POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Biomedica
Dipartimento di Elettronica, Informazione e Bioingegneria

JOINT LIMITS HANDLING UNDER KINEMATIC AND CARTESIAN CONSTRAINTS FOR MEDICAL ROBOTICS

Laboratorio di Neuroingegneria e Robotica Medica

Relatore:
Prof. Elena De Momi
Supervisors:
Dr. Ewald Lutscher, Dr. Thomas Neff

Tesi di Laurea di
Andrea Spinoglio, Matr. 853579

Anno Accademico 2017-2018

Embargo notice:

This final project contains confidential data from KUKA AG and all associated companies according to Section 15 ff of the German Stock Corporation Act. At the request of KUKA AG and all associated companies according to Section 15 ff of the German Stock Corporation Act, this final project shall be embargoed from public use for a period of 3 years.

Contents

List of Figures

List of Tables

Abstract

Negli ultimi anni la collaborazione tra robot e umani sta prendendo piede in diversi ambiti tra cui quello clinico: il numero di interventi assistiti e la vendita di sistemi robotici stanno aumentando notevolmente. I robot potrebbero garantire lo sviluppo o l'affinamento di nuove tecniche chirurgiche riducendone l'invasività, migliorandone l'accuratezza ed eliminando gli effetti creati da fatica o tremori.
Il lavoro qui presentato ha contribuito allo sviluppo di un progetto iniziato negli anni passati che mira a portare nelle sale operatorie un esemplare di LBR Med guidato manualmente come supporto ai chirurghi durante gli interventi di chirurgia spinale. Il progetto è stato interamente svolto durante un periodo di sei mesi presso la sede centrale della KUKA AG in Augsburg. Scopo finale di questa tesi è quello di presentare un nuovo algoritmo, per il controllo dei limiti di giunto, sviluppato per far fronte a tutti i requisiti medici richiesti dal progetto. I due principali requisiti da soddisfare sono stati:
- la possibilità di muoversi attorno ai limiti di giunto anche una volta raggiunto un limite andando a modificare e scalare il comando imposto dalla spinta dell'utilizzatore ma senza mai produrre alcun movimento inaspettato
- implementare vincoli cinematici sui giunti andando a imporre limiti di velocità e accelerazione assoluti e relativi all'attuale posizione, con lo scopo di impedire stop improvvisi del robot
La volontà di poter assicurare in ambito clinico specifiche modalità di movimento ha imposto la necessità di poter definire anche vincoli cartesiani al movimento del robot, i quali devono essere rigidamente rispettati in ogni momento. Nel'ambito specifico della chirurgia spinale è sicuramente richiesto, durante il posizionamento degli impianti spinali, un movimento limitato su di una retta per performare nel modo più lineare possibile le azioni di perforazione.

Lo sviluppo dell'algoritmo è descritto passo a passo in tutte le sue parti. La soluzione finale proposta prevede la risoluzione di un problema di ottimizzazione vincolato: tutti i vincoli cinematici imposti sono definiti da opportune matrici e la funzione da minimizzare è rappresentata dal quadrato dell'errore introdotto nella velocità cartesiana imposta al robot.
L'algoritmo è stato infine confrontato nella fase di test con la soluzione precedentemente implementata sul LBR Med e con la più diffusa soluzione reperibile in letteratura, per essa sono stati reclutati 42 soggetti, tra gli impiegati dell'azienda, ai quali è stato chiesto di compiere un semplice compito. I tre metodi sono stati confrontati tramite l'analisi di dati quantitativi riguardanti lo stress imposto all'hardware durante le esecuzioni e dati qualitativi riguardanti l'usabilità percepita dai partecipanti raccolti tramite un breve questionario. In entrambi casi la soluzione proposta è risultata significativamente migliore.

Abstract

In recent years, the collaboration between robots and humans is taking place in various areas including the clinical field: the number of assisted interventions and the sale of robotic systems have been considerably increased. Robots could guarantee the development or refinement of new surgical techniques, reducing their invasiveness, improving their accuracy and eliminating the effects created by fatigue or tremors.

The work presented has contributed to the development of a project begun in the past years that aims to bring a sample of the hand-guided LBR Med into operating rooms as a support for surgeons during spinal surgery. The project was entirely carried out during a period of six months at the KUKA AG headquarters in Augsburg.

The final aim of this thesis is to present a new algorithm, for joint limits control, developed to fulfil all the medical requirements stated by the project. The two main requirements to be met were:

- the ability to slide around joint limits even when a position limit has been reached by modifying and scaling the command imposed by the user's hand but without producing any unexpected movement
- implementing kinematic constraints to impose velocity and acceleration bounds both absolute and relative to the current position, with the aim of preventing sudden stops of the robot

The desire to be able to provide specific motion modalities for the clinical use has imposed the need to define Cartesian constraints to the motion, which must be strictly respected at all times. In the specific field of spinal surgery, during the positioning of the spinal implants, a limited movement on a straight line is surely required to perform the drilling actions in the most accurate linear way.

The development of the algorithm is described step by step. The proposed final solution involves the resolution of a constrained optimization problem: all the imposed kinematic constraints are defined by appropriate matrices

and the objective function to be minimized is represented by the square form of the error introduced in the Cartesian velocity imposed on the robot.
The algorithm was finally compared in the test phase with the solution previously implemented on the LBR Med and with the most widespread solution available in literature , for this purpose 42 subjects were recruited, among the employees of the company, and they were asked to perform a simple task. The three methods were compared through the analysis of quantitative data concerning the stress imposed on the hardware during the performances and qualitative data regarding the usability perceived by the participants collected through a short questionnaire. In both cases the proposed solution was significantly better.

Chapter 1

Medical Background

1.1 Spine surgery

Spine surgery is a delicate and hazardous procedure due to its proximity to both the central nervous system and main blood vessels.

Spine surgery may be recommended if non-surgical treatment such as medications and physical therapy fails to relieve symptoms. Anyway it is only considered in cases where the exact source of pain can be determined.

1.2 Common spine disorders and diseases

Deformities
The most common in both child and adults is the scoliosis, it is an abnormal curving of the spine that can happen at different levels. Kyphosis and lordosis are two other common deformities, in the respective case the natural kyphotic and lordotic curvature of the spine is overemphasize.

Disc disease
Degenerative disc disease describes the natural breakdown of an intervertebral disc of the spine. The disc can gradually weaken and lose water for daily stress or minor injuries. The collapsing of the disc can cause increased

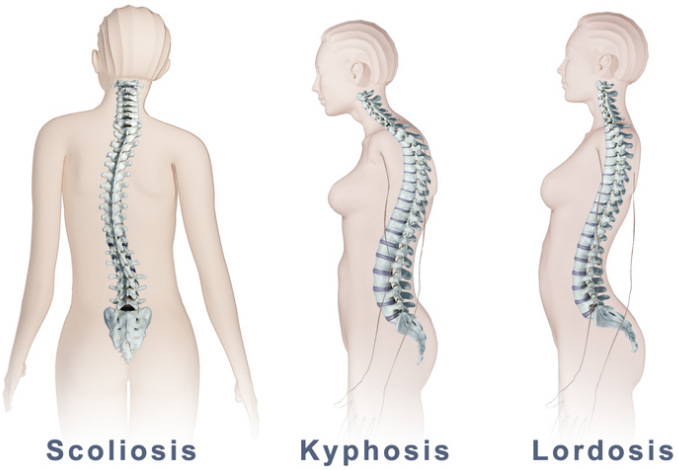


Figure 1.1: Spinal Deformities [1]

pressure on the nerves and consequent pain.

Spondylolosthesis
It is an anterior displacements of one vertebra over another, commonly involving the fifth lumbar vertebra. It can have different origins:

- degenerative
- traumatic
- pathological
- congenital

Stenosis
The term stenosis defines an abnormal narrowing of the bone spinal canal. This condition is usually related to aging and can cause chronic back pain due to high pressures on the nerves.

Instability
Segmental instability occurs when the relative motion is grater than normal (hypermobility) between two adjacent vertebrae. This instability can be caused by disc degeneration or spondylolosthesis.

Spine tumors
A surgical intervention can be necessary both for benign and malignant tumors. The growth can lead to further problems like spine deformities.

1.3 Common surgical procedures

Discectomy or Microdiscectomy
Removal of a herniated intervertebral disc. Therefore, removing pressure from the compressed nerve. Microdiscectomy is a MISS procedure.

Laminectomy
Removal of the thin bony plate on the back of the vertebra called the laminae to increase space within the spinal canal and relieve pressure.

Laminotomy
Removal of a portion of the vertebral arch (lamina) that covers the spinal cord. A laminotomy removes less bone than a laminectomy.

Both laminectomy and laminotomy are decompression procedures. "Decompression" usually means tissue compressing a spinal nerve is removed.

Foraminotomy
Removal of bone or tissue at/in the canal (called the neuroforamen) where nerve roots branch off the spinal cord and exit the spinal column.

Disc replacement
As an alternative to fusion, the injured disc is replaced with an artificial one.

Spinal fusion
Spinal fusion is a very common procedure and it is performed for a multitude of cases: fracture of vertebral body, degenerative disc disease,, spine tumors, scoliosis. By disabling the relative motion between two adjacent vertebrae is possible to prevent compression during movements.

The procedure involves the insertion of two screws per fused vertebra on the left and right spinal pedicles. The screws insertion angle is important to avoid perforation of the pedicles or damage to the spinal cord. A pilot hole for each screws is drilled by the surgeons. Once in place, the pedicle screws are fitted with a screw head and then a rod is placed to connect these two heads thus forming a unique rigid body.
For this type of procedures a robotic assisted surgery can always be recommended based on the patient physical status and on the planning phase.

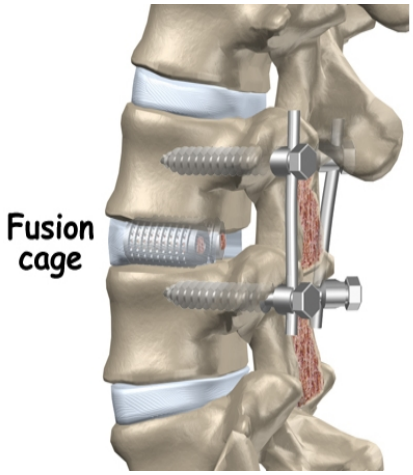


Figure 1.2: Spinal fusion [2]

1.4 Approaches

Spine surgery has experienced much technological innovation over the past several decades. The field has seen advancements in operative techniques, implants and biologics, and equipment such as computer-assisted navigation (CAN) and surgical robotics. With the arrival of real-time image guidance and navigation capabilities along with the computing ability to process and reconstruct these data into an interactive three-dimensional (3-D) spinal map so too have the applications of surgical robotic technology advanced. While spinal robotics represents promising potential for improving modern spinal

surgery, it remains to demonstrate its superiority as compared to traditional techniques prior to assimilation of its use amongst surgeons. The same is true of intraoperative navigation techniques, which have shown reliability in performance across many other surgical subspecialties, as well as success across several studies in their application for improved pedicle screw accuracy.

Spine surgery relies upon meticulous fine motor skills to manipulate neural elements and a steady hand while doing so, often exploiting small working corridors utilizing exposures that minimize collateral damage. Additionally, the procedures may be long and arduous, predisposing the surgeon to both mental and physical fatigue. In light of these characteristics, spine surgery may actually be an ideal candidate for the integration of navigation and robotic-assisted procedures. These platforms have been shown to dramatically improve a surgeon's manual dexterity allowing for greater control and manoeuvrability through a less invasive working portal, while dampening a surgeon's physiological tremor. By definition, robots do not fatigue and are capable of performing repetitive tasks with accuracy and precision, yielding infinitely reproducible outcomes.

Robotic-assisted surgery has been used for years by other surgical subspecialties. The robots come in a variety of designs with varying levels of "assistance" that can be broken down into 3 broad categories:

1. supervisory-controlled systems whereby the machine is programmed with predetermined actions that are carried out with robotic autonomy and close surgeon supervision

2. telesurgical systems, like the Da Vinci robot (Intuitive Surgical, Sunnyvale, California), that afford the surgeon complete control of the motions of the machine from a remote command station

3. shared-control models, a form of co-autonomy allowing both the surgeon and robot to simultaneously control motions

Traditional open surgery

In a traditional open surgery the surgeon makes an incision on the body of the patient and moves muscles to one side in order to have an open path to the spine. With muscles pulled away the doctor can access the spine to remove diseased or damaged bone or intervertebral disks. The surgeon can also directly see the spinal anatomy and place screws or any other materials necessary to stabilize the vertebra and help healing.

Minimally invasive surgery

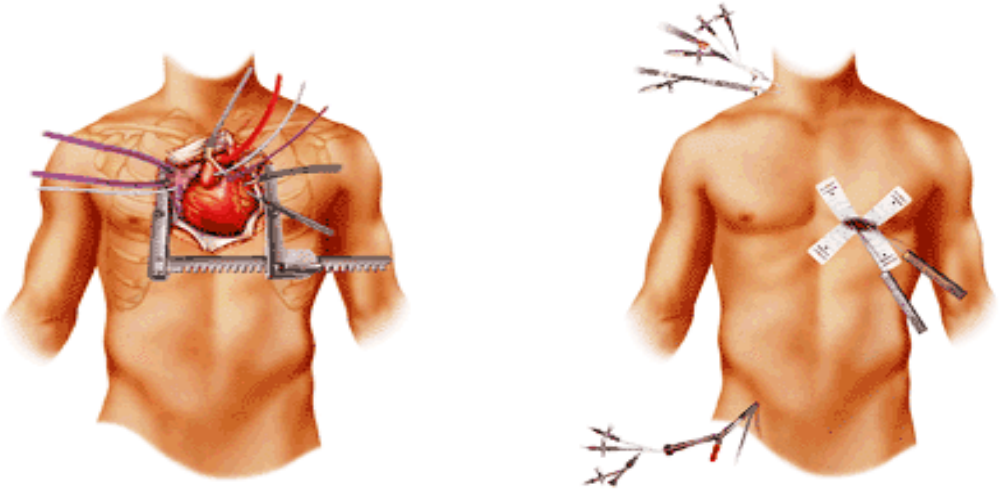
Minimally invasive spine surgery (MISS) does not involve long incisions and open manipulation of muscles or tissues surrounding the spine is avoided. The surgeon perform the procedure through small cuts.

There are a lot of advantages compared to a traditional open surgery, here are listed some of the most significant:

- shorter operation time
- lower blood loss
- less wound complications (i.e. infections)
- less damage to soft tissue
- shorter hospitalization
- short recovery time

Whether open surgery or MISS is performed, the spine can be accessed from different directions. These are referred to as surgical approaches:

- Anterior approach: the surgeon accesses the spine from the front of your body, through the abdomen.
- Posterior approach: the incision is made in your back.
- Lateral approach: the pathway to the spine is made through the side.



(a) Open surgery (b) Minimal invasive surgery

Figure 1.3: Approaches [3]

1.5 Computer Assisted Navigation (CAN) Options

Many platforms are currently available for use in the field of spine surgery. The Airo Mobile Intraoperative CT-based CAN platform (Brainlab) was approved for use in the US by the Food and Drug Administration (FDA) in October, 2013. The technology is one of the pioneers of the navigation platforms and has many similarities to other CAN systems. Some differences in the platform include its mobility and larger diameter of scanner than other manufacturer's scanner. The circular scanner is attached to the operating table and allows for full 360° scanning. Additionally, the entire unit is mobile. The patient is anaesthetized and intubated then transferred to the operating table. The instruments to be used in the surgery have 3 attached reference points that are recognized by the system's scanning stereotactic camera. These instruments may be calibrated during the anaesthetization process. Prior to intraoperative CT scanning, an anatomic reference clamp is attached to a spinous process that allows registration of the CT image with the same camera used for the instrument registration. The patient

is then put through the scanner and a full 32-slice CT scan is obtained. Once the image is obtained, it is automatically registered to the software, which generates a real-time 3-D map that is registered with the precalibrated instruments allowing for stereotactic guidance of instrumentation.

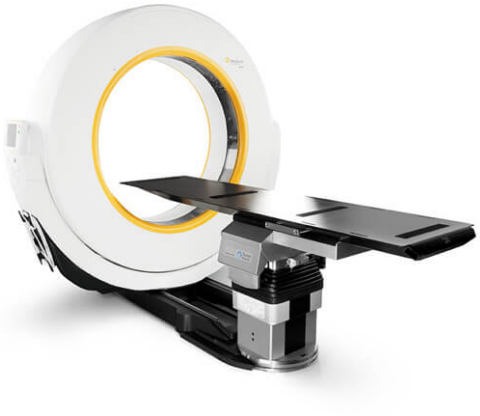


Figure 1.4: Airo Mobile [6]

The Stryker SpineMask Tracker provides a different form of reference. This system relies on a rectangle of trackers that is applied directly on to the patient. This referencing system negates the possibility of reference point translation and allows the surgeon to work free of obstacles. The novel design is not without its own peculiarities though. The camera must have full view, clear of obstruction to 5 of the 31 LEDs that are actively tracking. This means that a surgeon's arm or body resting on the patient must not obstruct more than half of the reference points of the rectangle tracker. Additionally, the operative field cannot extend beyond the predefined size parameters of the rectangular reference points. Additionally, undue skin tension or movement, such as what might be encountered with large vertical incisions and deep retraction, may alter the reference point location, effectively altering the 3-D computer-generated spinal map. For these reasons, this SpineMask Tracker seems well suited for minimally invasive percutaneous procedures, but may be less accommodating for larger, open procedures involving many segments

of the spine.

Safety and Applications of Intraoperative Navigation

The application of CT-based 3-D navigation in spine surgery has been well studied with over 20 clinical trials utilizing various manufacturers' platforms. Primary end points in the majority of studies have evaluated the accuracy and safety of pedicle screw placement utilizing this technology. In addition, several meta-analyses and systematic reviews have attempted to resolve the clinical equipoise surrounding CAN techniques. However, even with a critical mass of data on well over 10.000 pedicle screws, many still interpret the literature as equivocal. This likely has as much to do with the well-documented success rates and safety profile of free-hand (FH) pedicle screw instrumentation as it does with the evolving technological limitations of the many different platforms for CAN that may add significant heterogeneity to results.

Intraoperative MRI or MRI/CT Coregistration CAN

Leksell et al [29] first described the technique of stereotactic surgery utilizing MRI in 1985 for use during localization of deep brain tumors. This novel technique allowed surgeons to visualize, real-time, the brain with clear distinction of white and gray matter, ventricular anatomy, and deep brain pathology. Their technique utilized a 0.5-Tesla magnet to produce real-time imaging for stereotactic surgery which utilizes an aluminium head-ring. A decade later, Cohen et al [30] reported on their clinical results using coregistration of CT and MRI modalities. They compared the accuracy of their novel coregistration techniques to those of independent MR and CT alone and found that coregistration of CT/MRI resulted in a statistically significant decrease in error, measured in mm, in all 3 planes of view. This technology has since been used for functional neurosurgery, primarily in form of thalamotomy, placement of deep brain stimulators, and brain and spinal tumor resection as previously described. In following years, the application of 3-D coregistration technology was broadened to visualize both neural elements and bony structures of the cervical, thoracic, and lumbar spine

with great success and fine detail.

Efficiency in the OR and Outcomes

While many studies have demonstrated improved accuracy of pedicle screw instrumentation utilizing CAN techniques, there is a shortage of literature reporting on patient outcomes. Perhaps with greater accuracy and precision of instrumentation comes fewer complications and improved outcomes, though this inference is largely an assumption due to lack of existing data. The success of a spinal fusion procedure depends on many more variables aside from pedicle screw instrumentation.

The reported rate of pedicle screw misplacement can be as high as 20% to 40% based on historical data. However, even in studies with such high rates of misplaced screws, only a fraction of patients experience neurological, visceral, or vascular-related complications. Wiesner et al [31] reported on 408 percutaneously placed lumbosacral pedicle screws and found 27 instances of screw malposition. However, of those 27 misplaced screws, only 1 was found to cause a neurological complication. This salient point that differences in pedicle screw accuracy, such as that seen in FH, fluoroscopic guided, and CAN, are not associated with clinical significance has been reproduced in the literature with convincing power. This may not be surprising as the neural elements of the lumbosacral spine tend to be more forgiving due to their increased mobility as compared to cord-level neural structures, in which there is less room for error. Therefore, the increased accuracy of CAN pedicle screw placement may translate into improved safety to a larger degree in the cervical and thoracic spine.

1.6 Robot-Assisted Spinal Surgery

Robot assisted surgery (RAS) is an interdisciplinary growing field that aims to support surgeons during surgical interventions.

The first surgical robot prototypes had been developed in the early 1990s, today the most prominent commercial system are the MAZOR Renais-

sance/Spine Assist, the Medtech ROSA robot and the new Da Vinci surgical system.
The main advantages

- reduce intraoperative time
- reduce invasiveness of the procedure, enhance ability to perform minimally invasive surgery
- higher accuracy
- reduce surgeon's fatigue and hand tremor
- reduce staff exposure to radiation
- reduce operating room staff
- enable new procedure
- improve diagnostic abilities

In last years the number of performed robot-assisted procedures: from few hundreds in the early 2000s to more than 600.000 in 2017, according to Surgical Intuitive Inc. Anyway for several reasons surgical robotic system are slowly assimilated in the operating rooms and not homogeneously spread. Surgical robotic systems are expensive, their use is limited to few large research hospitals that can afford the high costs. Operating time may be reduced but usually surgeons must perform hundreds of procedures to become adept and efficient in their use. The systems have big volumes and they might occupy too much operative space raising safety issues.

The advances in technology that have led to the refinement of CAN techniques previously discussed offer a safe and efficacious alternative to traditional radiation-intense FH for pedicle screw instrumentation. These platforms are not without flaw; however, accuracy depends on several variables including a direct line of sight from the tracking system camera to the instrumentation tools, relative angles between the camera and registered instruments, camera quality, surgeon skill and expertise in acquiring and registering

images, and environmental conditions such as heat, humidity, and light. In an attempt to mitigate some of these shortcomings, miniature robotic systems that attach directly to bony landmarks were conceptualized in the early 2000s. These robotic assistants utilize the same CAN platforms, but lack the drawbacks of surgeon interference with the tracking system cameras and introduce the indefatigably and reproducibility inherent to robotic systems.

SpineAssist robot

One of the pioneers and by far the most studied of these robotic-assisted surgical devices for spine surgery is the SpineAssist/Renaissance robot (MAZOR Robotics Inc, Orlando, Florida). This device operates under a shared-control model, with 6° of freedom of motion positioning surgical instruments for spinal procedures. It utilizes 3 different outrigger arms, each accommodating a drill guide sleeve. The robotic software, in sync with a CAN, determines which arm produces the most accurate pathway for pedicle instrumentation based on the chosen implant and relative location of the SpineAssist robot to the predetermined entry point and screw trajectory. The robot may be attached directly to a spinous process in the case of open surgery, or attached to a frame triangulated by percutaneously placed guide wires for MIS procedures.

The first step of the process is to obtain and register CT images of the desired spinal levels with the SpineAssist software to create a virtual spinal map for the robot. The second step involves the templating of desired screw entry point, trajectory, and screw size. This may be done in the OR or even preoperatively based on the 3-D spinal map constructed by the software and transferred to the intraoperative SpineAssist workstation.

The final registration involves obtaining 6 still fluoroscopic images for calibration and intraoperative registration purposes. The SpineAssist software then determines the optimal position of the selected arm for insertion of the drill sleeve and a cannulated drill guide is placed in the arm, which is now aligned along the predetermined implant trajectory. The drill is then used to create a cortical punch at the desired entry point; a guide wire is

inserted into the vertebral body so a screw pilot hole may be drilled along the guide wire. The appropriate length and diameter screw is then inserted into the pilot hole after pedicle probing and surgeon confirmation of accuracy.

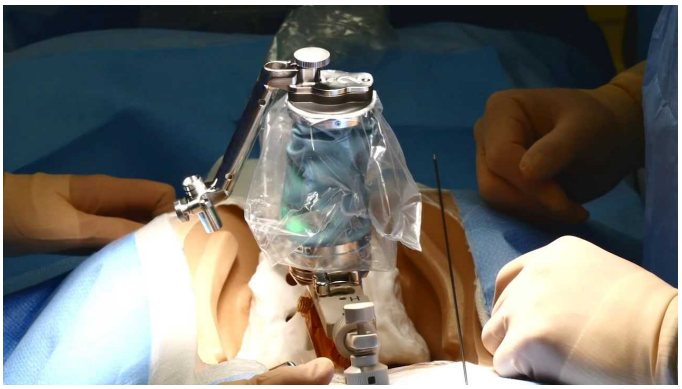


Figure 1.5: Spine Assist/Renaissance robot[7]

Studies first verified the accuracy of this novel robotic-assisted technique, reporting an average deviation of 1 mm or less of actual implant position compared to preoperative template. Soon thereafter, several clinical studies sought to expand upon the translational accuracy and efficacy of the SpineAssist robot in Vivo. Roser et al [32] found a 99% accuracy rate of lumbosacral pedicle instrumentation using the SpineAssist robot compared to 98% utilizing fluoroscopy guided, and 92% using navigation techniques. Interestingly, the only study to date demonstrating a reduced accuracy of screw placement came from Ringel et al in a randomized controlled trial that demonstrated a significantly reduced accuracy rate of lumbosacral pedicle screw instrumentation with the SpineAssist robot (85%) compared to fluoroscopic-guided screws (93%, P = .019). The authors also reported that 10 of the 146 screws placed with robotic assistance necessitated intraoperative removal and reimplantation. The authors utilized a percutaneous means of affixation of the robot to the spine, and noted instability in the wire leading to malposition of the drill sleeves.

ROSA robot

In a new application of an existing surgical robot, the ROSA robot by Medtech (Medtech S.A., Montpellier, France), originally designed for cranial neurosurgical applications, may provide the answer to the technical flaws encountered by Ringel et al [33].The ROSA robot is a freestanding robotic assistant with a floor-fixable base and a rigid robotic arm (Figure 6). This may help mitigate concerns of fixation strength to bony anatomy like those encountered by Ringel et al. Additionally, the robotic arm moves in concordance with the patient, based on the tracking camera monitoring, real-time, several percutaneously placed tracking pins to the patient's bony anatomy in reference to tracking spheres affixed to the robot.

This technology platform, however, has yet to be validated for use in spinal pedicle instrumentation but early clinical results are promising. In their preliminary study on the novel application of the ROSA robot for spinal surgery, Lonjon et al reported an accuracy rate of 97.3% for pedicle screw instrumentation compared to 92% in the FH group. Though seemingly better suited for percutaneous and MIS procedures due to improved robotic arm fixation, these are the first published data of the ROSA robot for spinal applications and more data are needed to validate its use.

Figure 1.6: ROSA robot [8]

Da Vinci Surgical System

A discussion of surgical robotics would not be complete without mention of

the Da Vinci Surgical System (Intuitive Surgical). The Da Vinci robot was approved in 2000 for general laparoscopic procedures and is most commonly used for prostatectomies and hysterectomies, but spinal applications of the technologically advanced system have been proposed. The Da Vinci robot operates under the telesurgical model by which the surgeon operates the robot as an extension of his or her own arm from a remote telesurgical booth. The system is equipped with 3-D vision screens and portals for the surgeon's hands to control robotic instruments. Among the benefits of this robotic assistant that have led to its widespread use in the fields of general surgery, urology, and gynaecology are high definition, stereoscopic vision with magnification up to 10×, tremor filtering, limitless wrist range of motion, and improved surgeon ergonomics. Additionally, the telesurgical model allows for close oversight from a separate both affording override, making it an ideal form of trainee education.

The Da Vinci Surgical System (Intuitive Surgical) has been utilized for laparoscopic anterior lumbar interbody fusion (ALIF) with promising results. The primary obstacles to ALIF remain the ureters and large vessels (aorta, vena cava, and branches) overlying the anterior spine. The first laparoscopic ALIF was reported in 1991, with hopes of shorter hospital stay, quicker recovery, less postoperative pain, and smaller incisions through the MIS approach. However, results failed to show any advantage over open ALIF in regards to length of stay, blood loss, or complication rates, and additionally, the technical demands, often foreign to spine surgeons, resulted in a steep learning curve with increased operative time. For these reasons, the procedure was largely abandoned by spine surgeons. However, with the improved usability of the Da Vinci robot, the procedure and its hypothesized improved efficacy, the Da Vinci-assisted laparoscopic ALIF has again become relevant in the spine realms. Several small case-series studies have evaluated this application of the Da Vinci robot demonstrating successful dissection of overlying large vessels and no ureter- or vessel-related complications. However promising, the use of the Da Vinci is not FDA approved for actual spinal instrumentation and more exploration is necessary to validate its use.

Figure 1.7: Da Vinci surgical system [9]

Chapter 2

Project and Starting Point

2.1 Project introduction

The goal of this thesis work is to implement and test a new software solution for handling the joint limits problem on an hand-guided robot, while respecting some precise requirements.
The entire work is the result of a collaboration between my university Politecnico di Milano and the medical department of the KUKA company in Augsburg and was developed during a six months internship.
It can be considered as a small contribution to a bigger project already started in the past years from the company. The aim is to introduce an LBR MED in the operating room for spine surgery interventions, the robot along with the existing imaging systems can support the surgeons in the drilling tasks required for safe and accurate spine fusion procedures.

In a standard procedure the surgeons have to perform the drilling manually using the available medical images as their only support.

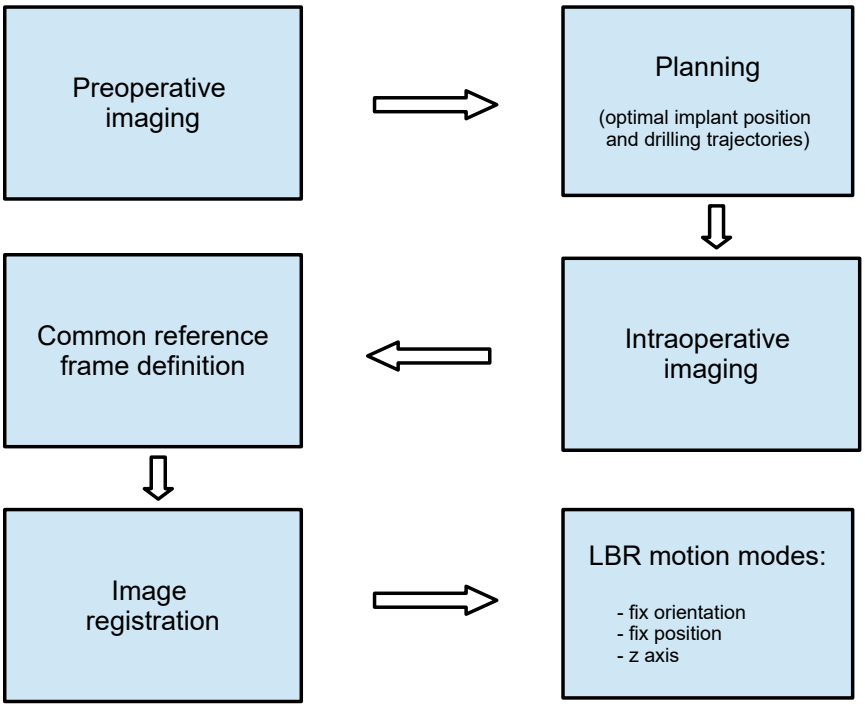


Figure 2.1: Assisted workflow scheme

Instead the ideal workflow using the LBR med will follow the steps reported in Figure 2.1.

From images it's already possible to define the optimal position and orientation for the drilling task, then by defining a common reference frame it would be possible to guide the robot flange to the desired pose while getting numeric feedback from the robot.

Figure 2.2: Admittance Control Scheme

At anytime if there are no hand guiding inputs commanded the robot will hold its current pose, thanks to the gravity compensation, so for the surgeons it will be a lot easier to check images and make corrections little by little while the robot remains stable.
The final step is characterized by specific cartesian constraints and three relative motion mode (fix orientation, fix position, z-axis).

- Position the tcp on the desired drilling starting point (fix orientation motion)
- Align the tcp on the desired drilling trajectory (fix position motion)
- Perform the drill with a constrained motion on the z-axis

Some of the most notable advantages from robot assisted task are:

- Increased positional accuracy
- Increased drilling task linear precision
- Reduced surgeons fatigue
- Provide numeric feedback
- Possibility to hold the current pose
- Improved ability to perform minimally invasive surgery

Robot Requirements

During the planning of the project a list of medical requirements have been provided. The final LBR Med prototype must be able to fulfil all of them.

Figure 2.3: LBR Med [5]

Medical Application Specific Requirements

1. Absolute positional accuracy of the robot (0.5 mm, 0.25 deg)
2. Cartesian kinematics constraints

Customer Additional Requirements

3. The workspace should be restricted as little as possible
4. Possibility to "slide along" joint limit (the robot must never freeze all the joints at the same time)
5. Avoid (computationally) unnecessary singularities
6. No noticeable breakaway force/torque
7. No drift
8. Adjustable joint velocity limits (without safety stop)
9. Smooth behaviour in the joint limits
10. No unexpected motion of the robot

2.2 Project focus and goal

As previously mentioned the focus of my work was the joint limit problem, a lot of the listed requirements are related to it, some directly other indirectly.

First goal
At first the robot must be able to "slide along the joint limit", the robot must never freeze all the joints at the same time even some limits are reached. As I will underline in the literature research chapter this is a common drawbacks for many of the existing solutions that make use of a common scaling factor for all the joints, in these cases when one limit is reached the maximum feasible velocity is set to zero and then the scaling factor is reduced to zero as well, resulting in no motion for the robot.

Second goal
Another requirement is the smooth behaviour in the joint limits, the robot need to gradually decrease its velocity while approaching one limit and never produce a hard stop into it. This point is strictly related to the definition of some joint velocity limits that will bound the maximum velocity and scale the command send to the robot motors.

Challenge
The most challenging requirement is not directly related to joint limits but is the presence of cartesian kinematics constraints. These constraints must always be respected in order to ensure a safe surgical intervention and at any time must be seen as our primary task to not be violated. When the robot tries to overcome a position or, if defined, a velocity limit the desired commanded task is not feasible any more and through the Inverse Kinematic function is necessary to introduce some kind of truncation or scaling. Under cartesian kinematics constraints if we just try to scale the critical joints individually the produced motion will violate them.

2.2.1 Original Inverse Kinematic

The original IK function solves the problem with a Task Priority formulation where the cartesian constraints have the highest priority and the hand guided motion the lowest. When the commanded position tries to violate one of the position limits the corresponding joint is frozen. To realize that the row of the Jacobian relative to the critical joint is entirely set to zero, while all other joints are still able to freely move, respecting the requirement, but the biggest drawback is the hard stop of joint motion at their limits.

The hard stop can produces an aggressive and behaviour for the user, especially if he is a non robotic expert and he is not aware of critical configuration, and at the same time may force the robot to produce unnecessary high peak torques to suddenly stop the motion, the hardware receives really high stress under intensive use.

2.3 Starting module features

In this section i want to give an overview of the already existing hand-guiding module of the KUKA medical team. My entire job was developed on it and some of its features are briefly described in order to provide a better general description of the entire hand-guiding module aside from the specific thesis focus on the joint limits problem solution.

These are the most noticeable features that will be described without many details:

- Admittance control (accuracy)
- Singularities detection and Elbow motion restriction (singularities avoidance)
- Drift compensation with dead zone (fine positioning)

2.3.1 Admittance Control

The module input are the measured external forces at the TCP and the output are joint positions.
More precisely external forces are integrated to obtain the desired cartesian velocity, than cartesian velocity is transformed into joint velocities through the differential Inverse Kinematic function. At last joint velocities are integrated and the resulting joint positions are commanded to the robot.
The admittance control can achieve the required absolute accuracy of 0.5 mm for the medical application. While the position accuracy has been proved to be a lot better than in the most used impedance control, the admittance control may suffer from stability problem in contact with external rigid environment, so it may result not well suited for object interaction and manipulation.

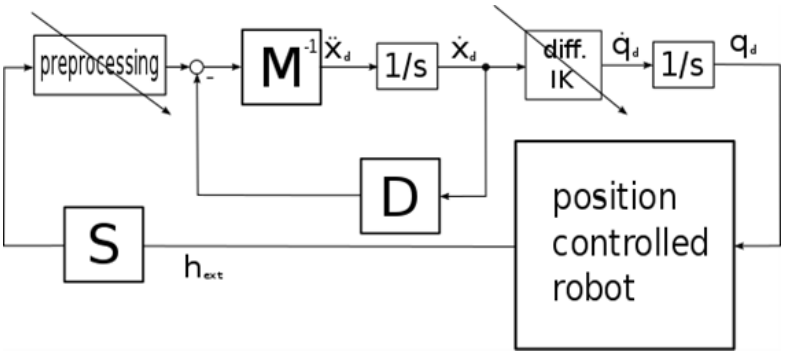


Figure 2.4: Admittance Control Scheme

In the figure the preprocessing block include some non-linear transformation on the input forces. For example a symmetric dead zone around the zero point is implemented, very low input under a certain threshold are suppressed, this transformation has been proved to ensure a better absolute accuracy.

2.3.2 Singularities

Singular points are particular configuration of the robot where the minimum singular value of the Jacobian is equal to zero. The inversion of the Jacobian around singular point can produce very high values that will lead to high

peaks in joint velocities. Singularities handling is another well known problem for robot control The singularities detection of the module is integrated in the function used to compute the inverse of the Jacobian and realize a particular weighted pseudo-inverse.

Chapter 3

Literature Research

3.1 Joint Limits Problem

3.1.1 Gradient Projection Method

This method is one of the first and most used for solving robot redundancy. Joint velocities $\dot{q}$ can be compute as the product between cartesian velocity $\dot{x}$ and the sudden inverse of the Jacobian $J^{\dagger}$; then the nullspace of the primary task is used to minimize an objective function h, $z = k\nabla h$.

$$\dot{q} = J^{\dagger}\dot{x} + (I - J^{\dagger}J)z \tag{3.1}$$

The vector z can be freely choose to optimize any desired criteria without affecting the primary task.

The most important optimization criteria used in past works are:

- Least-square joint velocities $h(q) = \frac{1}{2}\dot{q}^T W \dot{q}$ [1];
- Distance from joint limits $h(q) = \frac{1}{2}(q - \bar{q})^T W (q - \bar{q})$, [2];
- Singularity avoidance $h(q) = \sqrt{det(J(q)J^T(q))}$ [3];
- Obstacle avoidance $h(q) = \min_{p,o} \|p(q) - o\|$ [4].

The method provides only the direction of the homogeneous solution, the scalar coefficient k has to be manually set in order to adjust the magnitude of the self motion.
High k can result in excessive and unnecessary self-motion always trying to keep joints in the middle of their ranges even if they are far enough from the bounds. A large k has also been proven to produce oscillations near a minimum.
With low k the nullspace contribution will be able to change the robot configuration only when the projection vector become very large, in proximity of the limits, that may be too late to avoid them in time for hardware limitation.

Huo [5] in a more recent work presented a method to automatically tune parameters.

The GPM in general cannot guarantees joint limits avoidance both for the presence of local minimum and of the primary task.
For these cases the commanded joint motion may still violate some of the bounds, and its saturation produces then a wrong instantaneous behavior in the task space, as an hard stop of some joints.

3.1.2 Task Priority

The Task Priority formulation is a generalization of the previous method to multiple task problems. It aims to solve the conflicting task situation by suitably assigning an order of priority to them. The lower priority tasks are satisfied only in the Nullspace of the higher priority ones. [6] [7].

This is the general formula for joint velocities fro an arbitrary number of tasks t:

$$\begin{aligned} \dot{q}_i &= \dot{q}_{i-1} + (J_i P^A_{i-1})^\dagger (\dot{x}_i - J_i \dot{q}_{i-1}) \\ \dot{q} &= \dot{q}_t \end{aligned} \tag{3.2}$$

Where

$$P_i^A = I - J_i^{A\dagger} J_i^A$$

is the projector into the nullspace of the augmented Jacobian:

$$J_i^A = \begin{bmatrix} J_1 \\ J_2 \\ . \\ . \\ J_i \end{bmatrix}$$

The order of priority can be assigned to best fit in the developed application but is not possible to guarantee the fulfillment of all task at the same time.

3.1.3 Task Space Augmentation

This method is based on the augmentation of the Jacobian matrix to include joint angle and joint rate constraints. The difference from the previous strategy is that here sub-tasks are executed along with the primary one without any priority order.

Chiacchio proposed a CLIK (closed loop inverse kinematic) control scheme with the Jacobian transpose [8], while Baron's solution [9] is based on a virtual rotation of the tool-tip in order to obtain an undetermined linear algebraic system with at least one DoF of redundancy.

The augmented kinematic equation can be written as:

$$x = \begin{bmatrix} x_E \\ x_C \end{bmatrix} = \begin{bmatrix} f_E(q) \\ f_C(q) \end{bmatrix}$$

And by differentiating:

$$\dot{x} = \begin{bmatrix} J_E(q) \\ J_C(q) \end{bmatrix} \dot{q} = J(q)\dot{q} \tag{3.3}$$

where J(q) is the augmented Jacobian matrix.

The augmentation of the Jacobian can introduce singularities even in feasible configuration for the single tasks, Baillieul named them *algorithmic singularities* [10]. A crucial point then remains the specification of suitable constraint tasks.

Where the augmented Jacobian is not full rank this solutions introduces relevant errors in both the considered tasks, Chiacchio in his work solved this problem by switching to a task priority strategy where a suitable specification of constraint task is not allowed.

3.1.4 Jacobian Weighting

The Weighted Least Norm (WLN) solution was first proposed by Chan [11] and then generalized for multiple sub-tasks by Xiang [12].
This solution penalize the motion of some joints over others by defining a weighted norm of the velocities vector:

$$|\dot{q}_w| = \sqrt{\dot{q}^T W \dot{q}} \tag{3.4}$$

The general formula can be write as follow:

$$\dot{q}_w = W^{-1} J^T (J W^{-1} J^T)^{-1} \dot{x} \tag{3.5}$$

The weights matrix is usually a diagonal matrix where the ith element on the diagonal is defined as:

$$w_i = \begin{cases} 1 + |\frac{\partial H(q)}{\partial q_i}| & if\ \Delta|\frac{\partial H}{\partial q_i}| \geq 0 \\ 1 & if\ \Delta|\frac{\partial H}{\partial q_i}| < 0 \end{cases} \tag{3.6}$$

The WLN solution was compared to the GPM in the Chan work, here below are listed some of the resulting advantages:

– it guarantees joint limit avoidance

– it doesn't required to set the magnitude of the self motion

– it doesn't try to maximize the distance of the joints from their limits reducing all the unnecessary self-motion and oscillation

– it doesn't damp motion of joints that are moving away from their limits even if $\frac{\partial H(q)}{\partial q_i}$ is high

The bigger drawback of this algorithm is that it cannot always guarantees the correct execution of the primary task, the damping will introduce an error in its execution.

3.1.5 Quadratic Programming

This category of methods defines a constrained quadratic optimization problem (Quadratic Programming problem) to solve the inverse kinematic problem for redundant manipulators under hard physical constraints. In [13] [14] is possible to find all the mathematical concepts, the standard formulations and some resolution techniques.

In the robotic field a first formulation for torque minimization was proposed in the 1980s by Hollerbach and Suh [15], they presented several schemes at acceleration level but all affected by stability problems. Joint velocity norm can diverge to infinite value in finite time and it's possible to obtain local discontinuities.
The divergence and the instability problems were analyzed respectively by O'Neil [16] and Park [17], some possible solution are presented but none of them can entirely solve the problems.

In 2004 Zhang proposed a unified approach for physically constrained redundant manipulators [18][19]. This formulation is general in the sense that it incorporates equality, inequality and bound constraints at the same time.

There exist several schemes both at velocity and acceleration level based on different optimization criteria, the ones listed are some of most used:

– Velocity level

  MVN minimum velocity norm

  MKE minimum kinetic energy

– Acceleration level

  MAN minimum acceleration norm

  MTN minimum torque norm

Usually real time applications require sampling time in the order of milliseconds. Some optimization problem can be really expensive in term of computations and so not suited for those applications. Cheng [20] [21] proposed the Compact QP method, a computationally efficient solution, which makes use of the Gaussian elimination to eliminate some of the equality constraints and reduce the number of independent variables.

However these methods do not guarantee in general both the constraint satisfaction and the optimality of the solution, they can result expensive for real-time application or present some instability problems.

3.1.6 Compensation in the Nullspace

The basic idea here is to compensate the saturation of joints velocities with the self-motion, so that the velocity in the task space remains unchanged. In this way we could achieve maximal velocity in the task space within the allowed joint velocity limits.

The method was present at first by Chen for one primary task [22], while later Omrčen presented a generalized version for multiple sub-tasks and introduced a condition that shows if the compensation is kinematically and mathematically possible or not [23].

The shared drawback is again the simultaneous treatment of all over driven joints at the same time, by proportional decreasing all joint velocities, in case the condition is not satisfy.

3.1.7 Saturation in the Nullspace

This approach was presented by Flacco in his work [24].

The velocities control functions used are able to consider at the same time all the hard constraints of the robot in terms of position, velocity and acceleration bounds:

$$\dot{Q}_{max} = \ min \left\{ \frac{L_+ - q}{dt},\ V_{max},\ \sqrt{2A_{max}(L_+ - q)} \right\} \tag{3.7}$$

$$\dot{Q}_{min} = \ max \left\{ \frac{L_- - q}{dt},\ -V_{max},\ -\sqrt{2A_{max}(L_- - q)} \right\} \tag{3.8}$$

At first is checked if the original task can be performed with the joint velocity being within their constraints.

The joint velocities are computed with the following formula:

$$\dot{q}_{sns} = \dot{q}_N + (JW)^{\dagger}(s\dot{x} - J\dot{q}_N) \tag{3.9}$$

where the parameters are initialized in this way: $W = I$, $\dot{q}_N = 0$ and $s = 1$.

If not the algorithm starts by evaluating the most critical joint for task execution (its velocity needs the smallest task scaling to stay within the bounds).
The most critical joint is then saturate in the nullspace by setting $W_{jj} = 0$ and $\dot{q}_N = saturate\ value$.

If the task cannot be executed with this new parameters, so the rank of JW is strictly less than m (dimension of the task), the algorithm stops with the best parameters found, otherwise the joint velocity is recomputed with the current parameters and the process is repeated on the new most critical joint.

At each loop a new scaling factor is obtained. If it's larger than all the previous ones it's stored with the current W and $\dot{q}_N$.

The damping in all directions is still present in this method if no feasible solution is found, but the task scaling factor used is assured to be the higher possible.

In more recent works Flacco extended the SNS algorithm also for multiple prioritized tasks and proposed also an optimal solution to reduce computation time. [25].

3.1.8 Progressive Clamping

This section refers to all those methods that treat joints singularly by clamping them according to some decreasing function.

The method presented by Raunhardt [26] for the control of virtual mannequins is conceptually similar to the Jacobian Weighting.

He defined a velocities control function with a polynomial profile as follow:

$$h = \begin{cases} 1 & if\ q_{k+1} < \bar{q}_{min}\ or\ q_{k+1} > \bar{q}_{min} \\ -2d^3 + 3d^2 & if\ q_{k+1} < q_k < q_{min}\ or\ q_{max} < q_k < q_{k+1} \\ 0 & otherwise \end{cases} \tag{3.10}$$

Then the resulting damping factor h is used to module the motion command: $q_{k+1} = q_k + (1 - h)\Delta q$

The method damps only joints heading towards the limits and offers the same advantages of the JW, but as with the JW algorithm also this solution can introduce errors in the execution of the task because joints are treated singularly when any damping is required.

Chapter 4

Implementation

4.1 Goal

As seen in Chapter 2 the existing KUKA hand-guiding module was already able to fulfil a lot of the requirements for spine surgery.

Focus:

- Joint limits handling through IK function

Goals:

- Smooth approach to joint limits (kinematic constraints)
- Possibility to "slide along" joint limit (the robot must never freeze all the joints at the same time)

Challenge:

- Cartesian Constraints

4.2 Cartesian Constraints

The medical application requires cartesian kinematic constraints in order to ensure an accurate and safe surgical procedure, for this reason they have to be considered as the highest priority task. The precision and the accuracy are crucial during the execution of a surgical procedure, no rotational or positional error have to be present even when joint position or velocities are saturated.

The hand-guiding task, represented by the motion commanded by the user with his hand, is mapped into the nullspace of the considered cartesian constraint by following the general task priority formulation. In this way it's guaranteed that kinematic constraints are always respected.

4.2.1 Task Priority Formula

This is the general formula taken from literature (3.1.2) for an arbitrary number of tasks t:

$$\begin{aligned} \dot{q}_i &= \dot{q}_{i-1} + (J_i P_{i-1}^A)^\dagger (\dot{x}_i - J_i \dot{q}_{i-1}) \\ \dot{q} &= \dot{q}_t \end{aligned} \tag{4.1}$$

Where

$$P_i^A = I - J_i^{A\dagger} J_i^A$$

is the projector into the nullspace of the augmented Jacobian:

$$J_i^A = \begin{bmatrix} J_1 \\ J_2 \\ . \\ . \\ J_i \end{bmatrix}$$

4.2.2 Our Case

– Primary task: Cartesian Constraint

– Secondary task: Hand-guiding Motion

Cartesian Constraint

$$\begin{aligned} J_1 &= J_c \\ \dot{q}_1 &= \dot{q}_c = 0 \end{aligned} \tag{4.2}$$

Hand-guiding Motion

$$\begin{aligned} P_1 &= N_c \\ J_2 &= J_h g \\ \dot{q}_2 &= \dot{q}_{hg} = \dot{q}_1 + (J_2 P_1)^\dagger (\dot{x}_2 - J_2 \dot{q}_1) = (J_h g N_c)^\dagger (\dot{x}_h g) \end{aligned} \tag{4.3}$$

For spine surgery is really important to always keep to zero the error in the primary task. For this reason solutions that modify the Jacobian (i.e. "Jacobian Weighting"), or that clamp and saturate joints singularly (i.e. "Progressive Clamping") are not suitable for this application.

4.3 Smooth Approach

The smooth approach to joint limits was realized with the introduction of velocity limits. Their formulation is taken from the literature work "Saturation in the Nullspace".
With this formulation maximum and minimum velocities are computed taking into account not just velocity bounds but also position and acceleration bounds all together.

$$Q_{min} \leq q \leq Q_{max} \tag{4.4}$$

$$V_{min} \leq \dot{q} \leq V_{max} \tag{4.5}$$

$$A_{min} \leq \ddot{q} \leq A_{max} \tag{4.6}$$

At each step the joint velocity command must guarantee that:

- the joint range limits will not be violated
- the commanded joint velocity doesn't exceed the velocity absolute limit
- the joint will be able to stop the motion in time before reaching the position limit, considering the acceleration bounds

Position

Starting from a generic feasible position q_k, for a sampling time T, the new commanded position can be written as:

$$q_{k+1} = q_k + \dot{q}_k T$$

This new commanded position q_{k+1} has to fall in the admissible range:

$$Q_{min} \leq q_{k+1} \leq Q_{max}$$

From here is possible to formulate maximum and minimum velocities that always guarantee that position limits will not be violated:

$$\frac{Q_{min} - q_k}{T} \leq \dot{q}_k \leq \frac{Q_{max} - q_k}{T} \tag{4.7}$$

Acceleration

At each sample we also need to be able to stop the joints before they can hit their position limits. The fastest possible way to stop the motion is to

decelerate the joint, which is moving at $\dot{q}_k > 0$ speed, with the maximum acceleration A_{max}.

Position and velocity at some $t > t_k$ and subject to $-A_{max}$ can be written as follow:

$$q(t) = q_k + \dot{q}_k(t - t_k) - \frac{A_{max}}{2}(t - t_k)^2 \tag{4.8}$$

$$\dot{q}(t) = \dot{q}_k - A_{max}(t - t_k) \tag{4.9}$$

In the extreme situation the joint velocity is reduce to zero ($\dot{q}(t) = 0$) exactly in the position limit ($q(t) = Q_{max}$). If we solve the system with this parameters is possible to demonstrate that in this case the velocity is bounded as follow:

$$-\sqrt{2A_{max}(q - Q_{min})} \leq \dot{q}_k \leq \sqrt{2A_{max}(Q_{max} - q)} \tag{4.10}$$

Complete Formula

$$\dot{Q}_{max} = \min \left\{ \frac{Q_{max} - q}{dt},\ V_{max},\ \sqrt{2A_{max}(Q_{max} - q)} \right\} \tag{4.11}$$

$$\dot{Q}_{min} = \max \left\{ \frac{Q_{min} - q}{dt},\ -V_{max},\ -\sqrt{2A_{max}(q - Q_{min})} \right\} \tag{4.12}$$

The Figure 4.1 shows the profile of the resulting area of admissible velocities in function of the position. While approaching to the position limit the admissible velocity is smoothly reduced to zero.

If the user tries to command a velocity over the limit the robot motion will be damped and the user will feel an increasing resistance to the motion in direction of the limit.
The Appendix A.1 shows the integral function used in the algorithm.

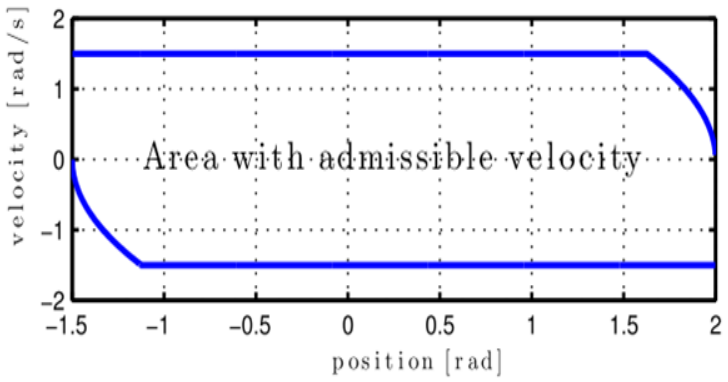


Figure 4.1: Feasible velocities area

4.4 Motion Along the Limits

The desired solution has to present the possibility to slide along the joint limits and never stop the robot if there are some degree of freedom left to achieve some motion.

In literature is possible to find a lot of works that always try to preserve the cartesian task direction. Here the redundancy is used in different way to deal with the joint limits problem but the common drawback is the simultaneous treatment of all joint at the same time if no feasible solution can be found. In this case the entire task is scaled by a common factor.

This factor $\alpha \in [0\ 1]$ represents the smallest scaling factor necessary to stay within the bounds and it is computed as follow:

$$\alpha = \min \left\{ \frac{\dot{Q}_{max_i}}{\dot{q}_i} \ , \ \frac{\dot{Q}_{min_i}}{\dot{q}_i} \right\} \tag{4.13}$$

Then the entire task is scaled according to it $\dot{x}_{new} = \alpha\ \dot{x}_d$. If one position limit is hit, the maximum velocity for this joint is equal to zero and then the scaling factor is also reduced to zero, the motion is stopped and all joints are frozen.

To move away from a joint limit is necessary to command a cartesian motion which doesn't present any cartesian component that tries to violate this position limit. If even a small component is present then the robot will remained frozen.

While for cartesian constraints was necessary to have no error in the primary task, to achieve the desired behaviour along joint limits will not be possible to always preserve the task direction but was required the introduction of an error in the hand-guiding task.

For a generic hand-guiding task is not strictly necessary to preserve the desired direction because the user has a continuous visual feedback on the position of the robot and he can decide to change trajectory at any time. Anyway it's important to never produce any unexpected or not intuitive motion, the user should never feel the robot moving against his intended motion.

My final solution can guide the user to move along joint limits but always in a natural way. The task direction is gradually changed while approaching to one limit and in the worst case, when a position limit is hit, the task direction is changed by 90 degrees.

4.4.1 First step: Geometric Clamping

From this section the fix orientation motion is considered, where the cartesian input velocity is a 3D vector with the three translational elements for x, y and z axis. All examples and pictures of this chapter will refer to this case because it is easier to visualize and to reason with more familiar 3D vectors and planes. [1]

[1]The same algorithm works also for other motion modes. For the fix position translational elements and axis are substituted by rotational ones, while the z-axis motion can be seen as a special case of the fix orientation where the hand-guiding task is mono dimensional.

The idea was to use the information contained in the Jacobian matrix to move the problem of joint limits from joint space to cartesian space.

The velocity of a generic joint can be written as:

$$\dot{q}_i = invJ_i \; \dot{x}_d \tag{4.14}$$

Where $invJ_i$ is the i_{th} row of the inverse Jacobian matrix, this vector can be seen as the "critical direction" for the joint at the current robot configuration.

The desired cartesian velocity is decomposed in two components. The first one parallel to the critical direction and the other one perpendicular to it.

$$\dot{x}_d = \dot{x}_{d\parallel} + \dot{x}_{d\perp} \tag{4.15}$$

Having the critical direction for the joint is possible to define a plane normal to it and passing from the zero point, which contains all the cartesian velocities that result in a zero velocity for the i_{th} joint, this happens because the scalar product between them and $invJ_i$ is always zero. The perpendicular component lays on this plane and doesn't have any influence on the resulting joint velocity.

$$invJ_i \; \dot{x}_{d\perp} = 0 \tag{4.16}$$

$$invJ_i \; \dot{x}_{d\parallel} = \dot{q}_i \tag{4.17}$$

A first feasible solution was computed by removing the parallel component.

$$\dot{x}_f = \dot{x}_\perp \tag{4.18}$$

In this way the i_{th} joint is blocked while the entire robot is still able to move on this perpendicular plane.

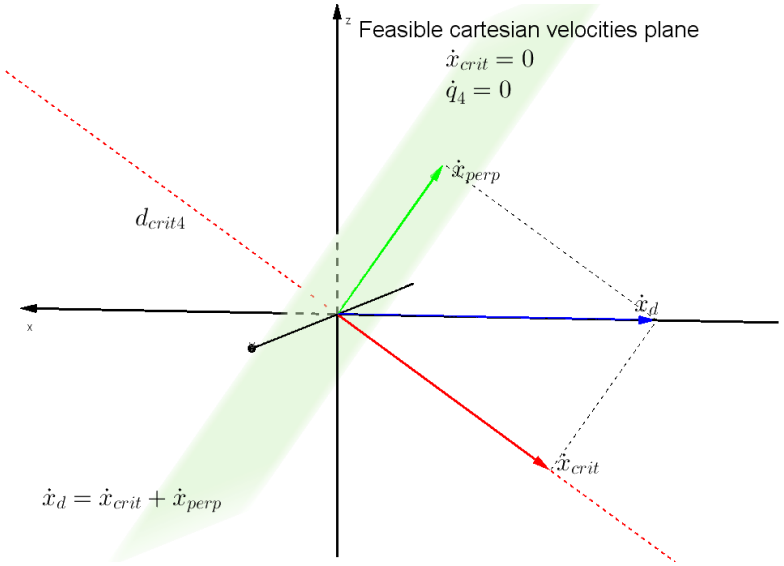


Figure 4.2: Geometric Clamping

4.4.2 Second step: Geometric Scaling

The second step done aims to guarantee continuity for joint and cartesian velocities.

Assuming to have one joint over its velocity limit

$$\dot{q}_{d_i} > \dot{Q}_{max_i}$$

to achieve a feasible solution the velocity of this joint need to be scaled by a factor $\alpha \in [0, 1]$

$$\alpha_i = \frac{\dot{q}_i}{\dot{Q}_{max_i}} \tag{4.19}$$

Instead of scaling the entire task we leave the robot free to move on the perpendicular direction, while we only scale the parallel direction.

From here the feasible solution found is:

$$\dot{x}_f = \alpha\ \dot{x}_{d\parallel} + \dot{x}_{d\perp} \tag{4.20}$$

It's important to remark that since $\dot{x}_{d\perp}$ as no effect on the resulting joint velocity ($invJ_i\ \dot{x}_{d\perp} = 0$), scaling the entire task or scaling just the parallel component will produce the same joint velocity, in this case equal to the maximum admissible.

$$\alpha\dot{x}_d = \alpha\dot{x}_{d\parallel} + \alpha\dot{x}_{d\perp} \tag{4.21}$$

$$\dot{x}_f = \alpha\dot{x}_{d\parallel} + \dot{x}_{d\perp} \tag{4.22}$$

$$invJ_i\ \alpha\dot{x}_d = invJ_i\ \dot{x}_f = \dot{Q}_{max_i} \tag{4.23}$$

Figure 4.3 shows how the parallel component is gradually damped while approaching to the joint limit. When the limit is reached only the perpendicular component is saved.

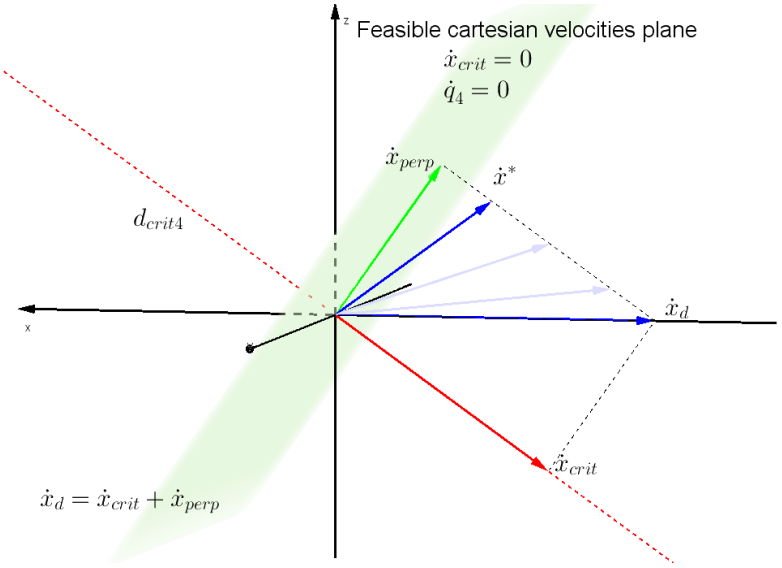


Figure 4.3: Geometric Scaling

4.4.3 Geometric Constraints

In this section is shown that each velocity limit at joint level can be converted in one cartesian constraint in 3D space.

From each velocity limit for each joint is possible to find the maximum feasible cartesian component on the critical direction: $\dot{x}_{\parallel_{max}} = \alpha\ \dot{x}_{d\parallel}$.
Then is possible to define a plane normal to the critical direction and passing through this point $\dot{x}_{\parallel_{max}}$, I will refer to it as "boundary plane". All cartesian velocities ending on that plane achieve a joint velocity equal to $\dot{Q}_{max_i}$.

$$invJ_i\ \dot{x}_{\parallel_{max}} = \dot{Q}_{max_i} \tag{4.24}$$

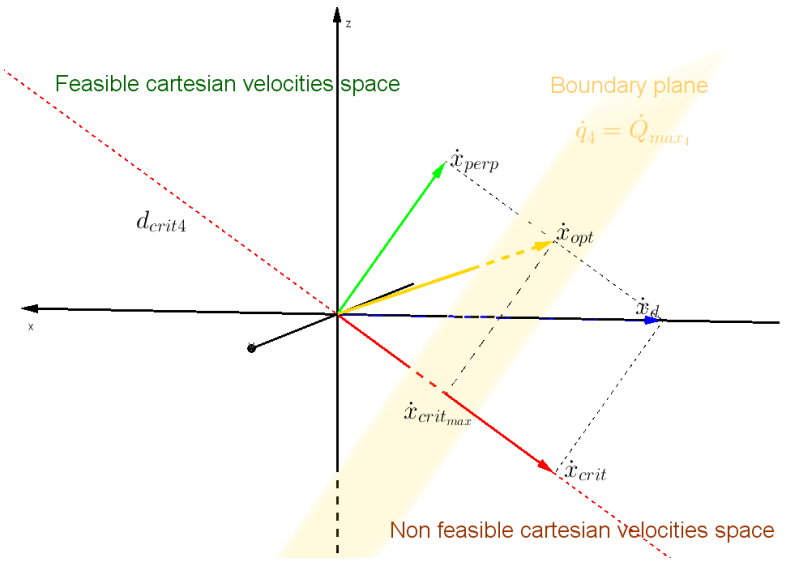


Figure 4.4: Geometric Constraints

Figure 4.4 shows an example of one "boundary plane". The plane divides into two hemisphere the entire space, all inputs on the left of the plane are feasible for the considered joint while all inputs on the right will produce a solution out of bounds.

Between the infinite feasible solutions we consider as optimal the one minimizing the error with the desired cartesian velocity $\dot{x}_d$. In presence of a

single constraints the optimal solution always coincides with the feasible one described in the previous section:

$$\dot{x}_{opt} = \dot{x}_f = \alpha\ \dot{x}_{d\parallel} + \dot{x}_{d\perp} \tag{4.25}$$

When the desired cartesian velocity violates more than a single joint velocity, two or more constraints become active.
The boundary planes can still be found easily one by one, the zero point is the only one to be always included in the feasible region for each of them.

A general solution to solve a case with multiple critical joints may be to search for the optimal solution on the intersection between the planes as shown in Figure 6, but this solution is not suitable to all particular cases.

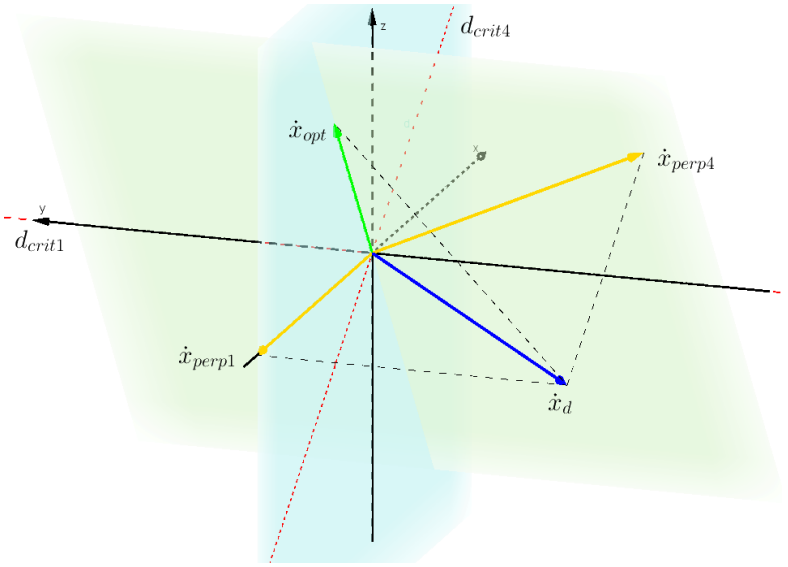


Figure 4.5: Multiple Critical Joints

For example there is the possibility to have two parallel boundaries which intersection is null, or is possible that the solution on the intersection is not the optimal with regard to the error.

For all these cases a more general method is required.

4.5 Third step: Quadratic Programming formulation

The proposed method attains the desired behaviour by formulating a Quadratic Programming problem.The achieved solution is the best feasible solution under an arbitrary number of constraints.

A generic QP problem is an optimization problem under linear constraints. It is defined by a quadratic objective function of the considered variables that we want to minimize, and by the constraints matrices of those variables.

4.5.1 QP classes

QP problems can be divided into four different classes:

Unconstrained QP $\min_x \left\{ \frac{1}{2} x^T W x + q^T x \right\}$	Box-constrained QP $\min_x \left\{ \frac{1}{2} x^T W x + q^T x \right\}$ $a \leq x \leq b$
Equality constrained QP $\min_x \left\{ \frac{1}{2} x^T W x + q^T x \right\}$ $Ax = a$	Inequality constrained QP $\min_x \left\{ \frac{1}{2} x^T W x + q^T x \right\}$ $Bx \leq b$

Table 4.1: QP classes

Unconstrained

Necessary and sufficient conditions for a local minimum:

- f has zero gradient at x^*

$$\nabla_x f(x^*) = 0$$

- the Hessian is semi-definite positive

$$w^T \nabla^2 f(x^*) w \geq 0 \ , \ \forall w$$

Equality Constrained

$$\begin{aligned} &\min_x \left\{ \frac{1}{2} x^T W x + q^T x \right\} \\ &s.t. \ Ax = a \end{aligned} \tag{4.26}$$

To solve a QP problem under equality constraints the Lagrange Multipliers method is commonly used. The Lagrange function is defined as follow:

$$\mathcal{L}(x, \lambda) = \left\{ \frac{1}{2} x^T W x + q^T x \right\} + \lambda^T (Ax - a) \tag{4.27}$$

If x^* is a local minimum for $\mathcal{L}$ then exists an unique λ^* that satisfies the conditions:

- $\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$
- $\nabla_\lambda \mathcal{L}(x^*, \lambda^*) = 0$
- $w^T \nabla^2 \mathcal{L}(x^*, \lambda^*) w \geq 0 \ , \ \forall w$

From the two first conditions is possible to define a linear system of two equations in two variables:

$$\begin{aligned} Qx + A^T \lambda &= -q \\ Ax &= a \end{aligned} \tag{4.28}$$

In matrix form:

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -q \\ a \end{bmatrix} \tag{4.29}$$

In this way the optimization problem is reduced to the solution of a system of linear equations.

For low dimension problems is possible to solve the system analytically by inverting the K matrix if it is not singular.

For higher dimension problems the inversion process could result computationally very expensive, in those cases usually is used a pre-conditioned conjugate gradient (PCG) method. This is an iterative algorithm used to find a numerical solution of particular system of linear equations whose matrix is symmetric and positive-definite.
The K matrix is always symmetric for its definition, in case it is not positive-definite is necessary to transform $Kx = y$ to $(K^T K)x = K^T y$ and then apply the PCG since the matrix $(K^T K)$ is always positive definite.

Inequality Constrained

In presence of inequality constraints the optimality conditions change and they are named KKT conditions/equations.

$$\begin{aligned} &\min_x \left\{ \frac{1}{2} x^T W x + q^T x \right\} \\ &s.t.\ Bx \leq b \end{aligned} \tag{4.30}$$

The Lagrange function:

$$\mathcal{L}(x, \mu) = \left\{ \frac{1}{2} x^T W x + q^T x \right\} + \mu^T (Bx - b) \tag{4.31}$$

If x^* is a local minimum for $\mathcal{L}$ then exists an unique μ^* that satisfies the KKT conditions:

- $\nabla_x \mathcal{L}(x^*, \mu^*) = 0$

- $\mu^* \geq 0$
- $\mu^*(Bx^* - b) = 0$
- $(Bx^* - b) \leq 0$
- $\mu^* \geq 0$
- $w^T \nabla^2 \mathcal{L}(x^*, \mu^*) w \geq 0 \ , \ \forall w$

4.5.2 Joint Limits as QP problem

In our case the unknown variable to minimize is the error in cartesian space while the velocity bounds represent the inequality constraints.

Objective Function

The objective function is the quadratic form of the cartesian error without any additional term.

$$\frac{1}{2}\varepsilon^T W \varepsilon \tag{4.32}$$

where

$$\dot{x}_f = \dot{x}_d + \varepsilon \tag{4.33}$$

If no constraints are violated the solution is always the zero point, so the QP problem introduces an error in the hand-guiding task only when really necessary.
Furthermore by minimizing the error with the desired commanded motion, the final solution will never produce any unexpected or unintuitive motion.

Inequality Constraints

As seen in a previous section each joint velocity limit can be formulated as a cartesian constraint.

For each limit all solutions that fall in the feasible region, delimited by the "boundary plane", are viable. The plane sets a limit for the cartesian component on the critical direction. The projection of the final solution on this direction must be lower or equal than the projection of the $\dot{x}_{\parallel_{max}} = \alpha\ \dot{x}_{d\parallel}$:

$$invJ_i\ \dot{x}_f \leq invJ_i\ \dot{x}_{\parallel_{max}} = \dot{Q}_{max_i} \tag{4.34}$$

Recalling Equation 4.33 is possible to formulate also this constraint in function of the error.

$$\begin{aligned} &invJ_i\ (\dot{x}_d + \varepsilon) \leq \dot{Q}_{max_i} \\ &invJ_i\ \varepsilon \leq \dot{Q}_{max_i} - invJ_i\ \dot{x}_d \\ &invJ_i\ \varepsilon \leq \dot{Q}_{max_i} - \dot{q}_{d_i} \end{aligned} \tag{4.35}$$

For each joint the constraint matrices will present two rows one for the maximum and one for the minimum velocities in the following form:

$$\begin{aligned} B_i &= invJ_i \qquad b_i = \dot{Q}_{max_i} - \dot{q}_{d_i} \\ B_i &= -invJ_i \quad b_i = -\dot{Q}_{min_i} + \dot{q}_{d_i} \end{aligned} \tag{4.36}$$

The lower limits formulated just by changing all the signs of the equation 4.35.

The final matrices can be written as

$$\begin{aligned} B &= \begin{bmatrix} invJ \\ -invJ \end{bmatrix} \\ b_i &= \begin{bmatrix} \dot{Q}_{max} - \dot{q}_d \\ -\dot{Q}_{min} + \dot{q}_d \end{bmatrix} \end{aligned} \tag{4.37}$$

The Appendix A.2 contains the function used to initialize those matrices and fully define the QP problem.

4.5.3 QP Solver - Active Set Method

All the already existing Matlab solvers make use of structures or cell arrays which are not allowed in simulink modules for real time code generation where static memory allocation is required.

Then I developed a QP solver, as an embedded Matlab function, following the so called Active-Set method.

Active set method

The active set method is an iterative algorithm used to solved QP problem. It can ensure fast convergence to the optimal solution for low dimension problems while for high dimensions different approaches can be suggested as the Interior Point Method.

The algorithm starts from an initial point x^0, that must be a feasible point for the problem, and it finds the next iterate by setting

$$x^{k+1} = x^k + \alpha_k d^k \tag{4.38}$$

Here α_k is the step-length and d^k is the search direction.

At each step the indexes of the active constraints for the current point x^k are stored in the so called Active Set:

$$\mathcal{A}^k = \{j \mid B_j^T x^k - b_j = 0\} \tag{4.39}$$

To determine the search direction d^k solve the equality constrained QP, sub-

ject to the active set and the equality constraints, if there are any.

$$\begin{aligned} &\min_d \left\{ \frac{1}{2} d^T W d + q^T d \right\} \\ &s.t.\ A_j^T d = a \\ &\qquad B_j^T d \leq b\ ,\ j \in \mathcal{A}^k \end{aligned} \tag{4.40}$$

The KKT optimality conditions lead to the matrix system:

$$\begin{bmatrix} Q & A^T & \tilde{B}^T \\ A & 0 & 0 \\ \tilde{B} & 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ \lambda \\ \tilde{\mu} \end{bmatrix} = \begin{bmatrix} -q \\ 0 \\ 0 \end{bmatrix} \tag{4.41}$$

If $d^k = 0$:

- If $\tilde{\mu}_k \geq 0$, x^k is an optimal point, stop the algorithm.
- If at least one $\tilde{\mu}_k \leq 0$ then x^k is not an optimal solution, some of the active constraints are not necessary for the final solution. So the constraint with the lower Lagrangian multiplier $\tilde{\mu}_j$ is removed from the active set and the problem is solved again.

If $d^k \neq 0$:

the algorithm can continue with the computation of the new step. The step-length α_k must guarantee that the new step $x^k + \alpha_k d^k$ is feasible to all the constraints. A common formulation is the following one:

$$\alpha_k = \min \left\{ 1\ ,\ \frac{b_j - B_j^T x^k}{B_j^T d^k} \mid j \notin \mathcal{A}^k \ and\ B_j^T d^k > 0 \right\} \tag{4.42}$$

It is important to observe that if $\alpha_k < 1$, then $\alpha_k = \dfrac{b_{j_0} - B_{j_0}^T x^k}{B_{j_0}^T d^k}$ for some $j_0 \notin \mathcal{A}^k$. This implies that $B_{j_0}^T(x^k + \alpha_k d^k) = b_{j_0}$ and so the inequality constraints corresponding to j_0 becomes active at the next step. The active

set has to be update by adding the index $\mathcal{A}^{k+1} = \mathcal{A}^k \cup \{j_0\}$.

In general the Active Set method is a proper solution for low dimension problems. The biggest issue is how to determine a good starting iterate x^0 that is required to be inside all constraints.

In our case the starting ε is set equal to $-\dot{x}_d$. In this way the corresponding cartesian velocity is:

$$\dot{x}_{f_{intial}} = \dot{x}_d + \varepsilon = 0 \tag{4.43}$$

that is obviously a generic feasible solution for every possible limit $\dot{Q}_{max}$ and $\dot{Q}_{min}$.

Active Set Method - Pseudo Code

1. Give a feasible start vector x^0
2. Identify the active set $\mathcal{A}^0$
3. Set $k = 0$
4. While (no convergence) do
5. Compute $g^k = Qx^k + q$:
6. Obtain $d^k, \lambda^k, \tilde{\mu}^k$ by solving the KKT equations for:

$$\begin{aligned} &\min_d \left\{ \frac{1}{2} d^T W d + q^T d \right\} \\ &s.t.\ A_j^T d = a \\ &\qquad B_j^T d \leq b\ ,\ j \in \mathcal{A}^k \end{aligned} \tag{4.44}$$

7. If $(d^k = 0)$
8. If $(\tilde{\mu}^k \geq 0)$

9. Stop: x^k is a KKT optimal point

10. Else

11. Compute the minimum negative Lagrangian multiplier:

$$\tilde{\mu}^k_{j_0} = \min\{\tilde{\mu}^k_j \mid \tilde{\mu}^k < 0,\ j \in \mathcal{A}^k\} \tag{4.45}$$

12. Update the index set $\mathcal{A}^k \to \mathcal{A}^k \backslash \{j_0\}$ and return to step 6

13. End if

14. End if

15. If $(d^k \neq 0)$

16. Compute the step length

$$\alpha_k = \min\left\{1\ ,\ \frac{b_j - B_j^T x^k}{B_j^T d^k} \mid j \notin \mathcal{A}^k \ and\ B_j^T d^k > 0\right\} \tag{4.46}$$

17. Update $x^{k+1} = x^k + \alpha_k d^k$

18. Update Active Set:

- if $\alpha_k = 1$, then $\mathcal{A}^{k+1} = \mathcal{A}^k$
- else $\mathcal{A}^{k+1} = \mathcal{A}^k \cup \{j_0\}$, where $\alpha_k = \dfrac{b_{j_0} - B_{j_0}^T x^k}{B_{j_0}^T d^k}$ for $B_{j_0}^T d^k > 0$

19. Update $k \to k+1$

20. End if

21. End while

Appendix A.3 encloses the integral embedded matlab function used in the simulink model.

4.5.4 Second Formulation

It is possible to formulate the QP problem in a different way to directly get the joint speeds as solution without the necessity to compute the inverse of the free Jacobian neither the nullspace matrix of the cartesian constraints.

Objective function

$$\frac{1}{2}\begin{bmatrix}\varepsilon\\ \dot{q}\end{bmatrix}^T W \begin{bmatrix}\varepsilon\\ \dot{q}\end{bmatrix} \tag{4.47}$$

In this formulation the variable to minimize is composed by two terms. Again the term with the highest minimization priority is the cartesian error ($\varepsilon = \dot{x}_f - \dot{x}_d$), while there is an additional second term that is represented by joint velocities $\dot{q}$.

Joint velocities $\dot{q}$ act as a regularization term in the problem, it's more important to achieve the desired cartesian task than minimize them. Just in case more than one solution will produce the same minimum cartesian error, then the algorithm will choose the one minimizing joint velocities.

To give higher minimization priority to ε the diagonal weights matrix W is defined as follow:

$$W = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & 10^{-5} & & \\ & & & & \ddots & \\ & & & & & 10^{-5} \end{bmatrix} \tag{4.48}$$

Constraints

Equality:

$$\begin{aligned} J_{hg}\dot{q} - \varepsilon &= \dot{x}_d \\ J_c\dot{q} &= 0 \end{aligned} \tag{4.49}$$

Here the first equation is necessary to define the error in the cartesian space, it can be seen as a soft constraints because ε is a variable term.

The second equation is an hard constraint related to the kinematic constraint defined by the chosen motion mode.

Inequality:

$$\begin{aligned} \dot{q} &\leq \dot{Q}_{max} \\ -\dot{q} &\leq -\dot{Q}_{min} \end{aligned} \tag{4.50}$$

Also in this formulation inequality constraints are bounded by minimum and maximum joint velocities, but this time they can be directly formulated in joint space, so for this case it's not necessary the computation of the inverse of the Jacobian.

4.6 Performance Analysis

The two formulations produced almost identical outputs in simulations and when tested on the robot in non singular configurations.
The singularity detection of the module is included in the computation of the inverse of the Jacobian, so the second solution doesn't have any protection for singular configuration, since it doesn't required the computation of the inverse of the Jacobian. At singular point when tested the robot produced big oscillations while running this second solution.
To find a practical application the second solution requires an additional development of a specific singularity handling solution.

Anyway a comparison between the two solutions was made in order to asses

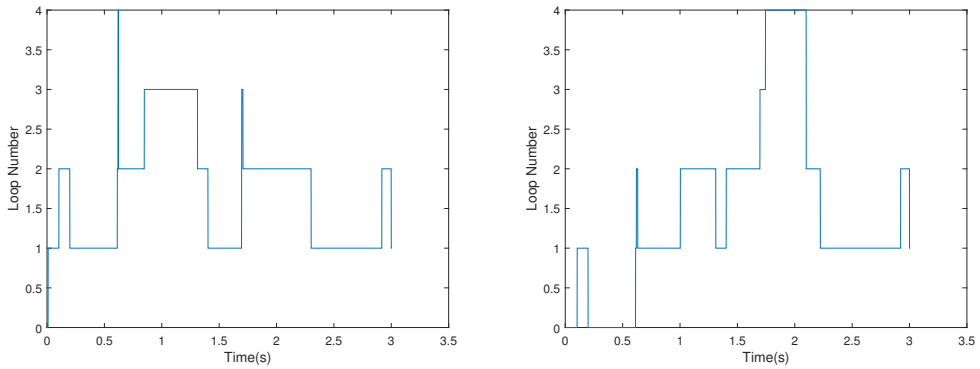
which one can guarantee lower computation time. Performances are particularly important for real-time application where the sampling time is in the order of milliseconds.
The two algorithm have been compared on the same trajectory, simulated for three times in the Simulink module. The "tic" and "toc" functions have been used to estimate the required computational time for each step.

First formulation:	Second formulation
$$\min\left\{\frac{1}{2}\varepsilon^T W \varepsilon\right\}$$ $$\begin{bmatrix} J_{hg}^{-1} \\ -J_{hg}^{-1} \end{bmatrix} \varepsilon \leq \begin{bmatrix} \dot{Q}_{max} - \dot{q}_d \\ -\dot{Q}_{min} + \dot{q}_d \end{bmatrix}$$	$$\min\left\{\frac{1}{2}\begin{bmatrix}\varepsilon \\ \dot{q}\end{bmatrix}^T W \begin{bmatrix}\varepsilon \\ \dot{q}\end{bmatrix}\right\}$$ $$\begin{bmatrix} -I & J_{hg} \\ 0 & J_c \end{bmatrix}\begin{bmatrix}\varepsilon \\ \dot{q}\end{bmatrix} = \begin{bmatrix}\dot{x}_d \\ 0\end{bmatrix}$$ $$\begin{bmatrix} 0 & I \\ 0 & -I \end{bmatrix}\begin{bmatrix}\varepsilon \\ \dot{q}\end{bmatrix} \leq \begin{bmatrix}\dot{Q}_{max} \\ -\dot{Q}_{min}\end{bmatrix}$$
– No equality constraints – Lower dimension problem – Requires computation of invJ	– Equality constraints – Higher dimension problem – No computation of invJ

Table 4.2: QP formulations

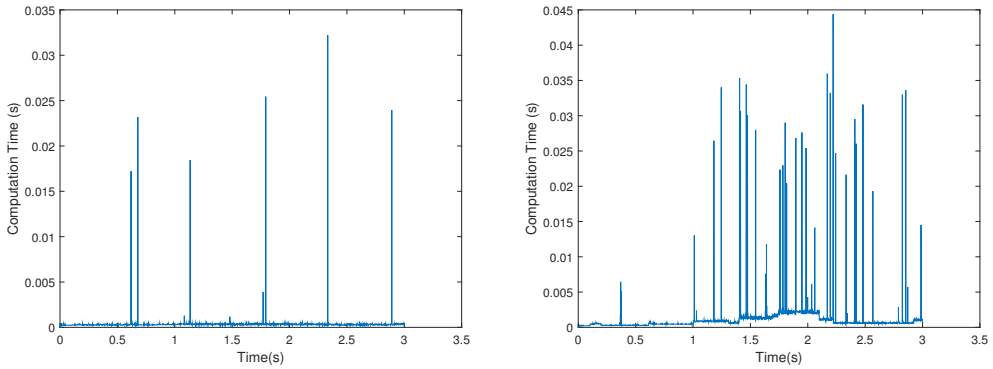
Figure 4.6 shows the number of loop required to the solver at each step. There are a lot of punctual differences between the two graphs, but in general it's not possible to asses if one has faster convergence than the other. So the same configuration may required a different number of loops for the two problems but none of them can be considered always better.

Figure 4.7 shows a first rough comparison of the performances in term of computation time. From these rough data is possible to notice how the sec-



(a) 1_{st} Loop Count (b) 2_{nd} Loop Count

Figure 4.6: Number of Loop



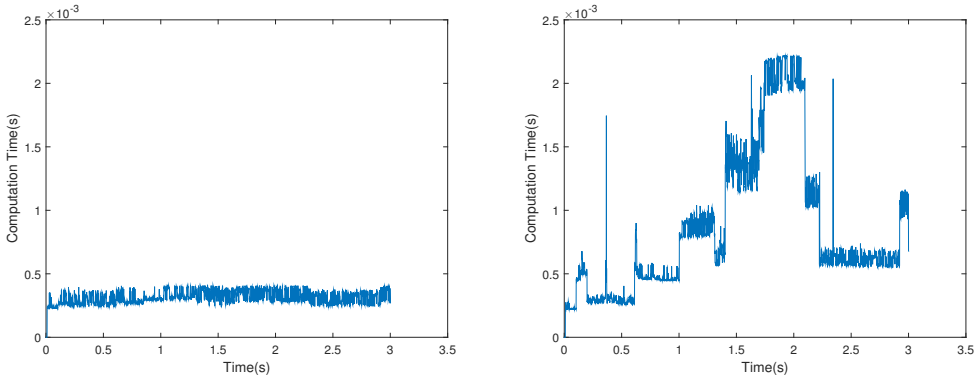
(a) 1_{st} Computation Time (b) 2_{nd} Computation Time

Figure 4.7: Performances

ond one has higher instability and how it presents a bigger number of outliers.

Figure 4.8 shows the same data cut at their 95 percentile, in this way the outliers had been removed. It's noticeable that the first formulation can guarantee better and more stable performances.

In the first solution the inversion of the Jacobian is required once at the beginning of the algorithm, then the number of loop required to the solver doesn't have a big influence on the final time, that is kept almost constant for



(a) 1_{st} Computation Time 95 (b) 2_{nd} Computation Time 95

Figure 4.8: Performances 95 percentile

the entire simulation. In the second solution each loop of the solver required to solve a more complex QP problem with higher dimension KKT-equations, then the total time is greatly influenced by the number of loops and its profile faithfully follows Figure 4.6b.

First formulation:	Second formulation
$\bar{\mu}_{1_{st}} = 0.312 \cdot 10^{-3}\ s$ $\bar{\sigma}_{1_{st}} = 5.122 \cdot 10^{-5}\ s$	$\bar{\mu}_{2_{nd}} = 0.806 \cdot 10^{-3}\ s$ $\bar{\sigma}_{2_{nd}} = 5.136 \cdot 10^{-4}\ s$

Table 4.3: Performances: statistical indicators

The table 4.3 contains the mean value and the standard deviation for the two problems. The second mean, that is more than double the first one, reflects the general worse performances of this second solution, while its higher variance is due to the strong dependency on the number of loops, a lower variance is always preferable cause it reflects a more stable and predictable model.

Chapter 5

Testing and Evaluation

5.1 Hypothesis

In the testing phase we want to evaluate this new solution in terms of:

- hardware stress
- usability

while respecting all the requirements.

During the test we compared our QP solution with the old version of the software, that realized a simple Task Priority control, and with the most interesting and popular state of the art work, Saturation in the Nullspace.

1. Task Priority (original solution):
   - Advantages: possibility to slide along joint limits
   - Drawbacks: hard stop without any velocity control
2. Saturation in the Nullspace (from literature):
   - Advantages: velocity control, smooth approach to the position limit
   - Drawbacks: single scaling factor for the entire task, freeze of all joints when the position limit is hit

5.2 Experiment design

The testing phase was entirely accomplished at the KUKA headquarters in Augsburg at the Med team laboratory.

Subjects

All the subjects have been recruited between KUKA office employees for a total of 42 testers. The final group was anyway rather various in terms of ages, gender and experience with hand-guided robot. In table 5.1 all the generic personal data are displayed as total occurrences.

Age	20-29 [*20*]	30-39 [*19*]	40+ [*3*]	
Gender	M [*28*]	F [*14*]		
Dominant hand	Left [*5*]	Right [*37*]		
Hand-guided robot experience	Never [*11*]	<5 trials [*8*]	<20 trials[*4*]	Regular [*19*]

Table 5.1: General subject data

Protocol

Each subject has been asked to grab the tcp of the robot with his dominant hand while standing in front of it. The task assigned was to follow with the tcp a semicircular trajectory previously drawn on a simple paper sets at the basis of the robot as shown in Figure 5.2.
The motion of the robot for the testing phase was constrained on the plane perpendicular to the ground, just in order to avoid unnecessary movements

Figure 5.1: Robot position

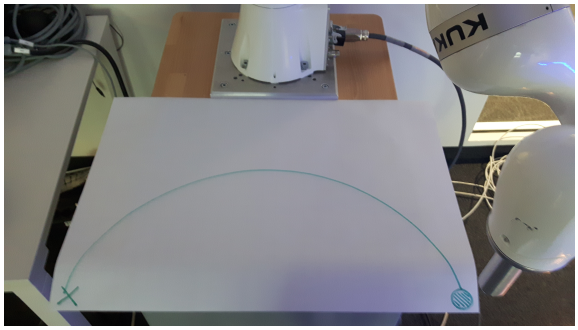


Figure 5.2: Trajectory line

from the subjects and to simplify the task. The starting point was fixed few centimetres away from the table over one end of the line.
Each subject repeated the trajectory task three times with each algorithm for a total of nine trials. The order of the algorithms was randomly changed between different subjects in order to ensure a fair and not biased evaluation. The trajectory was constructed in order to lead the subject into the positional limit of the fourth joint (elbow joint), but before the start of the test the subjects were instructed that at some point they wouldn't be able to exactly follow the trajectory, at this point they just had to continue the motion as close as possible to it.

Data acquisition

Both quantitative and qualitative data were gathered.
During each trial the running Java application saved a set of data acquired from the motor sensors. Each set is composed by:

- Cartesian position, velocity and acceleration of the tcp (3 dimensional)
- Joint position, velocity, acceleration and torques (7 degree of freedom)

Furthermore at the end of the nine trials each subject was asked to compile a short questionnaire about the usability impression they had with each algorithm with respect to the others. The questionnaire is composed by four main questions, listed below, and a following blank space to collect any further impression or suggestion from the users. The first three questions required to assign a score from 1 to 5 to each algorithm, while the fourth one to establish a ranking between them based on general impression. A sample of the questionnaire can be found in the appendix B.

- It was easy and natural to perform the required task (trajectory following); use scale 1 (very difficult) – 5 (very easy/natural)
- I'm satisfied by the amount of time required to perform the task; use scale 1 (much longer than expected) – 3 (as expected) - 5 (faster than expected)
- I didn't notice any unexpected movements/stop of the robot (abrupt movements or waving); use scale 1 (a lot of unexpected movements) – 5 (no unexpected movements)
- Which algorithm would you prefer from your general impression? assign a rank [1-3], where 1 is the favourite

5.3 Quantitative data analysis

In this section for each of the tested algorithms one sample set of data was chosen to be reported. Since subjects approached the experiment with different attitude and past experience on hand-guided robot, the results recorded

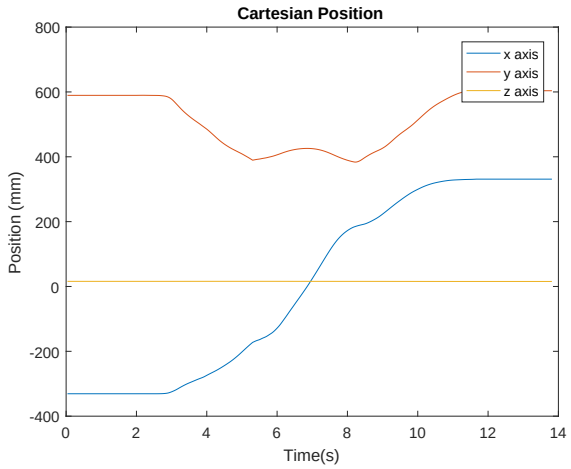
are quite various but the selected data set is representative of the most common pattern encountered during all the performed trials.

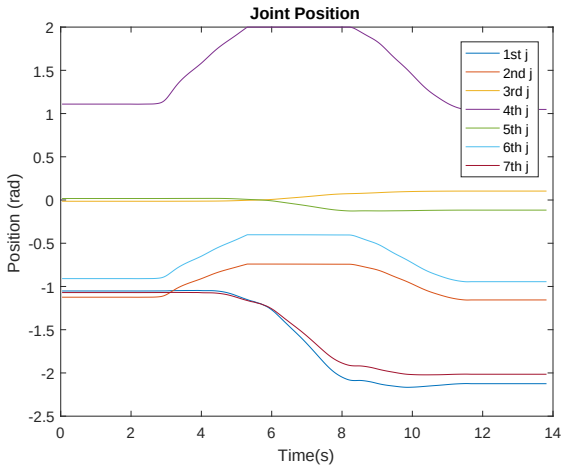
5.3.1 Algorithm A: Task Priority (old version)

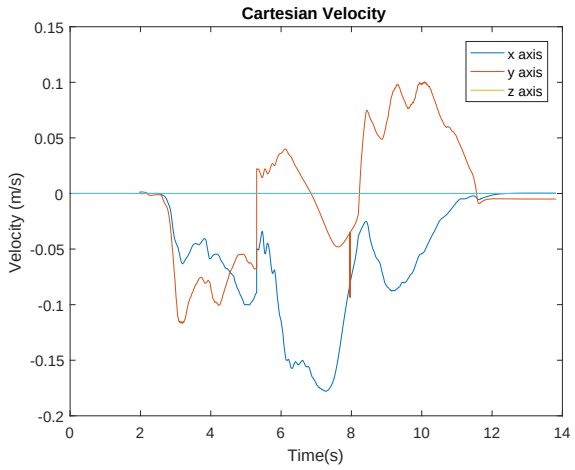
Solution A presents a very regular position series 5.3.1, in the middle of the graphs is possible to appreciate how after reaching the 4th limit position the motion continued around it (4th joint position saturated) without any interruption. Also the cartesian position is very smooth showing that for the subjects was easy to perform the task even after they were stopped by the limit.

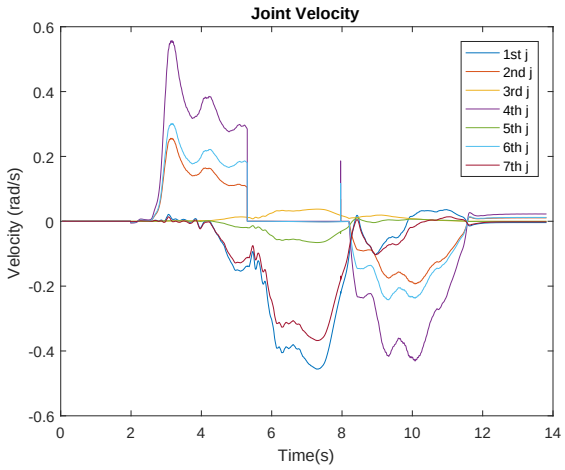
Velocity graph 5.3.1 shows two big discontinuity points when the user hit the limit and when he leaves it at the end. These points are easier to be visualized from the acceleration graph 5.3.1 where two big peaks are visible.

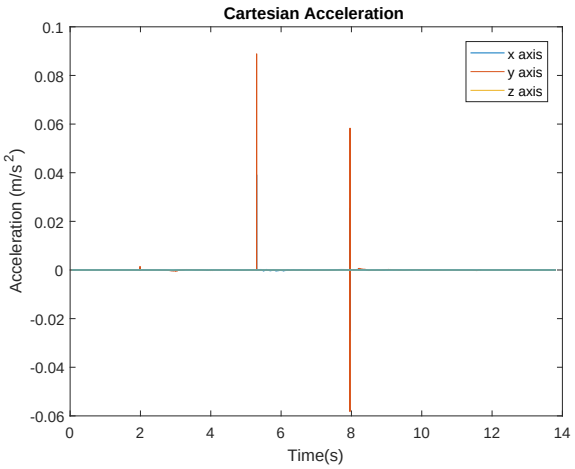
The resulting torques 5.3.1 produced to compensate the required acceleration of the robot present the same kind of peaks with the highest value recorded on the 4th joint when the position limit is hit.

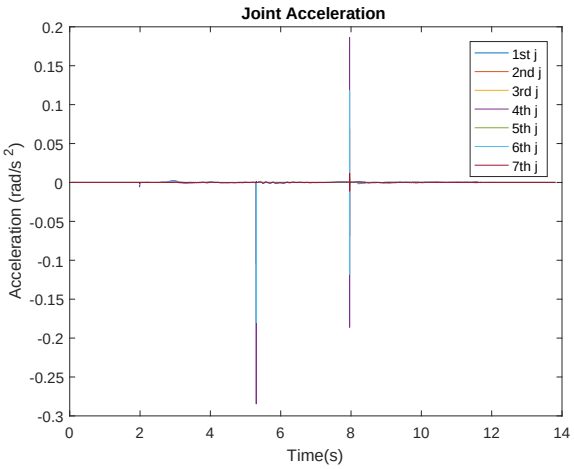


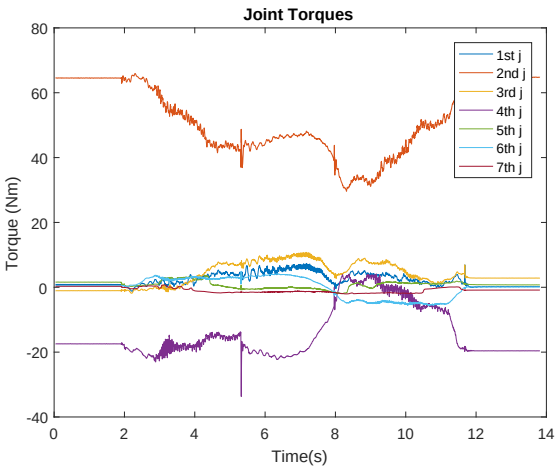


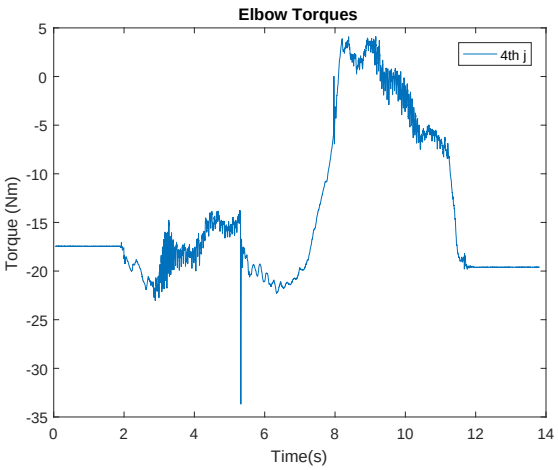

Cartesian Velocity
x axis
y axis
z axis
Velocity (m/s)
Time(s)


Joint Velocity
1st j
2nd j
3rd j
4th j
5th j
6th j
7th j
Velocity (rad/s)
Time(s)


Cartesian Acceleration
x axis
y axis
z axis
Acceleration (m/s 2)
Time(s)


Joint Acceleration
1st j
2nd j
3rd j
4th j
5th j
6th j
7th j
Acceleration (rad/s 2)
Time(s)

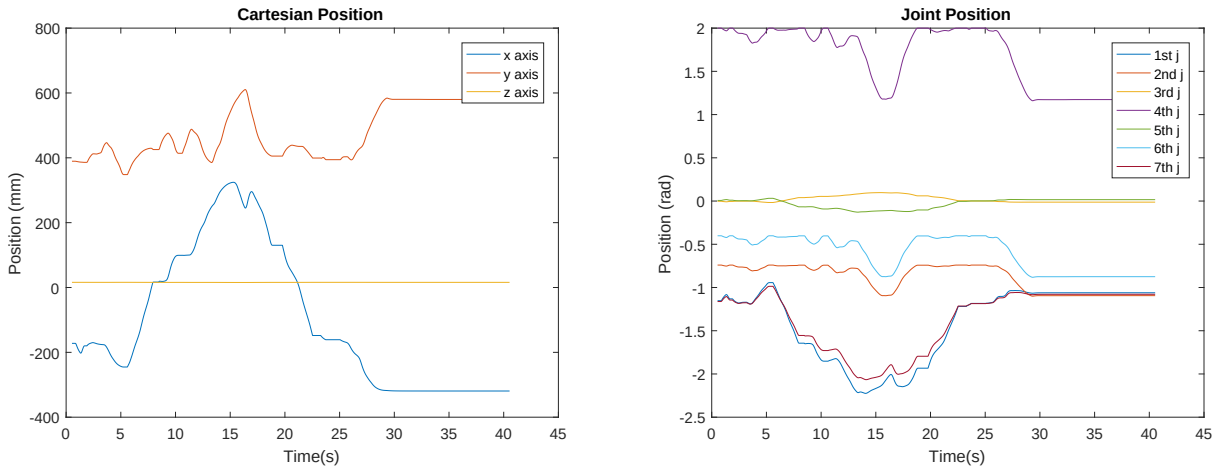

Joint Torques
1st j
2nd j
3rd j
4th j
5th j
6th j
7th j
Torque (Nm)
Time(s)

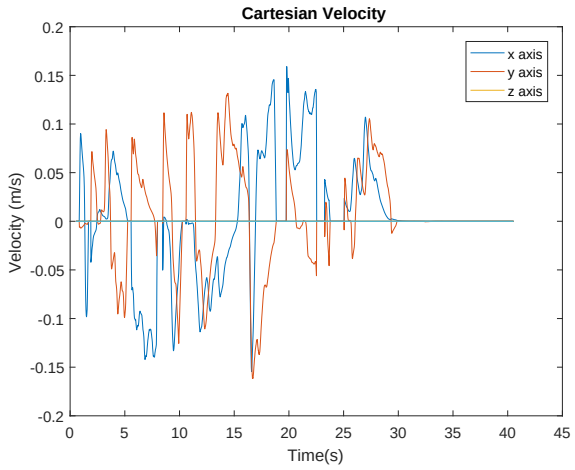

Elbow Torques
4th j
Torque (Nm)
Time(s)

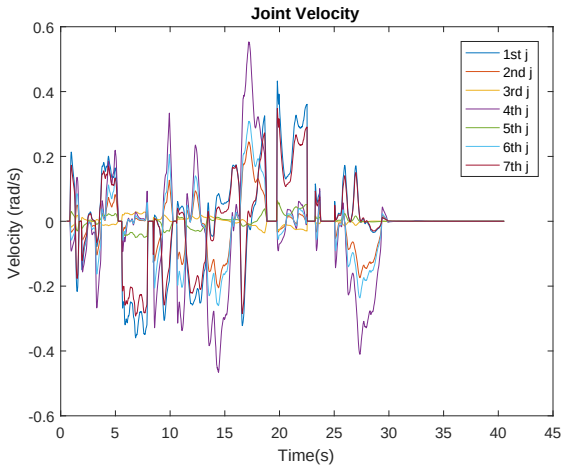
5.3.2 Algorithm B: Saturation in the Nullspace (from literature)

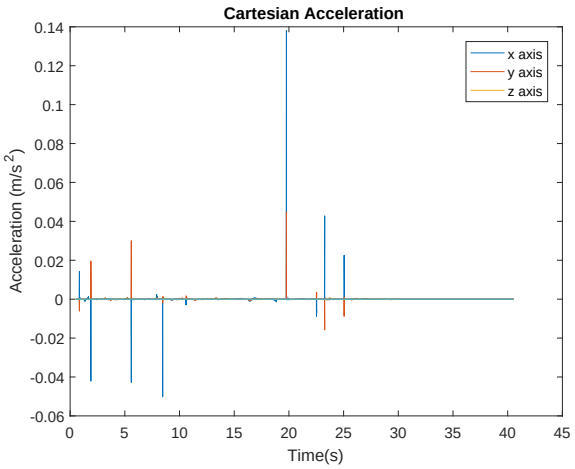
The biggest drawback of algorithm B is the impossibility to slide along the limit. As we can notice in 5.3.2 after the saturation of the joint position limit the user had to manually move the robot back away from the limit and then try to continue the motion around hit. More expert users were able to move around the limit with regular waving movements while less experience users tried to force the motion creating abrupt and rough movements of the robot.

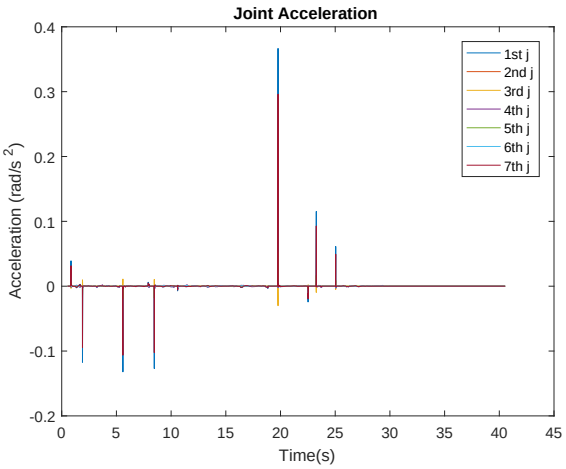
For this method acceleration 5.3.2 and torques peaks are not related to the 4th joint limit but are created by the user while trying to move away from it on the other joints. This behaviour is observable from figure 5.3.4 where elbow acceleration are reported and they always remain in the order of $10^{-3} rad/s^2$.

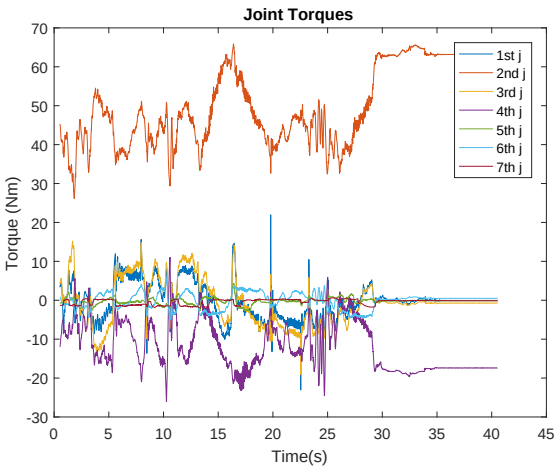


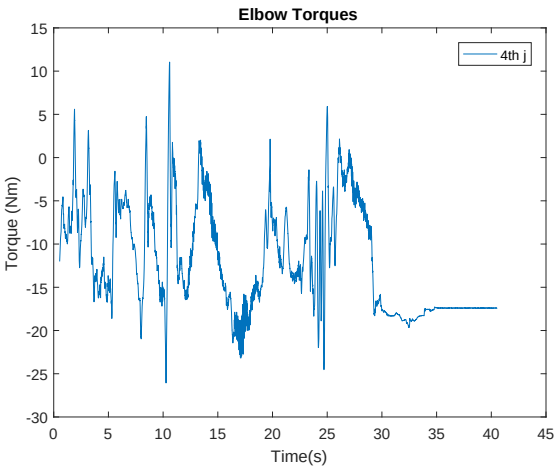

Cartesian Velocity
x axis
y axis
z axis
Velocity (m/s)
Time(s)


Joint Velocity
1st j
2nd j
3rd j
4th j
5th j
6th j
7th j
Velocity (rad/s)
Time(s)


Cartesian Acceleration
x axis
y axis
z axis
Acceleration (m/s 2)
Time(s)


Joint Acceleration
1st j
2nd j
3rd j
4th j
5th j
6th j
7th j
Acceleration (rad/s 2)
Time(s)

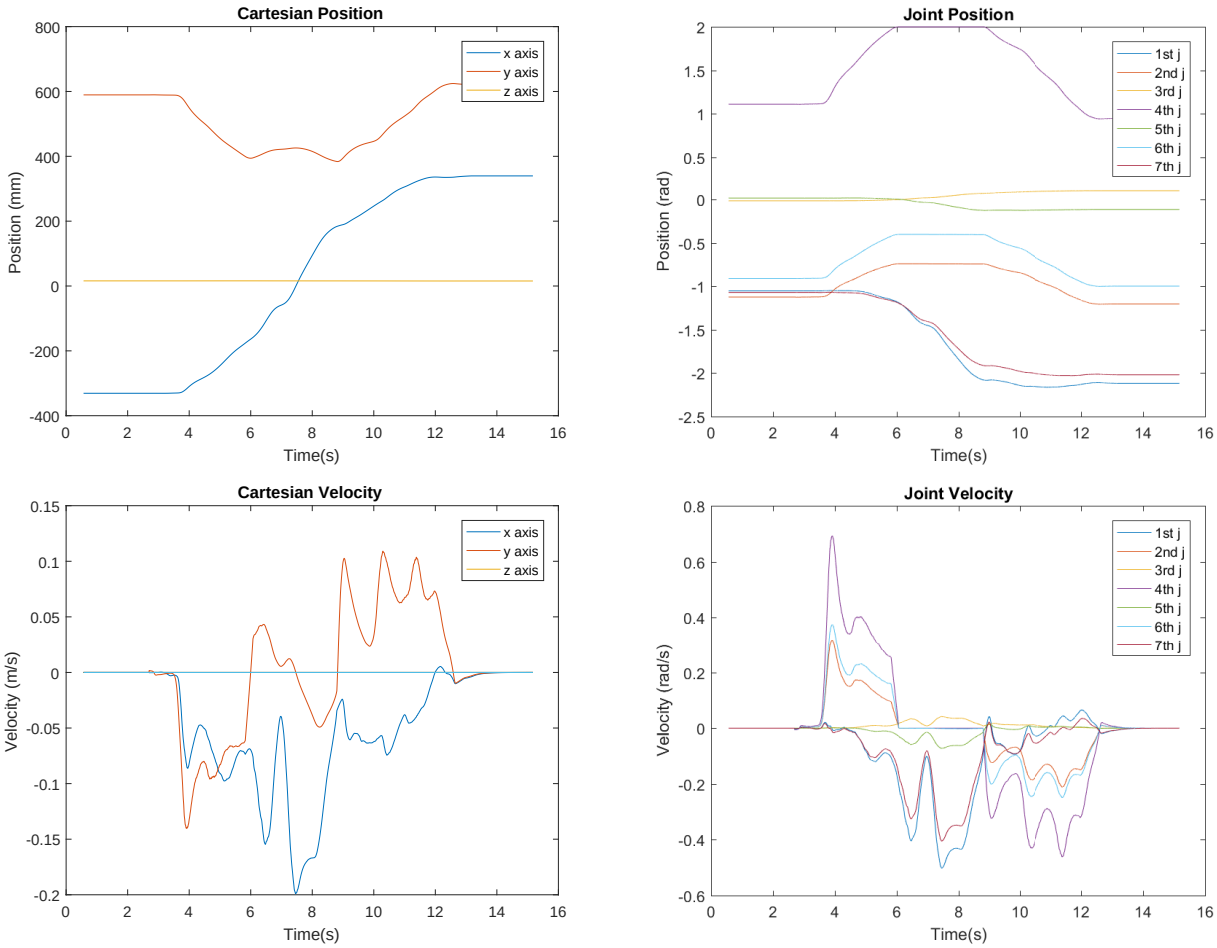

Joint Torques
1st j
2nd j
3rd j
4th j
5th j
6th j
7th j
Torque (Nm)
Time(s)

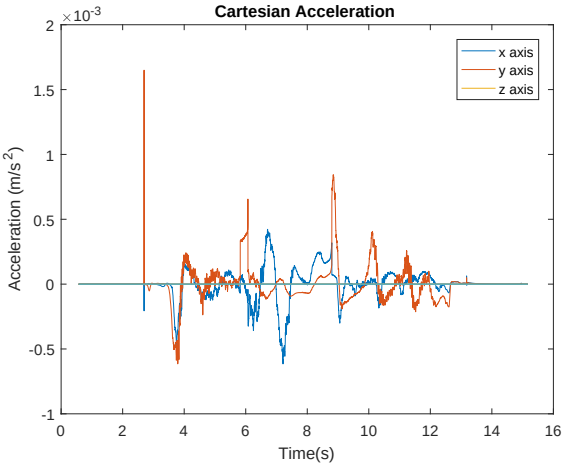

Elbow Torques
4th j
Torque (Nm)
Time(s)

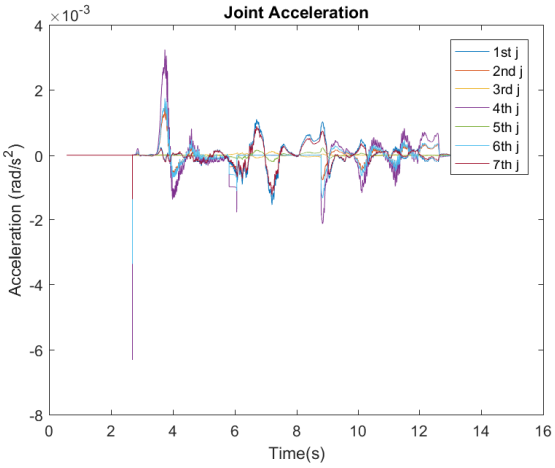
5.3.3 Algorithm C: Developed QP (my work)

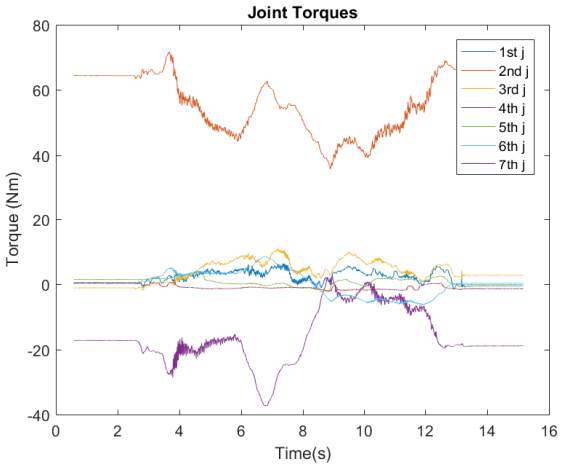
The solution developed for this thesis should be able to eliminate the drawbacks of the other algorithms and should ensure at the same time the possibility to slide along the joint limit and a smooth approach into it.

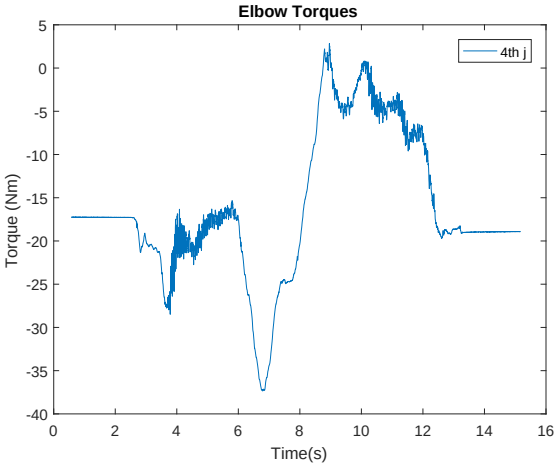
As expected the position graphs 5.3.3 are very similar to algorithm A, they remain very regular and the possibility to slide along the limit is noticeable. This time no discontinuities points can be found in the velocity graph 5.3.3 as their acceleration values 5.3.3 remains three orders of magnitude lower than for the previous algorithms, $10^{-3} rad/s^2$ and $10^{-3} m/s^2$ respectively.




Cartesian Acceleration
×10-3
x axis
y axis
z axis
Acceleration (m/s2)
Time(s)

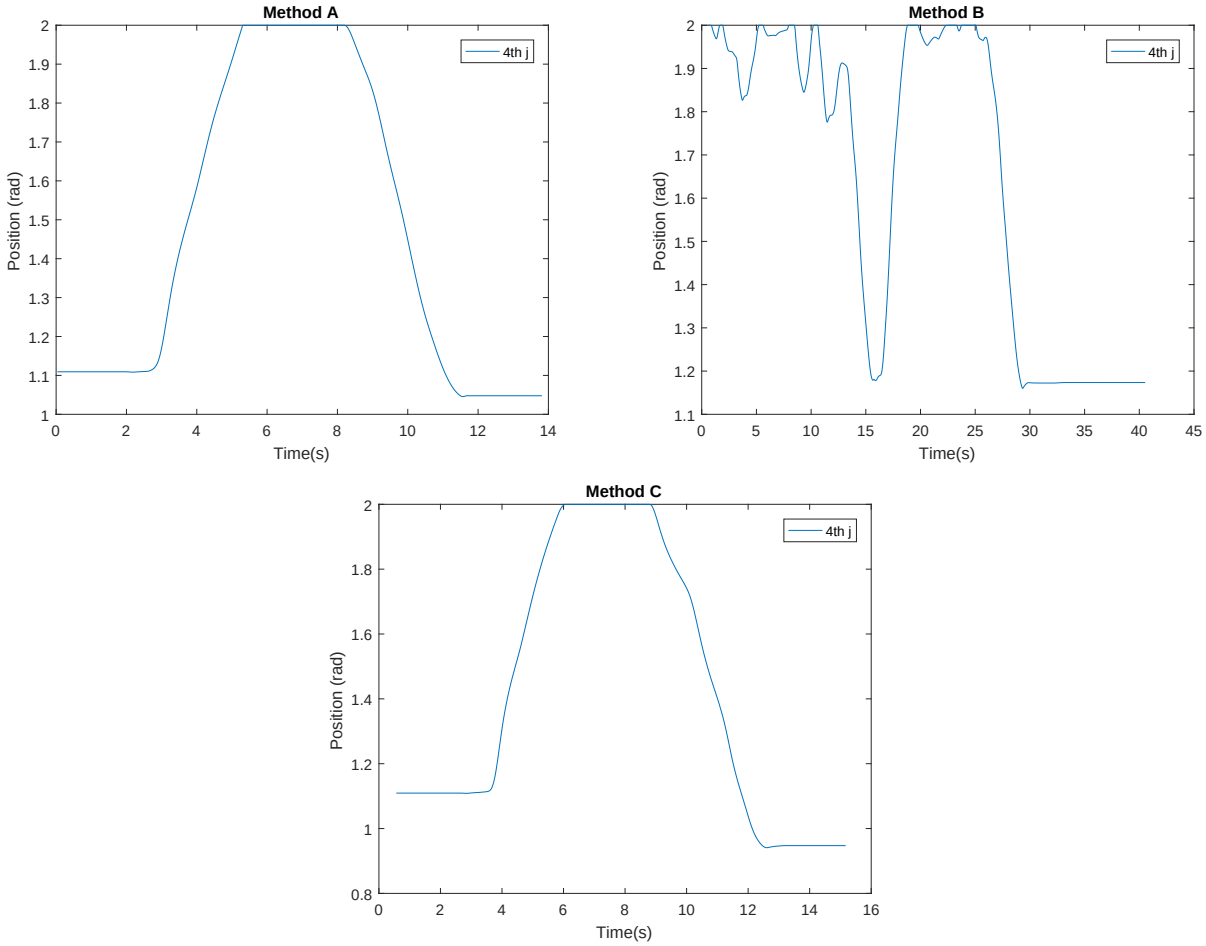

Joint Acceleration
×10-3
1st j
2nd j
3rd j
4th j
5th j
6th j
7th j
Acceleration (rad/s2)
Time(s)


Joint Torques
1st j
2nd j
3rd j
4th j
5th j
6th j
7th j
Torque (Nm)
Time(s)

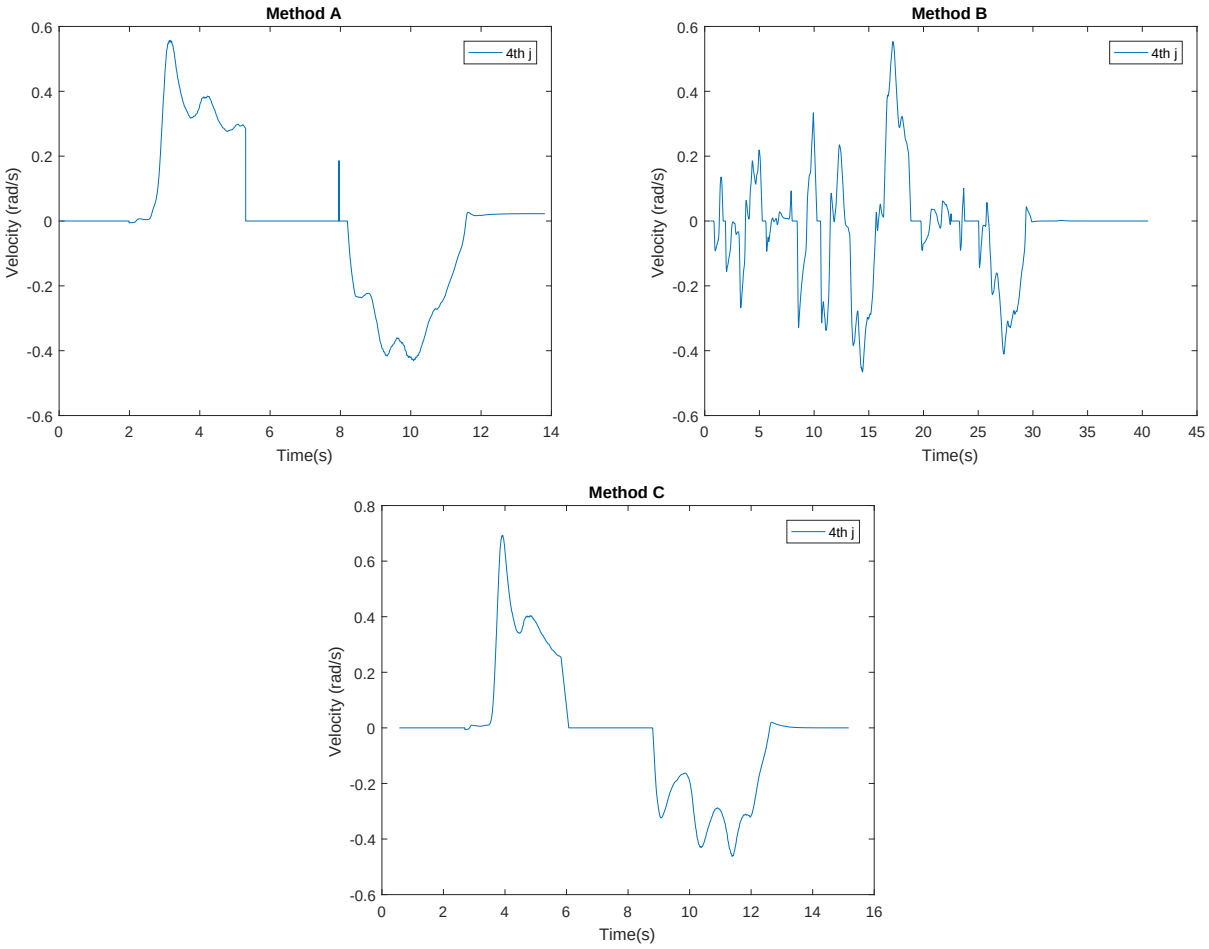

Elbow Torques
4th j
Torque (Nm)
Time(s)

5.3.4 Comparative 4th joint overview

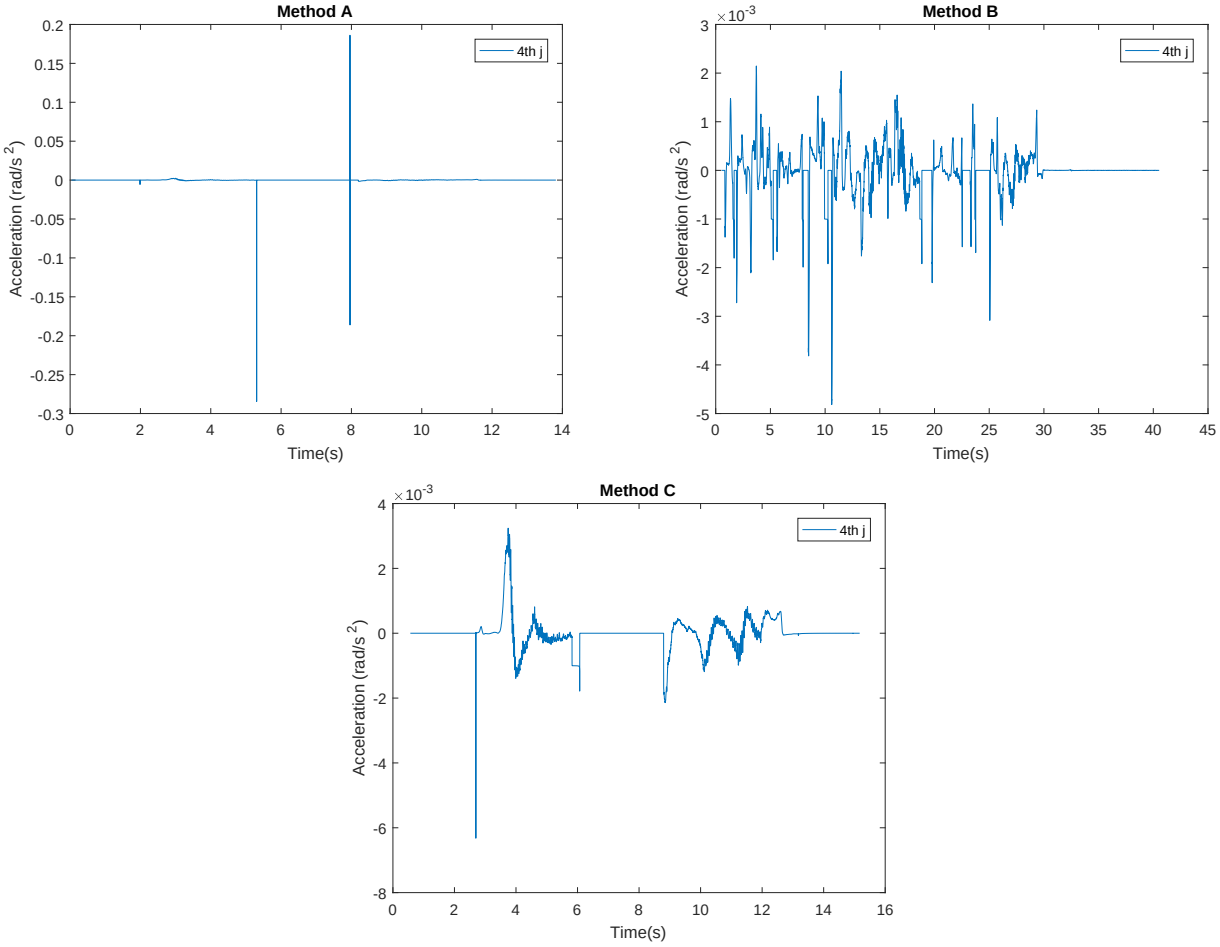
Elbow Position



Elbow Velocity



Acceleration



5.3.5 Conclusions

The developed algorithm can guarantee a stable and safe control over the joint velocities, that result in the achievement of a smooth approach to the joint limits with no discontinuity points.

The produced accelerations resulted to be three order of magnitude lower than with the other two algorithms and the produced torques never present the typical peaks recorded with the other solutions.

Under a continue and intense use the hardware will be subject to lower stress that may result in less maintenance and longer life span.

5.4 Qualitative data analysis

In this section results from the usability questionnaire are presented.
The assigned scores are displayed in three separated tables as total occurrences 5.3 5.4 5.5.

Algorithm A: Task Priority (old version)					
Scores:	1	2	3	4	5
It was esay and natural to perform the required task	0	2	17	19	4
I'm satisfied by the amount of time required to perform the task	0	5	18	14	5
I don't notice any unexpected or undesirable movemennts of the robot (abrupt movements or waving)	0	3	15	13	7

Figure 5.3: Algorithm A: questionnaire scores

Later the data had been grouped into three histograms where score frequencies from different algorithms are placed side by side 5.6 5.7 5.8.
The fourth histogram 5.9 shows data from the ranking question where each column represents the number of times the related algorithm have been placed in this rank position.

5.4.1 Statistical Inference

Due to the small sample size and ordinal data, a non-parametric statistical significance test was necessary to compare the scores received from each algorithm. The Median Test was selected for statistical inference, because it is considered a solid approach in presence of ordinal data set where none a priori

Algorithm B: Saturation in the Nullspace (from literature)					
Scores:	1	2	3	4	5
It was esay and natural to perform the required task	19	18	4	1	0
I'm satisfied by the amount of time required to perform the task	16	17	9	0	0
I don't notice any unexpected or undesirable movemennts of the robot (abrupt movements or waving)	28	10	2	2	0

Figure 5.4: Algorithm B: questionnaire scores

Algorithm C: Developed QP (my work)					
Scores:	1	2	3	4	5
It was esay and natural to perform the required task	0	1	9	17	15
I'm satisfied by the amount of time required to perform the task	0	4	15	17	6
I don't notice any unexpected or undesirable movemennts of the robot (abrupt movements or waving)	0	1	5	13	23

Figure 5.5: Algorithm C: questionnaire scores

hypothesis is needed. The null hypothesis states that no significance different can be assessed between the central tendency of the two sets. Statistically significant effects were assessed at $p < 0.05$.

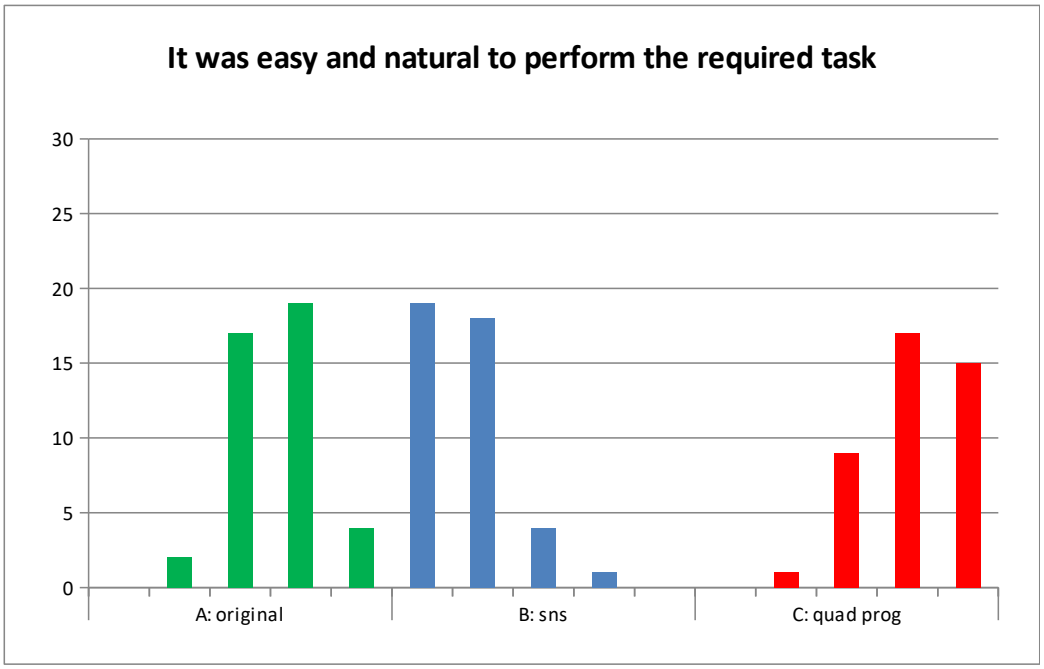


Figure 5.6: First question histogram

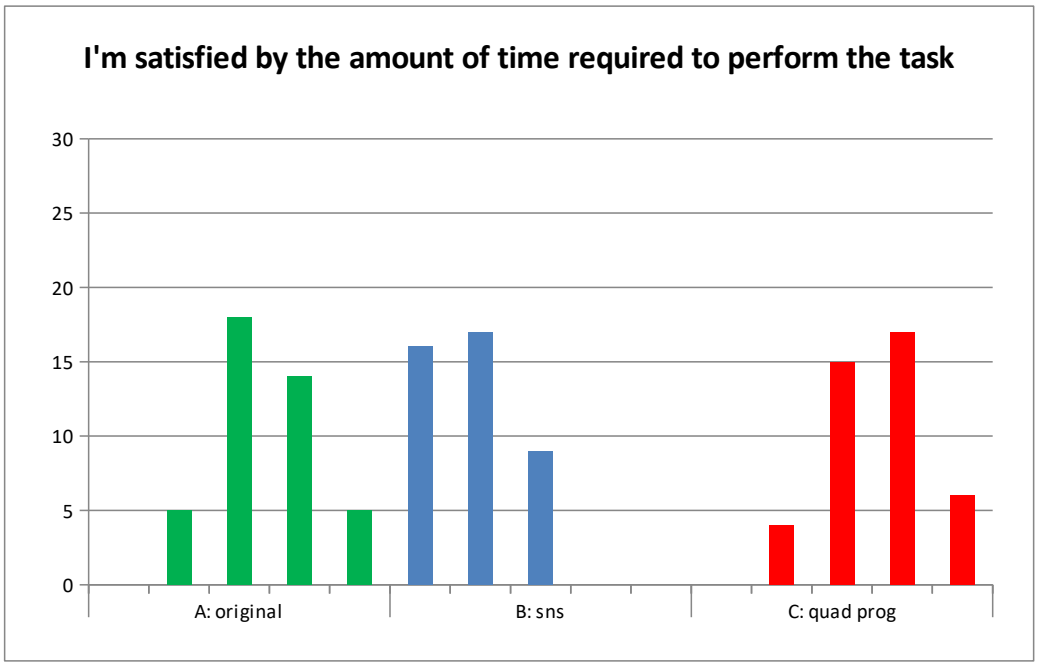


Figure 5.7: Second question histogram

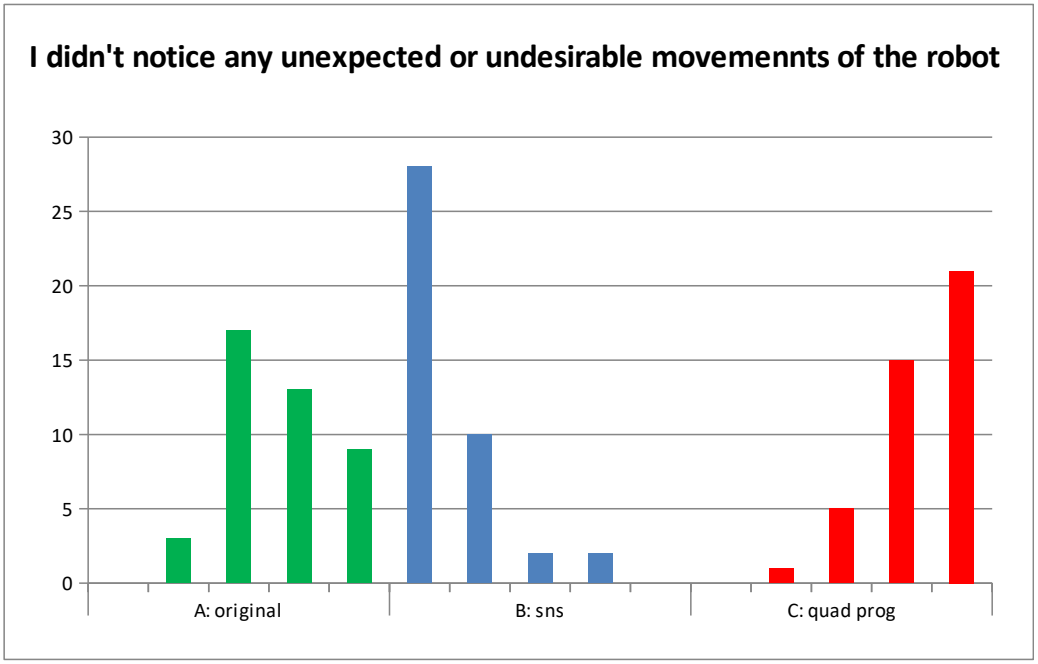


Figure 5.8: Third question histogram

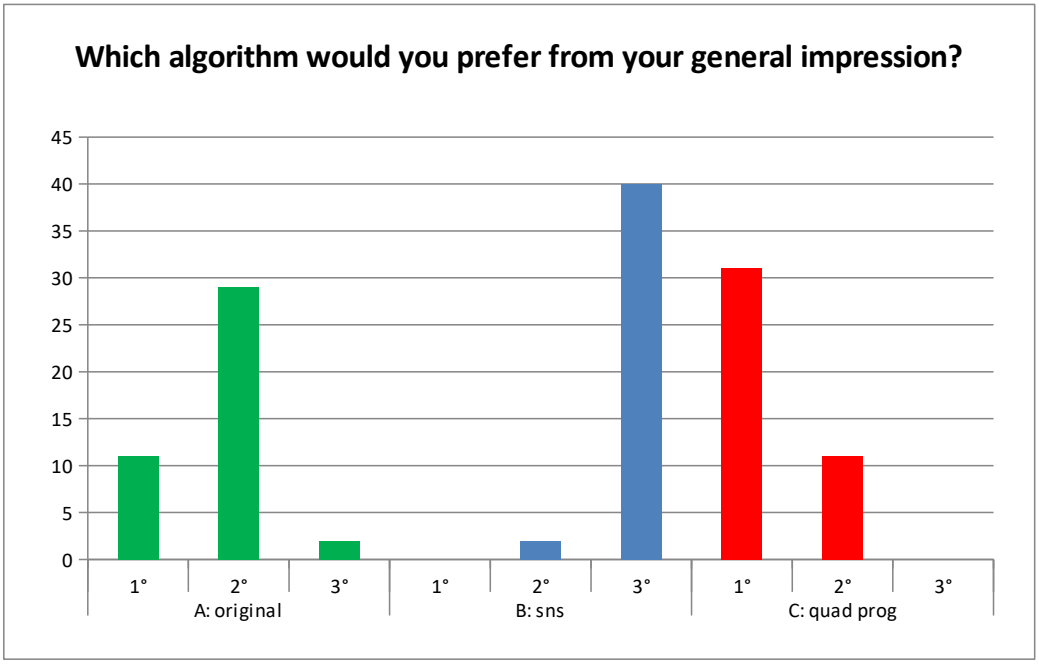


Figure 5.9: Third question histogram

- $H_0 : M_1 = M_2$
- $\chi^2_{critical} = 3,841$

For higher χ^2 values the null hypothesis must be rejected. The test was performed once for each question and each algorithm was compared with both the other ones.

The algorithm B always produced very high $\chi^2 > 25$ values in every comparison, this method received very bad usability scores in every section, the main reason is the already mention impossibility to slide along joint limits and the freeze of the entire robot when one of them is hit. Especially for non expert user having to manually move back away from the limit was not intuitive making this simple task rather tricky to perform. This result also into longer time and unexpected motions or stop from the robot.

The analysis between algorithm A and C leads to different significance results. The computed χ^2 value are listed below with the relative question.

- It was easy and natural to perform the required task:
  $\chi^2 = 8.23$, $p \approx 0,005$
- I'm satisfied by the amount of time required to perform the task:
  $\chi^2 = 0.76$, $p = 0,25$
- I didn't notice any unexpected movements/stop of the robot:
  $\chi^2 = 13.27$, $p \approx 0,005$

For question number two the null hypothesis cannot be rejected and the two distributions can't be considered statistically different. Since the big difference between the two approaches is the presence in C of velocity limits, that yield to a smoother approach to the elbow limit, this result appear reasonable since it won't affect too much the total time required to perform the task.

Question number 1 and 3 had greater value and also for these cases the distribution difference of the scores must be considered statistically significant.

Also in this case the hard stop produced by A yields to some unexpected and non natural motion for the users.

At last if the general ranking is considered the 76% of the subjects preferred the new solution C to the already existing ones, also this last value reinforces the assessed superiority of this approach.

Appendix A

Code

A.1 Velocity Bounds function

```
function [ QdotMax, QdotMin ] = MaxVel_allowed( q ,
    Vmax, Amax, Lmax, Lmin, dt)
%#eml

% The function computes maximum joint velocities for
    the robot configuration "q".

% The function requires:
%- joint position limits (Lmax,Lmin)
%- joint absolute velocity limit (Vmax)
%- joint absolute velocity limit (Amax)

% The formula takes into account all bounds (position,
    velocity, acceleration) at the same time

QdotMax=zeros(length(q),1);
QdotMin=zeros(length(q),1);

```

```
16 % If current position is over the position limit the
       respective velocity is set to 0
17
18 for i=1:length(q)
19
20     if q(i)>=Lmax(i)
21         QdotMax(i)=0;
22     else
23         QdotMax(i) = min( [ (Lmax(i)-q(i))/dt , Vmax ,
               sqrt(2*Amax*(Lmax(i)-q(i))) ] );
24     end
25
26     if q(i)<=Lmin(i)
27         QdotMin(i)=0;
28     else
29         QdotMin(i) = max( [ (Lmin(i)-q(i))/dt , -Vmax ,
               -sqrt(2*Amax*(q(i)-Lmin(i))) ] );
30     end
31
32 end
33
34 end
```

A.2 Inequality Constraints - matrix initialization function

```
1 function [B,b] = getConstraintsMatrices(invJ,QdotMax,
      QdotMin,qdot_d,n,m)
2 %#eml
3
```

```
4  % The function returns the inequality constraints
       matrices for a QP problem s.t. B x <= b
5
6  % We have 14 inequality constraints 7 for max and 7 for
       min velocity
7
8  B=zeros(2*n,m);
9  b=zeros(2*n,1);
10
11 for i=1:length(qdot_d)
12
13     B(i,:) = invJ(i,:);
14     b(i) = QdotMax(i)-qdot_d(i);
15     B(i+n,:) = -invJ(i,:);
16     b(i+n) = -QdotMin(i)+qdot_d(i);
17
18 end
19
20 end
```

A.3 QP solver - Active Set Method

```
1  function [x, k] = qp_solver(W, B, b, x0)
2  %#eml
3
4  % The function solves the QP problem defined by:
5  % - objective function: 1/2 x'Wx
6  % - subject to: B x <= b linear inequality constraints
7  % the resulting x minimizes the objective function
       while respects the constraints.
8
9  % The method used is called "Active set method".
```

```
10 % The active set is made of those constraints that are
       active at the current point xk, so for them B*xk=b
11
12 % At each step we look for the optimal solution by
       moving on the direction that solves the equality
       constraints problem defined by the active set.
13
14 % The starting point x0 is required and must be
       feasible
15 m = size(W,2);
16 x = x0(1:m);
17 k = 0;
18
19 % Active Set initialization (NaN: non active const, 1:
       active)
20 A = NaN(size(B,1),1);
21 A(B*x==b) = 1;
22
23
24 % While no convergence to optimal point
25 while 1
26
27     % Solve the QP to obtain the optimal search
           direction:
28     % 0.5 d'Qd + g'd
29         % under the active set constraints considered
               as equality ones:
30         % A d = 0
31
32
33     % gk = Q xk + q
34     g = x;
```

```
35
36      % KKT equations to solve:
37      % B d = 0;
38      % x + B au = 0;
39      % [I B_t';B_t 0] [d au] = [-g;0]
40
41      mu = zeros(size(B,1),1);
42      d = zeros(m,1);
43
44      % number of active constraints
45      a = size(B(~isnan(A),:),1);
46
47      c = [eye(m) , B(~isnan(A),:)' ; B(~isnan(A),:) ,
            zeros(a)] \ [ -g ; zeros(a,1)];
48
49      d = c(1:m);
50      mu(~isnan(A)) = c(m+1:m+a);
51
52      % Direction d check (case 1: d=0, case 2: d~=0)
53
54      % case 1: d=0
55      if norm(d) < 1e-5 && norm(d) > -1e-5
56
57          % Check lagrangian multipliers:
58
59          % all positive: optimal point
60          if all(mu >= 0)
61              break
62          % some negative: remove from active set the
                minor and solve again
63          % the KKT equations
64          else
```

```
65              tmp_log = mu<0 & ~isnan(A);
66              tmp_ind = find(tmp_log);
67
68              [~, i_mu0] = min(mu(tmp_log));
69
70              A(tmp_ind(i_mu0))=NaN;
71          end
72
73      % case 2: d~=0
74      else
75
76          % Compute the new step ( x(k+1) = x(k) + alfa*d
                )
77
78          % compute the alfa (maximum movement along the
               d direction before
79          % hitting a constraints)
80          tmp_log = isnan(A) & B*d>1e-6;
81          tmp_ind = find(tmp_log);
82
83          [alfa, i_alfa] = min([ 1 ; (b(tmp_log)-B(
               tmp_log,:)*x)./(B(tmp_log,:)*d) ]);
84
85          % update the new step
86          x = x + alfa*d;
87          k = k+1;
88
89          % if alfa lower then 1 with the new step we are
                hitting the
90          % respective constraint so we need to add it to
                the active set
91          if alfa~=1
```

```
92                  A(tmp_ind(i_alfa-1)) = 1;
93              end
94          end
95
96  end
97  end
```

Appendix B

Usability Questionnaire

Age: 20-29☐ 30-39☐ 40+☐

Gender: M☐ F ☐

Dominant hand: Left☐ Right☐

Previous experience with Hand-Guided robot?

Never☐ <5trials☐ <20 trials☐ Regularly☐

	Algorithm:		
	First (1-3 trials)	Second (4-6 trials)	Third (7-9 trials)
It was **easy** and **natural** to perform the required task (trajectory following) use scale 1 (very difficult) – 5 (very easy/natural)			
I'm satisfied by the amount of **time** required to perform the task use scale 1-5 1 (much longer than expected) – 3 (as expected) - 5 (faster than expected)			
I didn't notice any unexpected **movements/stop** of the robot (abrupt movements or waving) use scale 1 (a lot of unexpected movements) – 5 (no unexpected movements)			
Which algorithm would you prefer from your general impression? assign a rank [1-3], where 1 is the favorite			
And why? optional comments on the ranking			
Further impressions/comments: …… …… ……			

Bibliography

[1] Whitney, Daniel E. *"The mathematics of coordinated control of prosthetic arms and manipulators."* ASME Journal of Dynamic Systems, Measurement and Control 20.4 (1972): 303-309.

[2] A. Liegeois, *"Automatic supervisory control of the configuration and behavior of multibody mechanisms."* IEEE Trans. Syst. Man Cybern. 7 (12) (1977) 868–871.

[3] Yoshikawa, Tsuneo, *"Manipulability of robotic mechanisms."* The international journal of Robotics Research 4.2 (1985): 3-9.

[4] Maciejewski, Anthony A., and Charles A. Klein. *"Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments."* The international journal of robotics research 4.3 (1985): 109-117.

[5] Huo, Liguo, and Luc Baron. *"The self-adaptation of weights for joint-limits and singularity avoidances of functionally redundant robotic-task."* Robotics and Computer-Integrated Manufacturing 27.2 (2011): 367-376.

[6] Nakamura, Yoshihiko, Hideo Hanafusa, and Tsuneo Yoshikawa, *"Task-priority based redundancy control of robot manipulators."* The International Journal of Robotics Research 6.2 (1987): 3-15.

[7] Walker, Ian D., and Steven I. Marcus. *"Subtask performance by redundancy resolution for redundant robot manipulators."* IEEE Journal on Robotics and Automation 4.3 (1988): 350-354.

[8] Chiacchio, P., Chiaverini, S., Sciavicco, L., and Siciliano, B. (1991). *"Closed-Loop Inverse Kinematics Schemes for Constrained Redundant Manipulators with Task Space Augmentation and Task Priority Strategy."* The International Journal of Robotics Research, 10, 410–425.

[9] Baron, L. (2000). *"A Joint-Limits Avoidance Strategy for Arc-Welding Robots."* Int. Conf. on Integrated Design and Manufacturing in Mech. Eng., (January).

[10] Baillieul, John. *"Avoiding obstacles and resolving kinematic redundancy."* Robotics and Automation. Proceedings. 1986 IEEE International Conference on. Vol. 3. IEEE, 1986.

[11] Chan, Tan Fung, and Rajiv V. Dubey. *"A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators."* IEEE Transactions on Robotics and Automation 11.2 (1995): 286-292.

[12] Xiang, Ji, Congwei Zhong, and Wei Wei. *"General-weighted least-norm control for redundant manipulators."* IEEE Transactions on Robotics 26.4 (2010): 660-669.

[13] D. Luenberger, *"Linear and Nonlinear Programming."* Reading, MA, USA: Addison-Wesley, 1984.

[14] Lawson, Charles L., and Richard J. Hanson. *"Solving least squares problems."* Society for Industrial and Applied Mathematics, 1995.

[15] Hollerbach, J. O. H. N. M., and Ki Suh, *"Redundancy resolution of manipulators through torque optimization."* IEEE Journal on Robotics and Automation 3.4 (1987): 308-316.

[16] K. A. O'Neil, *"Divergence of linear acceleration-based redundancy resolution schemes."* IEEE Trans. Robot. Automat., vol. 18, pp. 625–631, Aug. 2002.

[17] J. Park, W.-K. Chung, and Y. Youm, *"Characterization of instability of dynamic control for kinematically redundant manipulators."* IEEE Int. Conf. Robotics and Automation, vol. 3, 2002, pp. 2400–2405.

[18] Zhang, Yunong, Shuzhi Sam Ge, and Tong Heng Lee, *"A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators."* IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34.5 (2004): 2126-2132.

[19] Zhang, Yunong, and Shugen Ma, *"Minimum-energy redundancy resolution of robot manipulators unified by quadratic programming and its online solution."* Mechatronics and Automation, 2007. ICMA 2007. International Conference on. IEEE, 2007.

[20] F. Cheng, T. Chen, and Y. Sun, *"Resolving manipulator redundancy under inequality constraints."* IEEE Trans. Robot. Autom., vol. 10, no. 1, pp. 65–71, Feb. 1994.

[21] F. Cheng, R. Sheu, and T. Chen, *"The improved compact QP method for resolving manipulator redundancy."* IEEE Trans. Syst.,Man, Cybern., vol. 25, no. 11, pp. 1521–1530, Nov. 1995.

[22] Chen, Weihai, I-Ming Chen, and Tianmiao Wang. *"Kinematic fault tolerant control for redundant robot based on joint velocities redistribution."* Proc. of Int. Conf. Advanced Robotics, ICAR, Tokyo, Japan. 1999.

[23] Omrčen, Damir, Leon Žlajpah, and Bojan Nemec. *"Compensation of velocity and/or acceleration joint saturation applied to redundant manipulator."* Robotics and Autonomous Systems 55.4 (2007): 337-344.

[24] Flacco, Fabrizio, Alessandro De Luca, and Oussama Khatib. *"Motion control of redundant robots under joint constraints: Saturation in the null space."* Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, 2012.

[25] Flacco, Fabrizio, 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 (2015): 637-654.

[26] Raunhardt, Daniel, and Ronan Boulic. *"Progressive clamping."* Robotics and Automation, 2007 IEEE International Conference on. IEEE, 2007.

[27] Merloz, Philippe, et al. *Fluoroscopy-based navigation system in spine surgery.* Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine 221.7 (2007): 813-820.

[28] Merloz, Philippe, et al. *"Computer assisted spine surgery."* Clinical orthopaedics and related research 337 (1997): 86-96.

[29] Leksell, L. A. R. S., Leksell, D., & Schwebel, J. O. H. N. *"Stereotaxis and nuclear magnetic resonance."* Journal of Neurology, Neurosurgery & Psychiatry, 14-18 (1985).

[30] Cohen DS, Lustarten JH, Miller Eet al. *"Effects of coregistration of MR to CT images on MR stereotactic accuracy."* J Neurosurg . 1995, 82, 5:772–779.

[31] Wiesner L, Kothe R, Schulitz KP, Ruther W. *"Clinical evaluation and computed tomography scan analysis of screw tracts after percutaneous insertion of pedicle screws in the lumbar spine."* Spine . 2000, 25, 5:615–621.

[32] Roser F, Tatagiba M, Maier G *"Spinal robotics: current applications and future perspectives."* Neurosurgery . 2013, 72, suppl 1:12–18.

[33] Ringel F, Stuer C, Reinke Aet al. *"Accuracy of robot-assisted placement of lumbar and sacral pedicle screws: a prospective randomized comparison to conventional freehand screw implantation."* Spine . 2012, 37, 8:E496–E501.

Picture Sources

[1] https://webcontent.temed.com/spine/disorders/symptoms/spinal-deformity

[2] http://www.newhealthadvisor.com/Spinal-Fusion-Recovery.html

[3] http://www.ece.ualberta.ca/ tbs/pmwiki/index.php?n=Research.TeleoperationSystemsForRobot-assistedMinimallyInvasiveSurgery

[4] https://www.healthcare.siemens.com/surgical-c-arms-and-navigation

[5] https://www.medica.de/cgi-bin/md_medica/lib/pub/tt.cgi/KUKA_Leichtbauroboter_LBR_Med_zur_Integration_in_ein_Medizinprodukt_zertifiziert.html?oid=85735&lang=1&ticket=g_u_e_s_t

[6] https://www.brainlab.com/surgery-products/overview-platform-products/intraoperative-ct/

[7] https://www.youtube.com/watch?v=0UOHVN-VeLM

[8] https://medical.electronicspecifier.com/bioengineering/3d-printing-helps-fix-6-year-old-boy-s-spine

[9] http://www.urologiagallo.it/il-robot-da-vinci-in-urologia-come-funziona-quando-dove-e-perche-si-usa-sempre-di-piu/