_P \dot{\mathbf{e}} + \mathbf{K}_I \mathbf{e}) \\ &+ \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\left(\dot{\mathbf{q}}_d + \mathbf{K}_P \mathbf{e} + \mathbf{K}_I \int \mathbf{e}\right) + \mathbf{g}(\mathbf{q}) + \mathbf{d}(t), \end{aligned} \quad (\text{A.2.2})$$

where \mathbf{K}_P , \mathbf{K}_I are the constant diagonal gain matrices, $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$ is the configuration error. Applying extended disturbance (A.2.2) to the Lagrangian system of (A.2.1) yields the following trajectory tracking control system

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{s}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} = \mathbf{w}\left(t, \dot{\mathbf{e}}, \mathbf{e}, \int \mathbf{e}\right) + \mathbf{u}, \quad (\text{A.2.3})$$

where $\mathbf{u} = -\boldsymbol{\tau}$ and $\mathbf{s} = \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} + \mathbf{K}_I \int \mathbf{e} dt$. If the state vector is defined as $\mathbf{x} = [\int \mathbf{e}^T, \mathbf{e}^T, \dot{\mathbf{e}}^T] \in \mathbb{R}^{3n}$ for the controlled system (A.2.3), then the state-space model is obtained as follows:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}, t)\mathbf{x} + \mathbf{B}(\mathbf{x}, t)\mathbf{w} + \mathbf{B}(\mathbf{x}, t)\mathbf{u}, \quad (\text{A.2.4})$$

where

$$\mathbf{A}(\mathbf{x}, t) = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{C}\mathbf{K}_I & -\mathbf{M}^{-1}\mathbf{C}\mathbf{K}_P - \mathbf{K}_I & -\mathbf{M}^{-1}\mathbf{C} - \mathbf{K}_P \end{bmatrix},$$

and

$$\mathbf{B}(\mathbf{x}, t) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{M}^{-T} \end{bmatrix}^T.$$

The nonlinear \mathcal{H}_∞ control is formulated as the induced \mathcal{L}_2 -gain attenuation problem of a nonlinear closed-loop system [174]. The energy gain is equal to or less than a specified value $\gamma > 0$. Let the \mathcal{H}_∞ performance index J be

$$J(t, \mathbf{x}, \mathbf{u}, \mathbf{w}) = \lim_{t \rightarrow \infty} \left[2V(\mathbf{x}(t), t) + \int_0^t (\mathbf{x}^T \mathbf{Q}(\mathbf{x}) \mathbf{x} + \mathbf{u}^T \mathbf{R}(\mathbf{x}) \mathbf{u} - \gamma^2 \mathbf{w}^T \mathbf{w}) d\sigma \right]. \quad (\text{A.2.5})$$

The Hamilton Jacobi Issacs(HJI) equation is derived from the optimization for the \mathcal{H}_∞ performance index. The difficulty of finding solutions for an HJI equation can be tackled by an inverse optimal controller. The inverse optimal control problems are to search for, not only a control law, but also function $\mathbf{Q}(\mathbf{x})$ when there is given Lyapunov function $\mathbf{V}(\mathbf{x})$ which is the solution of HJI. While the control input weighting matrix $\mathbf{R}(\mathbf{x})$ can be found inversely. Suppose that the control input weighting is given as follows:

$$\mathbf{R}(\mathbf{x}) = \left(\mathbf{K} + \frac{1}{\gamma^2} \mathbf{I} \right)^{-1}. \quad (\text{A.2.6})$$

Let the Lyapunov function be

$$\mathbf{V}(\mathbf{x}, t) = \frac{1}{2} \mathbf{x}^T \mathbf{P}(\mathbf{x}, t) \mathbf{x}. \quad (\text{A.2.7})$$

Here Lyapunov matrix is given by [99]

$$\mathbf{P}(\mathbf{x}, t) = \begin{bmatrix} \mathbf{K}_I \mathbf{M} \mathbf{K}_I + \mathbf{K}_I \mathbf{K}_P \mathbf{K} & \mathbf{K}_I \mathbf{M} \mathbf{K}_P + \mathbf{K}_I \mathbf{K} & \mathbf{K}_I \mathbf{M} \\ \mathbf{K}_P \mathbf{M} \mathbf{K}_I + \mathbf{K}_I \mathbf{K} & \mathbf{K}_P \mathbf{M} \mathbf{K}_P + \mathbf{K}_P \mathbf{K} & \mathbf{K}_P \mathbf{M} \\ \mathbf{M} \mathbf{K}_I & \mathbf{M} \mathbf{K}_P & \mathbf{M} \end{bmatrix}, \quad (\text{A.2.8})$$

where the positive definiteness of $\mathbf{P}(\mathbf{x}, t)$ is guaranteed if the following conditions are satisfied

$$\mathbf{K}, \mathbf{K}_P, \mathbf{K}_I > \mathbf{0}, \quad \mathbf{K}_P^2 > 2\mathbf{K}_I, \quad \gamma > 0, \quad (\text{A.2.9})$$

where \mathbf{K} , \mathbf{K}_p and \mathbf{K}_I are constant diagonal gain matrices. The HJI equation can be reformulated to the following differential matrix Riccati equation

$$\dot{\mathbf{P}} + \mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} + \frac{1}{\gamma^2} \mathbf{P} \mathbf{B} \mathbf{B}^T \mathbf{P} + \mathbf{Q} = \mathbf{0}. \quad (\text{A.2.10})$$

Then the state weighting matrix can be inversely obtained from the differential Riccati equation (A.2.10) by using the characteristics $\dot{\mathbf{M}} - \mathbf{C}^T - \mathbf{C} = \mathbf{0}$ of Lagrangian system and control input weighting (A.2.6)

$$\begin{aligned} \mathbf{Q} &= -(\dot{\mathbf{P}} + \mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{K} \mathbf{B}^T \mathbf{P}) \\ &= \begin{bmatrix} \mathbf{K}_I^2 \mathbf{K} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\mathbf{K}_P^2 - 2\mathbf{K}_I) \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K} \end{bmatrix}, \end{aligned} \quad (\text{A.2.11})$$

where \mathbf{Q} is a diagonal and constant matrix, and positive definite because of $\mathbf{K}_P^2 > 2\mathbf{K}_I$. The minimum \mathcal{H}_∞ performance index is achieved by the following control law

$$\mathbf{u} = -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \mathbf{x} = \left(\mathbf{K} + \frac{1}{\gamma^2} \mathbf{I} \right) \left(\dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} + \mathbf{K}_I \int \mathbf{e} \right) = \left(\mathbf{K} + \frac{1}{\gamma^2} \mathbf{I} \right) \mathbf{s}, \quad (\text{A.2.12})$$

and it happen to have PID control structure.

The magnitude of state-weighting matrix \mathbf{Q} of (A.2.11) effects on the system performance or control errors. If we would like to enhance the performance by two times, the magnitude of the diagonal elements in state-weighting \mathbf{Q} should be four times larger. Then the control errors would be reduced by half times. It means that the magnitude of a matrix \mathbf{K} in state weighting matrix should be larger. At the

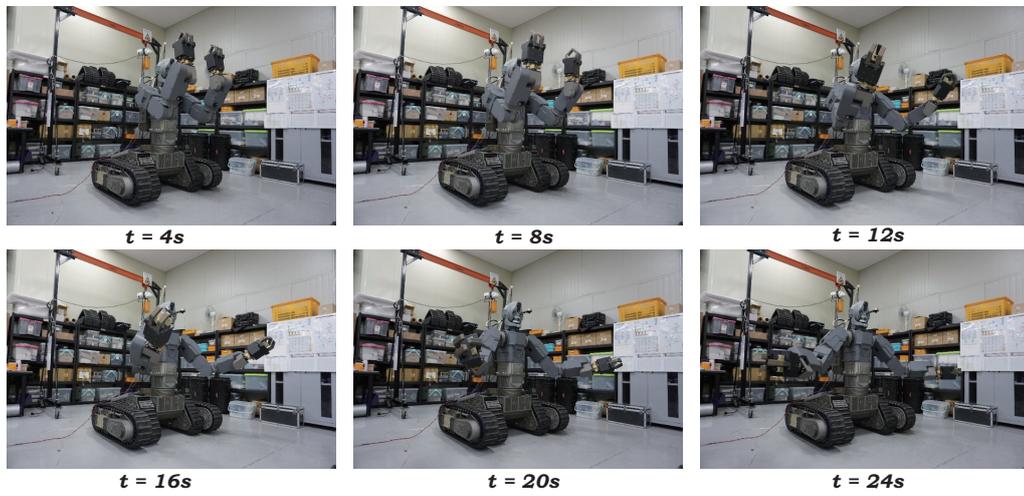


Figure A.1: Gain tuning postures with the seven-DoF manipulator.

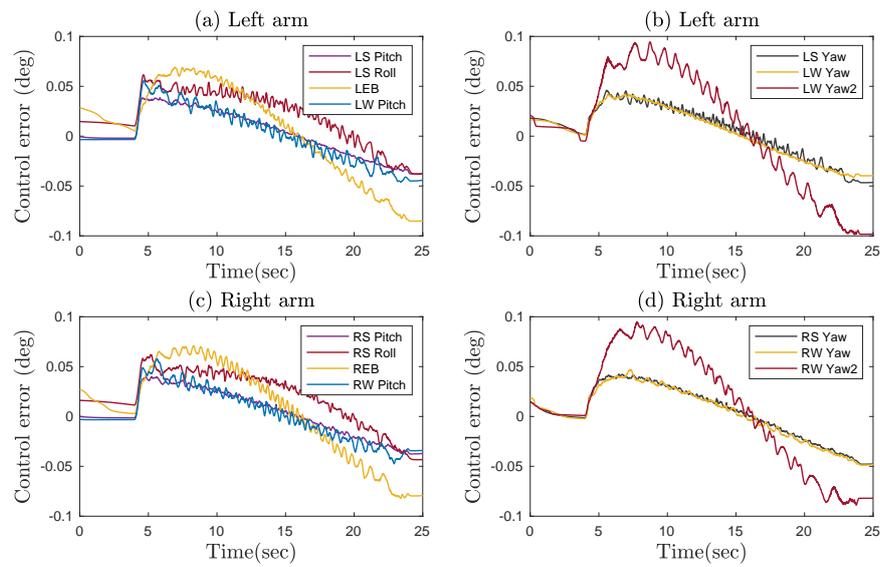


Figure A.2: Joint space control errors for the seven-DoF manipulators.

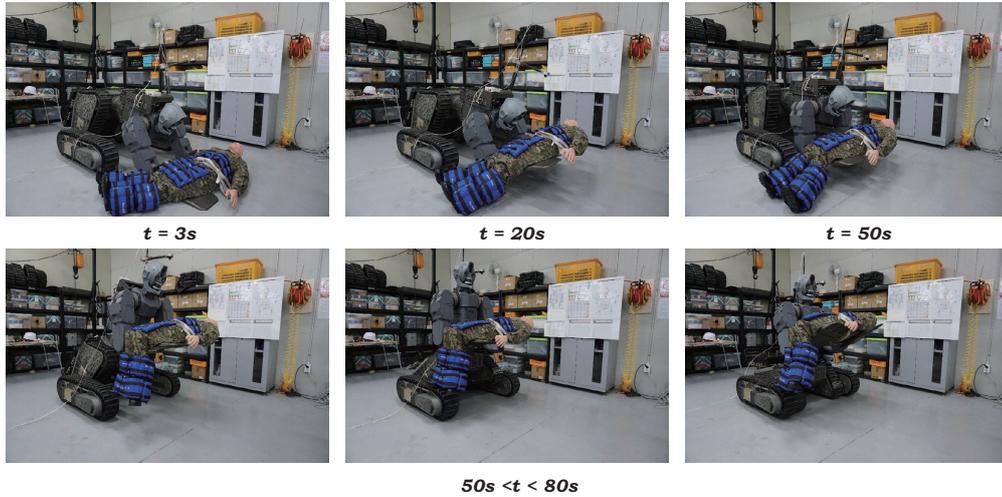


Figure A.3: Weight lifting motions with the maximum payload 120kg.

same time, the increased gain matrix \mathbf{K} decreases the magnitude of the control input weighting (A.2.6), and it requires more control energy.

On the contrary, if the magnitude \mathbf{K} is decreased, then it requires less control effort, and it induces larger control errors. The gain matrix \mathbf{K} shows the trade-off between the performance and control energy. It is well agreed with the result of linear quadratic optimal control theory [175]. Note that \mathcal{L}_2 -gain γ has an effect only on the control input weighting, and it enhances the control performance by increasing the disturbances rejection capability. The globally asymptotic stability is not guaranteed for a PID trajectory tracking controller, but extended disturbance input-to-state stability is guaranteed. It means that the behavior of the system remains bounded when the disturbance inputs are bounded, and the response should tend to equilibrium when the disturbance input tends to zero.

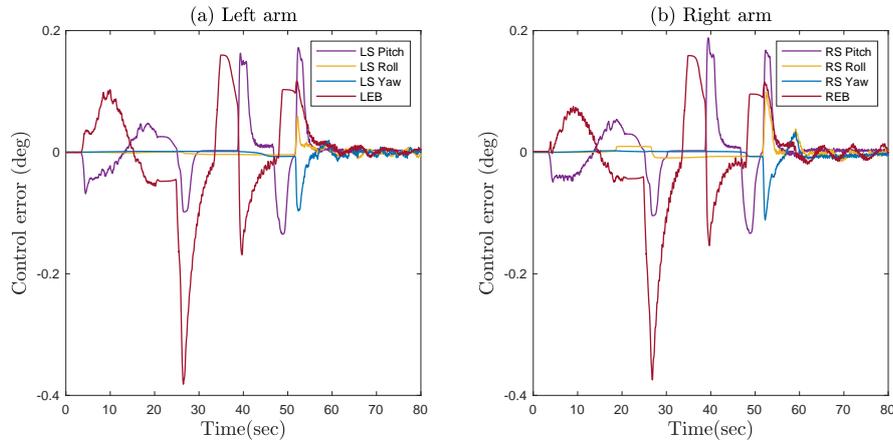


Figure A.4: Joint space control errors for the weighting lifting motions with the maximum payload (120kg).

A.3 Experimental Results and Conclusion

Experimental tests are conducted to verify the control performance under two different conditions. One experiment is conducted using the seven-DoF manipulator, and the other one was conducted using the four-DoF arm with the maximum load 120kg. The results are shown in Figure A.1, A.2, A.3, and A.4 where we can see that the controller is able to cope with large payload variation from 0kg to 120kg. Although the globally asymptotic stability is not guaranteed for a PID trajectory tracking controller, it guarantees the extended disturbance input to state stability for the Lagrangian system. From Figure A.4 we see that the maximum control error under the maximum load conditions is within 0.4 degrees. The robust PID joint controller maintains the control errors sufficiently small to execute a given rescue task.

Bibliography

- [1] Tamim Asfour, Kazuhito Yokoi, CS George Lee, and James Kuffner. Humanoid robotics. *IEEE Robotics & Automation Magazine*, 19(1):108–118, 2012.
- [2] Eric Krotkov, Douglas Hackett, Larry Jackel, Michael Perschbacher, James Pippine, Jesse Strauss, Gill Pratt, and Christopher Orłowski. The DARPA robotics challenge finals: Results and perspectives. *Journal of Field Robotics*, 34(2):229–240, nov 2016.
- [3] Gill Pratt and Justin Manzo. The darpa robotics challenge [competitions]. *IEEE Robotics & Automation Magazine*, 20(2):10–12, 2013.
- [4] Maurice Fallon, Scott Kuindersma, Sisir Karumanchi, Matthew Antone, Toby Schneider, Hongkai Dai, Claudia Pérez D’Arpino, Robin Deits, Matt DiCicco, Dehann Fourie, et al. An architecture for online affordance-based perception and whole-body planning. *Journal of Field Robotics*, 32(2):229–254, 2015.
- [5] Sanghyun Kim, Mingon Kim, Jimin Lee, Soonwook Hwang, Joonbo Chae, Beomyeong Park, Hyunbum Cho, Jaehoon Sim, Jaesug Jung, Hosang Lee, et al. Team snu’s control strategies for enhancing a robot’s capability: Lessons from the 2015 darpa robotics challenge finals. *Journal of Field Robotics*, 34(2):359–380, 2017.
- [6] Christopher G Atkeson, BPW Babu, N Banerjee, D Berenson, CP Bove, X Cui, M DeDonato, R Du, S Feng, P Franklin, et al. What happened at the darpa robotics challenge, and why. *submitted to the DRC Finals Special Issue of the Journal of Field Robotics*, 1, 2016.

- [7] Erico Guizzo and Evan Ackerman. Darpa robotics challenge: A compilation of robots falling down. *IEEE Spectrum*, 2015.
- [8] Duane Knudson. *Fundamentals of biomechanics*. Springer Science & Business Media, 2007.
- [9] Shuuji Kajita, Hirohisa Hirukawa, Kensuke Harada, and Kazuhito Yokoi. *Introduction to humanoid robotics*, volume 101. Springer, 2014.
- [10] Paolo Baerlocher and Ronan Boulic. Task-priority formulations for the kinematic control of highly redundant articulated structures. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 323–329, 1998.
- [11] Mingxing Liu, Yang Tan, and Vincent Padois. Generalized hierarchical control. *Autonomous Robots*, 40(1):17–31, 2016.
- [12] Jun Nakanishi, Rick Cory, Michael Mistry, Jan Peters, and Stefan Schaal. Operational space control: A theoretical and empirical comparison. *The International Journal of Robotics Research*, 27(6):737–757, 2008.
- [13] Luis Sentis and Oussama Khatib. A whole-body control framework for humanoids operating in human environments. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2641–2648. IEEE, 2006.
- [14] Karim Bouyarmane and Abderrahmane Kheddar. On weight-prioritized multitask control of humanoid robots. *IEEE Transactions on Automatic Control*, 63(6):1632–1647, 2018.

- [15] Rafael Cisneros, Mehdi Benallegue, Abdelaziz Benallegue, Mitsuharu Morisawa, Hervé Audren, Pierre Gergondet, Adrien Escande, Abderrahmane Kheddar, and Fumio Kanehiro. Robust humanoid control using a qp solver with integral gains. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7472–7479. IEEE, 2018.
- [16] Pierre-Brice Wieber, Adrien Escande, Dimitar Dimitrov, and Alexander Sherikov. Geometric and numerical aspects of redundancy. In *Geometric and Numerical Foundations of Movements*, pages 67–85. Springer, 2017.
- [17] Yoshihiko Nakamura, Hideo Hanafusa, and Tsuneo Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3–15, 1987.
- [18] Bruno Siciliano and Jean-Jacques Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. *Advanced Robotics*, pages 1211–1216, 1991.
- [19] Alexander Dietrich, Christian Ott, and Alin Albu-Schäffer. An overview of null space projections for redundant, torque-controlled robots. *The International Journal of Robotics Research*, 34(11):1385–1400, 2015.
- [20] Oussama Khatib, Luis Sentis, Jaeheung Park, and James Warren. Whole-body dynamic behavior and control of human-like robots. *International Journal of Humanoid Robotics*, 1(01):29–43, 2004.
- [21] Andrea Del Prete, Francesco Nori, Giorgio Metta, and Lorenzo Natale. Prioritized motion–force control of constrained fully-actuated robots: “task space inverse dynamics”. *Robotics and Autonomous Systems*, 63:150–157, 2015.

- [22] Alexander Dietrich, Thomas Wimbock, Alin Albu-Schaffer, and Gerd Hirzinger. Reactive whole-body control: Dynamic mobile manipulation using a large number of actuated degrees of freedom. *IEEE Robotics & Automation Magazine*, 19(2):20–33, 2012.
- [23] Fabrizio Flacco, Alessandro De Luca, and Oussama Khatib. Control of redundant robots under hard joint constraints: Saturation in the null space. *IEEE Transactions on Robotics*, 31(3):637–654, 2015.
- [24] Nicolas Mansard and François Chaumette. Task sequencing for high-level sensor-based control. *IEEE Transactions on Robotics*, 23(1):60–72, 2007.
- [25] Nicolas Mansard, Anthony Remazeilles, and François Chaumette. Continuity of varying-feature-set control laws. *IEEE Transactions on Automatic Control*, 54(11):2493–2505, 2009.
- [26] Nicolas Mansard and François Chaumette. Directional redundancy for robot control. *IEEE Transactions on Automatic Control*, 54(6):1179–1192, 2009.
- [27] Luis Sentis, Jaeheung Park, and Oussama Khatib. Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *IEEE Transactions on robotics*, 26(3):483–501, 2010.
- [28] Gianluca Antonelli. Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. *IEEE Transactions on Robotics*, 25(5):985–994, 2009.
- [29] An Sang-Ik and Dongheui Lee. Prioritized inverse kinematics: Generalization. *IEEE Robotics and Automation Letters*, 40(4):3537–3544, 2019.

- [30] Christian Ott, Alexander Dietrich, and Alin Albu-Schäffer. Prioritized multi-task compliance control of redundant manipulators. *Automatica*, 53:416–423, 2015.
- [31] Alexander Dietrich, Christian Ott, and Jaeheung Park. The hierarchical operational space formulation: Stability analysis for the regulation case. *IEEE Robotics and Automation Letters*, 3(2):1120–1127, 2018.
- [32] Jan Peters, Michael Mistry, Firdaus Udwadia, Jun Nakanishi, and Stefan Schaal. A unifying framework for robot control with redundant dofs. *Autonomous Robots*, 24(1):1–12, 2008.
- [33] Youngjin Choi. Singularity-robust inverse kinematics using lagrange multiplier for redundant manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 130(5):051009, 2008.
- [34] Oussama Khatib, Emel Demircan, Vincent De Sapio, Luis Sentis, Thor Besier, and Scott Delp. Robotics-based synthesis of human motion. *Journal of Physiology-Paris*, 103(3-5):211–219, 2009.
- [35] Emel Demircan, Thor F Besier, and Oussama Khatib. Muscle force transmission to operational space accelerations during elite golf swings. In *2012 IEEE International Conference on Robotics and Automation*, pages 1464–1469. IEEE, 2012.
- [36] Nicolas Mansard. *Semiotics of Motion: Toward a Robotics Programming Language*. PhD thesis, University of Toulouse, 2013.
- [37] Oussama Kanoun. *Contribution à la planification de mouvement pour robots humanoïdes*. PhD thesis, University of Toulouse, 2009.

- [38] Oussama Kanoun, Florent Lamiroux, and Pierre-Brice Wieber. Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks. *IEEE Transactions on Robotics*, 27(4):785–792, 2011.
- [39] Martin de Lasa, Igor Mordatch, and Aaron Hertzmann. Feature-based locomotion controllers. *Transactions on Graphics*, 29(4):131, 2010.
- [40] Alexander Herzog, Nicholas Rotella, Sean Mason, Felix Grimmering, Stefan Schaal, and Ludovic Righetti. Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. *Autonomous Robots*, 40(3):473–491, 2016.
- [41] Hilario Tomé, Luca Marchionni, and RA Tsouroukdissian. Whole body control using robust & online hierarchical quadratic optimization. In *IROS14 International Conference on Intelligent Robots and Systems*, pages 14–18, 2014.
- [42] Marcus Johansson. Online whole-body control using hierarchical quadratic programming: Implementation and evaluation of the hiqp control framework, 2016.
- [43] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Fast resolution of hierarchized inverse kinematics with inequality constraints. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 3733–3738, 2010.
- [44] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, 2014.

- [45] Dimitar Dimitrov, Alexander Sherikov, and Pierre-Brice Wieber. Efficient resolution of potentially conflicting linear constraints in robotics. <https://hal.inria.fr/hal-01183003>, 2015.
- [46] Kai Pfeiffer, Adrien Escande, and Abderrahmane Kheddar. Singularity resolution in equality and inequality constrained hierarchical task-space control by adaptive nonlinear least squares. *IEEE Robotics and Automation Letters*, 3(4):3630–3637, 2018.
- [47] Mathieu Geisert, Andrea Del Prete, Nicolas Mansard, Francesco Romano, and Francesco Nori. Regularized hierarchical differential dynamic programming. *IEEE Transactions on Robotics*, 33(4):819–833, 2017.
- [48] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913. IEEE, 2012.
- [49] Yuval Tassa, Nicolas Mansard, and Emanuel Todorov. Control-limited differential dynamic programming. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1168–1175. IEEE, 2014.
- [50] Jan Peters and Stefan Schaal. Learning to control in operational space. *The International Journal of Robotics Research*, 27(2):197–212, 2008.
- [51] Noémie Jaquier, Leonel Rozo, Darwin G Caldwell, and Sylvain Calinon. Geometry-aware manipulability learning, tracking, and transfer. *The International Journal of Robotics Research*, pages 624–650, 2018.

- [52] Donghyeon Lee, Woongyong Lee, Jonghoon Park, and Wan Kyun Chung. Task space control of articulated robot near kinematic singularity: Forward dynamics approach. *IEEE Robotics and Automation Letters*, 5(2):752–759, 2020.
- [53] Min Jun Kim, Youngjin Choi, and Wan Kyun Chung. Bringing nonlinear H_∞ optimality to robot controllers. *IEEE Transactions on Robotics*, 31(3):682–698, 2015.
- [54] Jie Tan, Karen Liu, and Greg Turk. Stable proportional-derivative controllers. *IEEE Computer Graphics and Applications*, 31(4):34–44, 2011.
- [55] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.
- [56] David G Luenberger and Yinyu Ye. *Linear and nonlinear programming*. *International series in operations research & management science*, volume 10. Springer, 2008.
- [57] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [58] I. Goodfellow, Y. Benjio, and A. Courville. *Deep learning*. The MIT Press, 2016.
- [59] Taeyoon Lee and Frank C Park. A geometric algorithm for robust multi-body inertial parameter identification. *IEEE Robotics and Automation Letters*, 3(3):2455–2462, 2018.

- [60] Taeyoon Lee, Patrick M Wensing, and Frank C Park. Geometric robot dynamic identification: A convex programming approach. *IEEE Transactions on Robotics*, 36(2):348–365, 2019.
- [61] Samuel R Buss and Jin-Su Kim. Selectively damped least squares for inverse kinematics. *Journal of Graphics tools*, 10(3):37–49, 2005.
- [62] Marc G Carmichael, Dikai Liu, and Kenneth J Waldron. A framework for singularity-robust manipulator control during physical human-robot interaction. *The International Journal of Robotics Research*, 36(5-7):861–876, 2017.
- [63] Adria Colomé and Carme Torras. Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements. *IEEE/ASME Transactions on Mechatronics*, 20(2):944–955, 2015.
- [64] Daniele Di Vito, Ciro Natale, and Gianluca Antonelli. A comparison of damped least squares algorithms for inverse kinematics of robot manipulators. *IFAC-PapersOnLine*, 50(1):6869–6874, 2017.
- [65] Tomomichi Sugihara. Solvability-unconcerned inverse kinematics by the levenberg–marquardt method. *IEEE Transactions on Robotics*, 27(5):984–991, 2011.
- [66] Tomomichi Sugihara. Robust solution of prioritized inverse kinematics based on hestenes-powell multiplier method. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 510–515. IEEE, 2014.
- [67] Paolo Baerlocher and Ronan Boulic. An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The visual computer*, 20(6):402–417, 2004.

- [68] Stefano Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.
- [69] Fabrizio Flacco, Alessandro De Luca, and Oussama Khatib. Prioritized multi-task motion control of redundant robots under hard joint constraints. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3970–3977. IEEE, 2012.
- [70] A Escande, N Mansard, and PB Wieber. Hierarchical quadratic programming (part 1). Technical report, LAAS-CNRS, 2012.
- [71] Layale Saab, Oscar E Ramos, François Keith, Nicolas Mansard, Philippe Soueres, and Jean-Yves Fourquet. Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Transactions on Robotics*, 29(2):346–362, 2013.
- [72] Pierre-Brice Wieber, Russ Tedrake, and Scott Kuindersma. Modeling and control of legged robots. In *Springer handbook of robotics*, pages 1203–1234. Springer, 2016.
- [73] G Clark Haynes, David Stager, Anthony Stentz, J Michael Vande Weghe, Brian Zajac, Herman Herman, Alonzo Kelly, Eric Meyhofer, Dean Anderson, Dane Bennington, et al. Developing a robust disaster response robot: Chimp and the robotics challenge. *Journal of Field Robotics*, 34(2):281–304, 2017.
- [74] Sisir Karumanchi, Kyle Edelberg, Ian Baldwin, Jeremy Nash, Jason Reid, Charles Bergh, John Leichty, Kalind Carpenter, Matthew Shekels, Matthew Gildner, et al. Team robosimian: semi-autonomous mobile manipulation

- at the 2015 darpa robotics challenge finals. *Journal of Field Robotics*, 34(2):305–332, 2017.
- [75] Taejin Jung, Jeongsoo Lim, Hyoin Bae, Kang Kyu Lee, Hyun-Min Joe, and Jun-Ho Oh. Development of the humanoid disaster response platform drc-hubo+. *IEEE Transactions on Robotics*, 34(1):1–17, 2018.
- [76] Jeongsoo Lim, Inho Lee, Inwook Shim, Hyobin Jung, Hyun Min Joe, Hyoin Bae, Okkee Sim, Jaesung Oh, Taejin Jung, Seunghak Shin, et al. Robot system of drc-hubo+ and control strategy of team kaist in darpa robotics challenge finals. *Journal of Field Robotics*, 34(4):802–829, 2017.
- [77] Max Schwarz, Tobias Rodehuts Kors, David Droschel, Marius Beul, Michael Schreiber, Nikita Araslanov, Ivan Ivanov, Christian Lenz, Jan Razlaw, Sebastian Schüller, et al. Nimbro rescue: Solving disaster-response tasks with the mobile manipulation robot momaro. *Journal of Field Robotics*, 34(2):400–425, 2017.
- [78] Vecna Robotics. The bear battlefield extraction-assist robot, 2015.
- [79] Jienan Ding, Yi-Je Lim, Mario Solano, Kevin Shadle, Chris Park, Chris Lin, and John Hu. Giving patients a lift-the robotic nursing assistant (rona). In *2014 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pages 1–5. IEEE, 2014.
- [80] John Hu, Aaron Edsinger, Yi-Je Lim, Nick Donaldson, Mario Solano, Aaron Solocheck, and Ronald Marchessault. An advanced medical robotic system augmenting healthcare capabilities-robotic nursing assistant. In *2011 IEEE international conference on robotics and automation*, pages 6264–6269. IEEE, 2011.

- [81] Toshiharu Mukai, Shinya Hirano, Hiromichi Nakashima, Yo Kato, Yuki Sakaida, Shijie Guo, and Shigeyuki Hosoe. Development of a nursing-care assistant robot riba that can lift a human in its arms. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5996–6001. IEEE, 2010.
- [82] Christopher Mario Maes. *A regularized active-set method for sparse convex quadratic programming*. PhD thesis, Stanford University, 2011.
- [83] Elizabeth Lai Sum Wong. *Active-set methods for quadratic programming*. PhD thesis, UC San Diego, 2011.
- [84] Christian Kirches, Andreas Potschka, Hans Georg Bock, and Sebastian Sager. A parametric active set method for quadratic programs with vanishing constraints. *Technical Report*, 2012.
- [85] Luis Sentis and Oussama Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(04):505–518, 2005.
- [86] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.
- [87] Alexander Dietrich and Christian Ott. Hierarchical impedance-based tracking control of kinematically redundant robots. *IEEE Transactions on Robotics*, 36(1):204–221, 2019.
- [88] Seongil Hong, Gyuhyun Park, Wonsuk Lee, and Sincheon Kang. Realization of the fast real-time hierarchical inverse kinematics for dexterous full body

- manipulation. In *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, pages 166–171. IEEE, 2018.
- [89] Adi Ben-Israel and Thomas NE Greville. *Generalized inverses: theory and applications*. Springer, 2003.
- [90] Stefano Chiaverini, Giuseppe Oriolo, and Anthony A Maciejewski. Redundant robots. In *Springer Handbook of Robotics*, pages 221–242. Springer, 2016.
- [91] Yoshihiko Nakamura. *Advanced robotics: redundancy and optimization*. Addison-Wesley Longman Publishing Co., Inc., 1990.
- [92] Gilbert Strang. *Introduction to linear algebra*. Wellesley-Cambridge Press, 2016.
- [93] Zhijun Zhang and Yunong Zhang. Variable joint-velocity limits of redundant robot manipulators handled by quadratic programming. *IEEE/ASME Transactions on Mechatronics*, 18(2):674–686, 2012.
- [94] Bruno Siciliano. A closed-loop inverse kinematic scheme for on-line joint-based robot control. *Robotica*, 8(3):231–243, 1990.
- [95] Pasquale Chiacchio, Stefano Chiaverini, Lorenzo Sciavicco, and Bruno Siciliano. Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *The International Journal of Robotics Research*, 10(4):410–425, 1991.

- [96] Seongil Hong, Youngwoo Lee, Kyu Hyun Park, Won Suk Lee, Byunghun Choi, Okkee Sim, Inhyeok Kim, Jun-Ho Oh, and Youn Sik Kang. Dynamics based motion optimization and operational space control with an experimental rescue robot, hubo t-100. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 773–778. IEEE, 2015.
- [97] Seongil Hong, Gyuhyun Park, Youngwoo Lee, Wonsuk Lee, Byunghun Choi, Okkee Sim, and Jun-Ho Oh. Development of a tele-operated rescue robot for a disaster response. *International Journal of Humanoid Robotics*, 15(4):1–30, 2018.
- [98] Youngjin Choi, Wan Kyun Chung, and Il Hong Suh. Performance and H_∞ optimality of pid trajectory tracking controller for lagrangian systems. *IEEE Transactions on Robotics and Automation*, 17(6):857–869, 2001.
- [99] Jonghoon Park and Wan Kyun Chung. Design of a robust H_∞ pid control for industrial manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 122(4):803–812, 2000.
- [100] Seongil Hong, Won Suk Lee, Youn Sik Kang, and Yong Woon Park. Kinematic control algorithms and robust controller design for rescue robot. In *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, pages 637–642. IEEE, 2014.
- [101] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*,, pages 2149–2154, 2004.
- [102] Youngwoo Lee, Wonsuk Lee, Byunghun Choi, Gyuhyun Park, and Yongwoon Park. Reliable software architecture design with ethercat for a rescue robot.

- In *2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, pages 34–39. IEEE, 2016.
- [103] Gaël Guennebaud, Benoit Jacob, et al. Eigen. URL: <http://eigen.tuxfamily.org>, 2010.
- [104] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics—Modelling, Planning and Control*. London, UK: Springer-Verlag, 2009.
- [105] Kenneth Kreutz-Delgado, M Long, and Homayoun Seraji. Kinematic analysis of 7 dof anthropomorphic arms. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 824–830, 1990.
- [106] Andrea Maria Zanchettin, Luca Bascetta, and Paolo Rocco. Achieving humanlike motion: Resolving redundancy for anthropomorphic industrial manipulators. *IEEE Robotics & Automation Magazine*, 20(4):131–138, 2013.
- [107] Christian Smith, Yiannis Karayiannidis, Lazaros Nalpantidis, Xavi Gratal, Peng Qi, Dimos V Dimarogonas, and Danica Kragic. Dual arm manipulation—a survey. *Robotics and Autonomous systems*, 60(10):1340–1353, 2012.
- [108] Nicolas Mansard, Oussama Khatib, and Abderrahmane Kheddar. A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Transactions on Robotics*, 25(3):670–685, 2009.
- [109] Sanghyun Kim, Keunwoo Jang, Suhan Park, Yisoo Lee, Sang Yup Lee, and Jaeheung Park. Continuous task transition approach for robot controller

- based on hierarchical quadratic programming. *IEEE Robotics and Automation Letters*, 4(2):1603–1610, 2019.
- [110] Keunwoo Jang, Sanghyun Kim, and Jaeheung Park. Reactive self-collision avoidance for a differentially driven mobile manipulator. *Sensors*, 21(3):890, 2021.
- [111] Karim Bouyarmane, Kevin Chappellet, Joris Vaillant, and Abderrahmane Kheddar. Quadratic programming for multirobot and task-space force control. *IEEE Transactions on Robotics*, 35(1):64–77, 2018.
- [112] Fabrizio Caccavale, Ciro Natale, Bruno Siciliano, and Luigi Villani. Integration for the next generation: embedding force control into industrial robots. *IEEE Robotics & Automation Magazine*, 12(3):53–64, 2005.
- [113] Kevin M Lynch and Frank C Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.
- [114] Luigi Villani and Joris De Schutter. Force control. In *Springer handbook of robotics*, pages 195–220. Springer, 2016.
- [115] Stefano Chiaverini, Bruno Siciliano, and Luigi Villani. A survey of robot interaction control schemes with experimental comparison. *IEEE/ASME Transactions on mechatronics*, 4(3):273–285, 1999.
- [116] Andrea Calanca, Riccardo Muradore, and Paolo Fiorini. A review of algorithms for compliant control of stiff and fixed-compliance robots. *IEEE/ASME Transactions on Mechatronics*, 21(2):613–624, 2015.
- [117] Jaeheung Park. *Control strategies for robots in contact*. PhD thesis, Stanford University, 2006.

- [118] Joris De Schutter and Hendrik Van Brussel. Compliant robot motion ii. a control approach based on external control loops. *The International Journal of Robotics Research*, 7(4):18–33, 1988.
- [119] Jaydeep Roy and Louis L Whitcomb. Adaptive force control of position/velocity controlled robots: theory and experiment. *IEEE Transactions on Robotics and Automation*, 18(2):121–137, 2002.
- [120] Wen Yu, Jacob Rosen, and Xiaou Li. Pid admittance control for an upper limb exoskeleton. In *Proceedings of the 2011 American control conference*, pages 1124–1129. IEEE, 2011.
- [121] Matteo Parigi Polverini, Simone Formentin, Luca Merzagora, and Paolo Rocco. Mixed data-driven and model-based robot implicit force control: A hierarchical approach. *IEEE Transactions on Control Systems Technology*, 28(4):1258–1271, 2019.
- [122] Arne Wahrburg and Kim Listmann. Mpc-based admittance control for robotic manipulators. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7548–7554. IEEE, 2016.
- [123] Luigi Villani, Ciro Natale, Bruno Siciliano, and Carlos Canudas de Wit. An experimental study of adaptive force/position control algorithms for an industrial robot. *IEEE Transactions on Control Systems Technology*, 8(5):777–786, 2000.
- [124] Ciro Natale, Ralf Koeppe, and Gerd Hirzinger. A systematic design procedure of force controllers for industrial robots. *IEEE/ASME transactions on mechatronics*, 5(2):122–131, 2000.

- [125] Rodrigo Pérez-Ubeda, Ranko Zotovic-Stanisis, and Santiago C Gutiérrez. Force control improvement in collaborative robots through theory analysis and experimental endorsement. *Applied Sciences*, 10(12):4329, 2020.
- [126] Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.
- [127] Oscar Efran Ramos Ponce. Generation of complex dynamic motion for a humanoid robot. Master’s thesis, LAAS-CNRS, FR, 2011.
- [128] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. springer, 2016.
- [129] Christian Ott, Ranjan Mukherjee, and Yoshihiko Nakamura. Unified impedance and admittance control. In *2010 IEEE international conference on robotics and automation*, pages 554–561. IEEE, 2010.
- [130] Yuquan Wang, Arnaud Tanguy, Pierre Gergondet, and Abderrahmane Kheddar. Impact-aware humanoid robot motion generation with a quadratic optimization controller. *arXiv preprint arXiv:2001.08454*, 2020.
- [131] Silvio Traversaro, Stanislas Brossette, Adrien Escande, and Francesco Nori. Identification of fully physical consistent inertial parameters using optimization on manifolds. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5446–5451. IEEE, 2016.
- [132] Alexander Dietrich, Christian Ott, and Stefano Stramigioli. Passivation of projection-based null space compliance control via energy tanks. *IEEE Robotics and automation letters*, 1(1):184–191, 2015.

- [133] Gianluca Garofalo and Christian Ott. Hierarchical tracking control with arbitrary task dimensions: Application to trajectory tracking on submanifolds. *IEEE Robotics and Automation Letters*, 5(4):6153–6160, 2020.
- [134] Manuel Keppeler, Dominic Lakatos, Christian Ott, and Alin Albu-Schäffer. Elastic structure preserving (esp) control for compliantly actuated robots. *IEEE Transactions on Robotics*, 34(2):317–335, 2018.
- [135] Siyuan Feng, Eric Whitman, X Xinjilefu, and Christopher G Atkeson. Optimization-based full body control for the darpa robotics challenge. *Journal of Field Robotics*, 32(2):293–312, 2015.
- [136] Stanislas Brossette, Adrien Escande, and Abderrahmane Kheddar. Multi-contact postures computation on manifolds. *IEEE Transactions on Robotics*, 34(5):1252–1265, 2018.
- [137] Russ Tedrake. Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation (course notes for mit 6.832), 2021.
- [138] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 40(3):429–455, 2016.
- [139] Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, 2018.

- [140] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016.
- [141] John T Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.
- [142] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [143] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- [144] Patrick M Wensing, Sangbae Kim, and Jean-Jacques E Slotine. Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution. *IEEE Robotics and Automation Letters*, 3(1):60–67, 2017.
- [145] Sergey Levine and Vladlen Koltun. Continuous inverse optimal control with locally optimal examples. *arXiv preprint arXiv:1206.4617*, 2012.
- [146] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

- [147] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007.
- [148] Ioannis Havoutis and Subramanian Ramamoorthy. Motion planning and reactive control on learnt skill manifolds. *The International Journal of Robotics Research*, 32(9-10):1120–1150, 2013.
- [149] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y Ng. An application of reinforcement learning to aerobatic helicopter flight. *Advances in neural information processing systems*, 19:1, 2007.
- [150] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [151] Dimitri P Bertsekas. *Dynamic programming and optimal control: Vol. 1*. Athena scientific Belmont, 2000.
- [152] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.
- [153] Emanuel Todorov and Michael I Jordan. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226–1235, 2002.
- [154] Sergey Levine. *Motor skill learning with local trajectory methods*. Stanford University, 2014.
- [155] Benjamin Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:253–279, 2019.

- [156] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, pages 1–47, 2019.
- [157] Felix Berkenkamp. *Safe Exploration in Reinforcement Learning: Theory and Applications in Robotics*. PhD thesis, ETH Zurich, 2019.
- [158] Hilbert J Kappen. Linear theory for control of nonlinear stochastic systems. *Physical review letters*, 95(20):200201, 2005.
- [159] Russ Tedrake, Ian R Manchester, Mark Tobenkin, and John W Roberts. Lqr-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8):1038–1052, 2010.
- [160] Ugo Rosolia, Xiaojing Zhang, and Francesco Borrelli. Data-driven predictive control for autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:259–286, 2018.
- [161] Karl Johan Åström and PR Kumar. Control: A perspective. *Automatica*, 50(1):3–43, 2014.
- [162] Kiam Heong Ang, Gregory Chong, and Yun Li. Pid control system analysis, design, and technology. *IEEE transactions on control systems technology*, 13(4):559–576, 2005.
- [163] Karl Johan Åström and Tore Hägglund. *PID controllers: theory, design, and tuning*, volume 2. Instrument society of America Research Triangle Park, NC, 1995.

- [164] Karl Johan Åström, Tore Hägglund, Chang C Hang, and Weng Kuen Ho. Automatic tuning and adaptation for pid controllers—a survey. In *Adaptive Systems in Control and Signal Processing 1992*, pages 371–376. Elsevier, 1993.
- [165] Nick J Killingsworth and Miroslav Krstic. Pid tuning using extremum seeking: online, model-free performance optimization. *IEEE control systems*, 26(1):70–79, 2006.
- [166] Katsuhiko Ogata. *Modern control engineering*. Prentice hall, 2010.
- [167] John C Doyle, Bruce A Francis, and Allen R Tannenbaum. *Feedback control theory*. Courier Corporation, 2013.
- [168] Huibert Kwakernaak. Robust control and H_∞ optimization—tutorial paper. *Automatica*, 29(2):255–273, 1993.
- [169] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*, volume 2. Wiley New York, 2007.
- [170] Seongil Hong and Ki Dae Cho. Kinematic algorithms and robust controller design for inertially stabilized system. *IEEE/ASME Transactions on Mechatronics*, 19(1):76–87, 2012.
- [171] Vijay Sekhar Chellaboina and Wassim M Haddad. *Nonlinear dynamical systems and control: A Lyapunov-based approach*. Princeton University Press, 2008.
- [172] Alberto Isidori and Alessandro Astolfi. Disturbance attenuation and H_∞ -control via measurement feedback in nonlinear systems. *IEEE transactions on automatic control*, 37(9):1283–1293, 1992.

- [173] Eduardo D Sontag and Yuan Wang. On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24(5):351–359, 1995.
- [174] Arjan J Van Der Schaft. L_2 -gain analysis of nonlinear systems and nonlinear state-feedback H_∞ control. *IEEE transactions on automatic control*, 37(6):770–784, 1992.
- [175] Brian DO Anderson and John B Moore. *Optimal control: linear quadratic methods*. Courier Corporation, 2007.

국문초록

최근에 등장한 새로운 세대의 로봇은 기존에는 인간만이 할 수 있었던 복잡한 일을 로봇 또한 수행할 수 있음을 보여주었다. 특히 DARPA Robotics Challenge를 통해 이러한 사실을 잘 확인할 수 있으며, 이 로봇들은 공장과 같은 정형화된 환경에서 자동화된 일을 반복적으로 수행하던 임무에서 더 나아가 극한의 환경에서 인간을 대신하여 위험한 임무를 수행할 수 있는 방향으로 발전하고 있다. 그래서 사람들은 재난환경에서 안전하고 시의 적절하게 대응할 수 있는 여러 가지 대안 중에서 실현 가능성이 높은 대처 방안으로 로봇을 생각하게 되었다. 하지만 이러한 로봇은 동적으로 변화하는 비정형 환경에서 임무를 수행할 수 있어야 하기 때문에 불확실성에 대해 강건해야하고, 다양한 환경 조건에서 능동적으로 반응을 할 수 있어야 한다. 본 학위논문에서는 로봇이 비정형 환경에서 강건하면서도 적응적으로 동작할 수 있는 실시간 최적화 기반의 동작 계획 및 제어 방법과 구조 로봇의 설계 개념을 제안하고자 한다.

인간은 많은 자유도를 가지고 있으며, 하나의 전신 동작을 생성할 때 다양한 기구학 혹은 동역학적 특성을 가지는 세부 동작 혹은 작업을 정의하고, 이를 효과적으로 종합할 수 있다. 그리고 학습을 통해 각 동작 요소들을 최적화할 뿐만 아니라 상황에 따라 각 동작 요소에 우선순위를 부여하여 이를 효과적으로 결합하거나 분리하여 실시간으로 최적의 동작을 생성하고 제어한다. 즉, 상황에 따라 중요한 동작요소를 우선적으로 수행하고 우선순위가 낮은 동작요소는 부분 혹은 전체적으로 포기하기도 하면서 매우 유연하게 전체 동작을 생성하고 최적화한다.

인간과 같이 다자유도를 보유한 로봇 또한 기구학과 동역학적 특성을 가지는 다양한 세부 동작 혹은 작업을 작업공간(task space) 혹은 관절공간(configuration space)에서 정의할 수 있으며, 우선순위에 따라 이를 효과적으로 결합하여 전체 동작을 생성하고 제어할 수 있다. 서로 양립하기 어려운 로봇의 동작 문제를 해결하기 위해 동작들 사이에 우선순위를 부여하여 계층을 생성하고, 이에 따라 로봇의 전신 동작을 구현하는 방법은 오랫동안 연구가 진행되어 왔다. 이러한 계층적 최적화를 이용하면

우선순위가 높은 동작부터 순차적으로 실행하지만, 우선순위가 낮은 동작요소들도 가능한 만족시키는 최적의 해를 찾을 수 있다.

하지만 관절의 구동 범위와 같은 부등식의 조건이 포함된 계층적 최적화 문제에서 특이점에 대한 강건성까지 확보할 수 있는 방법에 대해서는 아직까지 많은 부분이 밝혀진 바가 없다. 따라서 본 학위논문에서는 등식과 부등식으로 표현되는 구속조건 혹은 동작요소를 계층적 최적화에 동시에 포함시키고, 특이점이 존재하더라도 강건성과 수렴성을 보장하는 관절공간에서의 최적해를 확보하는데 집중한다. 왜냐하면 비정형 임무를 수행하는 로봇은 사전에 계획된 동작을 수행하는 것이 아닌 변화하는 환경조건에 따라 실시간으로 동작을 계획하고 제어해야 하기 때문에 특이점이 없는 자세로 로봇을 항상 제어하기가 어렵다. 그리고 이렇게 특이점을 회피하는 방향으로 로봇을 제어하는 것은 로봇의 운용성을 심각하게 저해시킬 수 있다. 특이점 근방에서의 해의 강건성이 보장되지 않으면 로봇 관절에 과도한 속도 혹은 토크가 발생하여 로봇의 임무 수행이 불가능하거나 환경과 로봇의 손상을 초래할 수 있으며, 나아가 로봇과 함께 임무를 수행하는 사람에게 상해를 가할 수도 있다.

특이점에 대한 강건성을 확보하기 위해 우선순위 기반의 계층적 최적화와 정규화(regularization)를 통합하여 정규화된 계층적 최적화 (RHQP: Regularized Hierarchical Quadratic Program) 문제를 다룬다. 부등식이 포함된 계층적 최적화에 정규화를 동시에 고려함으로써 야기되는 많은 문제점들을 해결하고 해의 최적성과 강건성을 확보할 수 있는 방법을 제안한다. 특히 외부의 최적화 프로그램을 사용하지 않고 수치적 최적화(numerical optimization) 이론과 우선순위에 기반을 두는 여유자유도 로봇의 해석 기법을 이용하여 계산의 효율성을 극대화할 수 있는 이차 프로그램(quadratic programming)을 제안한다. 또한 이와 동시에 정규화된 계층적 최적화 문제의 이론적 구조를 철저하게 분석한다. 특히 특이값 분해(singular value decomposition)를 통해 최적해와 부등식 조건을 처리하는데 필요한 라그랑지 승수를 재귀적인 방법으로 해석적 형태로 구함으로써 계산의 효율성을 증대시키고 동시에 부등식의 조건을 오류 없이 정확하게 처리할 수 있도록 하였다. 그리고 정규화된 계층적 최적화를 힘

제어까지 확장하여 환경과 로봇의 안전한 상호작용을 보장하여 로봇이 적절한 힘으로 환경과 접촉할 수 있도록 하였다.

불확실성이 존재하는 비정형 환경에서 비정형 임무를 수행할 수 있는 구조로봇의 핵심 설계 개념을 제시한다. 비정형 환경에서의 조작 성능과 이동 성능을 동시에 확보할 수 있는 형상으로 로봇을 설계하여 구조 로봇으로 하여금 최종 목적으로 설정된 인간을 대신하여 부상자를 구조하고 위험물을 처리하는 임무를 효과적으로 수행할 수 있도록 한다. 구조 로봇에 필요한 매니플레이터는 부상자 구조 임무와 위험물 처리 임무에 따라 교체 가능한 모듈형으로 설계하여 각각의 임무에 따라 최적화된 매니플레이터를 장착하여 임무를 수행할 수 있다. 하체는 트랙과 관절이 결합된 하이브리드 형태를 취하고 있으며, 주행 임무와 조작임무에 따라 형상을 변경할 수 있다. 형상 변경과 모듈화된 매니플레이터를 통해서 조작 성능과 험한 지형에서 이동할 수 있는 주행 성능을 동시에 확보하였다.

최종적으로 구조로봇의 설계와 실시간 계층적 제어를 이용하여 비정형 실내외 환경에서 구조로봇이 주행임무, 위험물 조작임무, 부상자 구조 임무를 성공적으로 수행할 수 있음을 해석과 실험을 통하여 입증함으로써 본 학위논문에서 제안한 설계와 정규화된 계층적 최적화 기반의 제어 전략의 유용성을 검증하였다.

주요어: 구조로봇, 작업 우선순위, 정규화, 최소제곱 최적화, 특이점 강건성, 부등식의 구속조건.

학번: 2016-30181