

Multi-Prioritized Torque-Based Control Framework with Cartesian and Joints Constraints and its Medical Robotics Application

Giampiero Esposito¹, Vincenzo Schettino², Vito Daniele Perfetta¹, Domenico Tuccillo¹, Paolino De Risi¹, Bruno Siciliano¹, and Fanny Ficuciello¹

¹ University of Naples “Federico II”, 80125 Napoli NA, Italy,
giampiero.esposito123@gmail.com, v.perfetta@studenti.unina.it,
domen.tuccillo@studenti.unina.it, paolino.derisi@unina.it,
bruno.siciliano@unina.it, fanny.ficuciello@unina.it,

² Stryker Corporation, 49002 Kalamazoo MI, US,
v.schettino@hotmail.it

Abstract. Robot-assisted surgical systems offer significant advantages in bone resection procedures, increasing accuracy and minimizing risk to surrounding anatomical structures. This work presents an innovative control framework integrating robot hand guidance and high-speed burr navigation to improve precision and efficiency in spinal surgery. A task space control algorithm is implemented for accurate positioning of the tool center point (TCP) within the surgical navigation frame. Safe resection areas are encoded as dynamic workspace constraints for the robot, and employed for the first time, to our knowledge, in the context of impedance control. Two control strategies are employed: one neglecting external torques, the other compensating for them. Our approach is based on exploiting the redundancies of the robot, ensuring constrained motion within the safe surgical workspace. Experimental validation, implemented on a KUKA LBR Med robot, shows improved accuracy, reduced surgical procedure time, and lower cognitive load for surgeons.

Keywords: Impedance Control, Saturation in Cartesian Space, Saturation in Joint Space, Null-Space Projection, Robot-assisted surgery.

1 Introduction

The spinal column provides structural support for the body while protecting the spinal canal. Treating conditions such as spinal tumors, herniated discs, and spinal canal stenosis often requires surgical intervention. This work aims to define a control

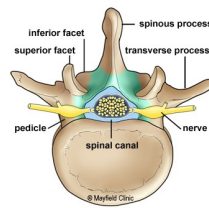


Fig. 1a : Vertebra Superior View



Fig. 1b : Laminotomy

framework to assist experienced surgeons in performing robotics-aided bone resection procedures, such as laminotomy. This is a minimally invasive spinal surgery that involves the partial removal of the lamina of a vertebra (see Fig. 1a). During the surgery, a small incision is made over the affected spinal segment, and the lamina is exposed. A portion of the lamina is then removed (see Fig. 1b) to decompress the nerves [9]–[6]. In this context, we implement a control procedure that ensures the surgeon can handguide the robot end-effector (EE) while operating within predefined spatial boundaries — an essential feature in the context of safety critical surgical areas as in laminotomy — enhancing surgery precision. We define a hierarchical control framework based on the Saturation in the Null-Space (SNS) method [2], tested in the context of robotic-assisted spinal surgery. In the context at hand, several tasks have to be executed in the in a precise priority order [4]: **Joint limits avoidance**, **Elbow constraints** (for optimal performance, it is constrained within a predefined box), **TCP constraints** (to avoid damages to tissues, it is constrained within a safe site), **Cartesian task execution** (hand-guiding during surgery), **Stability enhancement** (minimizing oscillations).

2 Related Works

Prioritized task management is a well-documented challenge in the literature, with Null-Space projection techniques widely recognized for exploiting the inherent functional redundancy of robotic systems to address this issue [1]. Additionally, methods such as Saturation in Cartesian Space (SCS) and its joint space variant (SJS) [10] are commonly employed to prevent robots from exceeding specified physical or virtual boundaries, thereby ensuring safe interactions with both the environment and human operators, although their application has been predominantly in motion and kinematic control. Handling of task conflicts in recursive Null-Space projections is exploited in [11], [8]. In this study, we integrate the SCS and SJS algorithms with Null-Space projection techniques to concurrently manage multiple safety-critical tasks within a hand-guiding framework for surgical applications, representing, to our knowledge, the first application of such algorithms together with an impedance control framework.

3 Technical Approach

Redundant manipulators can achieve multiple prioritized tasks (we denote a Cartesian task as $T_{c,r}$ and a task in joint space as $T_{j,r}$ with $r \in \{1, \dots, R\}$, defining the priority level) via Null-Space projections [1]. We denote the torque resolving a general stack of tasks as:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{c/j,1} + \mathbf{N}_{c/j,1} \{ \boldsymbol{\tau}_{c/j,2} + \dots + \mathbf{N}_{c/j,r-1} \{ \boldsymbol{\tau}_{c/j,r} + \dots + \mathbf{N}_{c/j,R-1} \{ \boldsymbol{\tau}_{c/j,R} \} \dots \} \} + \mathbf{g}(\mathbf{q}), \quad (1)$$

where $\mathbf{N}$ and $\boldsymbol{\tau}$ are the Null-Space projector and the torque associated to a Cartesian or joint-space task $T_{c/j,r}$ with priority level r , $\mathbf{q} \in \mathbb{R}^n$ denotes the

robot's joint configuration vector and $\mathbf{g}(\mathbf{q})$ is the gravity compensation force. This formulation ensures that lower-priority tasks do not interfere with higher-priority ones [2]. In our framework, Cartesian tasks execution is ensured by Saturation in Cartesian Space (SCS) algorithm, while for joint tasks Saturation in Joint Space (SJS) algorithm [10] is employed. Dynamic constraints are introduced and employed in the context of impedance control.

3.1 Impedance Control

In the following we assume that the lowest priority task $T_{c,R}$ in the stack is a Cartesian impedance task, allowing manual guidance of the robot. Consider the robot dynamics in the joint space [7], in absence of gravity effects:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \boldsymbol{\tau} - \mathbf{J}^T(\mathbf{q})\mathbf{h}_e, \quad (2)$$

where $\mathbf{J} \in \mathbb{R}^{m \times n}$ is the geometric Jacobian, $\mathbf{M}(\mathbf{q})$ the inertia matrix, $\mathbf{h}_e$ the external forces, $\boldsymbol{\mu}(\mathbf{q}, \dot{\mathbf{q}})$ denotes the centrifugal and Coriolis effects matrix. The control input is defined as:

$$\boldsymbol{\tau}_{c,R} = \mathbf{M}(\mathbf{J}_A^{-1}\boldsymbol{\Lambda}_d^{-1}(\boldsymbol{\Lambda}_d\ddot{\mathbf{x}}_d + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} - \boldsymbol{\Lambda}_d\dot{\mathbf{J}}_A\dot{\mathbf{q}} - \mathbf{h}_A)) + \boldsymbol{\mu}\dot{\mathbf{q}} + \mathbf{J}^T\mathbf{h}_e, \quad (3)$$

where $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$, with $\mathbf{x}_d$ and $\mathbf{x} \in \mathbb{R}^m$ denoting the desired and actual Cartesian pose, respectively, $\mathbf{h}_A$ is the external force expressed with respect to the analytical Jacobian $\mathbf{J}_A$, $\boldsymbol{\Lambda}_d$ is the desired inertia in the Cartesian space, $\mathbf{K}_v$ and $\mathbf{K}_p$ are the two positive definite damping and stiffness matrices. For the sake of brevity, the corresponding dependencies from $\mathbf{q}$ and $\dot{\mathbf{q}}$ are omitted in (3). The resulting mass-spring-damper closed loop system is:

$$\boldsymbol{\Lambda}_d\ddot{\mathbf{e}} + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = \mathbf{h}_A, \quad (4)$$

representing a fully decoupled system if measurements of the external forces are employed for compensation in (3), otherwise residual coupling effect is present.

3.2 Saturation in Cartesian Space Algorithm

The Saturation in Cartesian Space (SCS) algorithm enforces constraints on the motion of a point of interest $\mathbf{c}_r$ for the generic Cartesian task $T_{c,r}$ (e.g. the EE, or the robot's elbow tasks), ensuring it remains confined within a predefined virtual workspace. To achieve this, a set of static Cartesian constraints can be defined for the task $T_{c,r}$ such that:

$$\mathbf{c}_{r,min} \leq \mathbf{c}_r \leq \mathbf{c}_{r,max}, \quad \mathbf{v}_{r,min} \leq \dot{\mathbf{c}}_r \leq \mathbf{v}_{r,max}, \quad \mathbf{a}_{r,min} \leq \ddot{\mathbf{c}}_r \leq \mathbf{a}_{r,max}. \quad (5)$$

Assuming that the stack of tasks is only composed of Cartesian tasks, we denote the output of the task hierarchy computation at the priority level r as the torque $\boldsymbol{\tau}_{r,stack}$. However, assuming that $\boldsymbol{\tau}_{r,stack}$ produces a constant acceleration of the point associated to the higher priority task $T_{c,r-1}$, i.e. $\ddot{\mathbf{c}}_{r-1,h} = \ddot{\mathbf{c}}_{r-1}(t_h) =$

$\ddot{\mathbf{c}}_{r-1}(h\delta t)$ over a sample time δt , excessive deceleration may occur when $\mathbf{c}_{r-1}$ reaches a positional limit while moving at maximum velocity, with possible discontinuities in the velocity profile. To mitigate this, dynamic velocity limits are introduced in this work, as:

$$\begin{aligned}\dot{\mathbf{c}}_{r-1,max}(t_h) &= \min(-\dot{\mathbf{d}}_{r-1,max}(t_h) - \mathbf{K}\mathbf{d}_{r-1,max}(t_h), \mathbf{v}_{r-1,max}) \\ \dot{\mathbf{c}}_{r-1,min}(t_h) &= \max(-\dot{\mathbf{d}}_{r-1,min}(t_h) - \mathbf{K}\mathbf{d}_{r-1,min}(t_h), \mathbf{v}_{r-1,min}) ,\end{aligned}\quad (6)$$

where $\mathbf{d}_{r-1,max}(t_h) = \mathbf{c}_{r-1}(t_h) - \mathbf{c}_{r-1,max}$, $\mathbf{d}_{r-1,min}(t_h) = \mathbf{c}_{r-1}(t_h) - \mathbf{c}_{r-1,min}$ and $\mathbf{K}$ is a gain matrix weighting the positional constraint over the velocity one, in a “PD” control fashion. Consequently, (6) yields dynamic acceleration constraints expressed as:

$$\begin{aligned}\ddot{\mathbf{c}}_{r-1,min}(t_h) &= \max\left(\frac{2(\mathbf{d}_{r-1,min} - \dot{\mathbf{c}}_{r-1}\delta t)}{\delta t^2}, \frac{\dot{\mathbf{c}}_{r-1,min} - \dot{\mathbf{c}}_{r-1}}{\delta t}, \mathbf{a}_{r-1,min}\right) \\ \ddot{\mathbf{c}}_{r-1,max}(t_h) &= \min\left(\frac{2(\mathbf{d}_{r-1,max} - \dot{\mathbf{c}}_{r-1}\delta t)}{\delta t^2}, \frac{\dot{\mathbf{c}}_{r-1,max} - \dot{\mathbf{c}}_{r-1}}{\delta t}, \mathbf{a}_{r-1,max}\right) .\end{aligned}\quad (7)$$

In this manner, the acceleration limits are dynamically adjusted based on the proximity of the point $\mathbf{c}_{r-1}$ to its positional, velocity, or acceleration constraints. To check if these new limits are respected, we compute the Cartesian acceleration $\ddot{\mathbf{c}}_{r-1}$ as if $\boldsymbol{\tau}_{r,stack}$ would be the input to the system. First, we obtain $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ from the forward dynamics with $\boldsymbol{\tau}_{r,stack}$ as input. Then, from forward kinematics, we get $\ddot{\mathbf{c}}_{r-1} = \mathbf{J}_{r-1}\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{r-1}\dot{\mathbf{q}}$, where $\mathbf{J}_{r-1}$ is the Jacobian associated to the task $T_{c,r-1}$. If all the limits are respected, then $\boldsymbol{\tau}_{r,stack}$, becomes $\boldsymbol{\tau}_{r-1,stack}$. Conversely, if any Cartesian acceleration in $\ddot{\mathbf{c}}_{r-1}$ exceeds its constraints, a limiting torque $\boldsymbol{\tau}_{r,lim}$ is computed to saturate the acceleration and respect the limits:

$$\boldsymbol{\tau}_{r,lim} = \mathbf{J}_{r-1,lim}^T(\boldsymbol{\Lambda}(\mathbf{q})\ddot{\mathbf{c}}_{r-1,lim} + \boldsymbol{\mu}_c(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}) , \quad (8)$$

where $\boldsymbol{\Lambda}$ and $\boldsymbol{\mu}_c$ are the inertia and Coriolis matrices in the Cartesian space, respectively. The terms $\ddot{\mathbf{c}}_{r-1,lim}$ and $\mathbf{J}_{r-1,lim}$ are constructed as follows:

$$\ddot{\mathbf{c}}_{r-1,lim}(i) = \begin{cases} \ddot{\mathbf{c}}_{r-1,max}(i), & \text{if } \ddot{\mathbf{c}}_{r-1}(i) > \ddot{\mathbf{c}}_{r-1,max}(i) \\ \ddot{\mathbf{c}}_{r-1,min}(i), & \text{if } \ddot{\mathbf{c}}_{r-1}(i) < \ddot{\mathbf{c}}_{r-1,min}(i) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

$$\mathbf{J}_{r-1,lim}(i, :) = \begin{cases} \mathbf{J}_{r-1}(i, :), & \text{if } |\ddot{\mathbf{c}}_{r-1}(i)| > |\ddot{\mathbf{c}}_{r-1,lim}(i)| \\ \mathbf{0}, & \text{otherwise} \end{cases} . \quad (10)$$

Then, the control input for the task hierarchy at level $r - 1$ is computed as:

$$\boldsymbol{\tau}_{r-1,stack} = \boldsymbol{\tau}_{r,lim} + \mathbf{N}_{r-1,lim}\{\boldsymbol{\tau}_{r,stack}\} , \quad (11)$$

where $\mathbf{N}_{r-1,lim} = \mathbf{I} - \mathbf{J}_{r-1,lim}^T \bar{\mathbf{J}}_{r-1,lim}^T$ is the Null-Space projector associated to the Jacobian $\mathbf{J}_{r-1,lim}$ with $\bar{\mathbf{J}}_{r-1,lim}$ representing the pseudoinverse of $\mathbf{J}_{r-1,lim}$.

The process is repeated for every higher priority task until the final torque $\tau_{1,stack}$ is generated. The algorithm pseudocode is presented in Alg.1. Moreover, considering the impedance task $T_{c,R}$, if the external forces $\mathbf{h}_e$ are compensated in (3), the acceleration limits in (7) are adjusted to incorporate the effect of the external forces as follows:

$$\ddot{\mathbf{c}}_{r-1,min} = \ddot{\mathbf{c}}_{r-1,min} - \Lambda_d \Lambda^{-1}(\mathbf{q}) \mathbf{h}_A, \quad \ddot{\mathbf{c}}_{r-1,max} = \ddot{\mathbf{c}}_{r-1,max} - \Lambda_d \Lambda^{-1}(\mathbf{q}) \mathbf{h}_A. \quad (12)$$

Algorithm 1 Saturation in Cartesian Space (SCS) Algorithm

Require: Cartesian tasks $T_{c,r}$, bounds $\mathbf{c}_{r,min}$, $\mathbf{c}_{r,max}$, $\mathbf{v}_{r,min}$, $\mathbf{v}_{r,max}$, $\mathbf{a}_{r,min}$, $\mathbf{a}_{r,max}$
Ensure: Torque command $\tau_{1,stack}$

```

for  $r = R$  to 2 do ▷ Start from lowest-priority task
  Compute  $\tau_{r,stack}$  for task  $T_{c,r}$  using task hierarchy
  Predict  $\dot{\mathbf{q}}, \ddot{\mathbf{q}}$  via forward dynamics with  $\tau_{r,stack}$ 
  Compute Cartesian acceleration:  $\ddot{\mathbf{c}}_{r-1} = \mathbf{J}_{r-1}^T \ddot{\mathbf{q}} + \dot{\mathbf{J}}_{r-1}^T \dot{\mathbf{q}}$ 
  Compute dynamic velocity limits for task  $T_{c,r}$  as in (6)
  Compute dynamic acceleration limits for task  $T_{c,r}$  as in (7)
  if any  $\ddot{\mathbf{c}}_{r-1}$  violates limits then
    for each dimension  $i$  do
      if  $\ddot{c}_{r-1}(i) > \ddot{c}_{r-1,max}(i)$  then
         $\ddot{c}_{r-1,lim}(i, :) \leftarrow \ddot{c}_{r-1,max}(i)$ 
         $\mathbf{J}_{r-1,lim}(i, :) \leftarrow \mathbf{J}_{r-1}(i, :)$ 
      else if  $\ddot{c}_{r-1}(i) < \ddot{c}_{r-1,min}(i)$  then
         $\ddot{c}_{r-1,lim}(i, :) \leftarrow \ddot{c}_{r-1,min}(i)$ 
         $\mathbf{J}_{r-1,lim}(i, :) \leftarrow \mathbf{J}_{r-1}(i, :)$ 
      else
         $\ddot{c}_{r-1,lim}(i, :) \leftarrow 0$ 
         $\mathbf{J}_{r-1,lim}(i, :) \leftarrow \mathbf{0}$ 
    end if
  end for
  Compute saturated torque:  $\tau_{r,lim} = \mathbf{J}_{r-1,lim}^T (\Lambda \ddot{\mathbf{c}}_{r-1,lim} + \mu_c \dot{\mathbf{q}})$ 
  Compute Null-Space projector:  $\mathbf{N}_{r-1,lim} = \mathbf{I} - \mathbf{J}_{r-1,lim}^T \mathbf{J}_{r-1,lim}^T$ 
  Update torque:  $\tau_{r-1,stack} = \tau_{r,lim} + \mathbf{N}_{r-1,lim} \{\tau_{r,stack}\}$ 
else
   $\tau_{r-1,stack} = \tau_{r,stack}$ 
end if
end for
return  $\tau_{1,stack}$ 

```

Algorithm 2 Saturation in Joint Space (SJS) Algorithm

Require: Joint Tasks $T_{j,r}$, bounds $\mathbf{q}_{r,min}$, $\mathbf{q}_{r,max}$, $\mathbf{v}_{r,min}$, $\mathbf{v}_{r,max}$, $\mathbf{a}_{r,min}$, $\mathbf{a}_{r,max}$
Ensure: Torque command $\tau_{1,stack}$

```

for  $r = R$  to 2 do ▷ Start from lowest-priority task
  Compute  $\tau_{r,stack}$  for task  $T_{j,r}$  using task hierarchy
  Predict  $\dot{\mathbf{q}}$  via forward dynamics with  $\tau_{r,stack}$ 
  Compute dynamic acceleration limits for task  $T_{j,r}$  as in (14)
  if any  $\dot{\mathbf{q}}$  violates limits then
    for each joint  $i$  do
      if  $\dot{q}(i) > \dot{q}_{r-1,max}(i)$  then
         $\dot{q}_{r-1,lim}(i) \leftarrow \dot{q}_{r-1,max}(i)$ 
         $\mathbf{J}_{r-1,lim}(i, :) \leftarrow \mathbf{1}$ 
      else if  $\dot{q}(i) < \dot{q}_{r-1,min}(i)$  then
         $\dot{q}_{r-1,lim}(i) \leftarrow \dot{q}_{r-1,min}(i)$ 
         $\mathbf{J}_{r-1,lim}(i, :) \leftarrow \mathbf{1}$ 
      else
         $\dot{q}_{r-1,lim}(i) \leftarrow 0$ 
         $\mathbf{J}_{r-1,lim}(i, :) \leftarrow \mathbf{0}$ 
    end if
  end for
  Compute saturated torque:  $\tau_{r,lim} = \mathbf{J}_{r,lim}^T (M \dot{\mathbf{q}}_{r-1,lim} + \mu \dot{\mathbf{q}})$ 
  Compute Null-Space projector:  $\mathbf{N}_{r-1,lim} = \mathbf{I} - \mathbf{J}_{r-1,lim}^T \mathbf{J}_{r-1,lim}^T$ 
  Update torque:  $\tau_{r-1,stack} = \tau_{r,lim} + \mathbf{N}_{r-1,lim} \{\tau_{r,stack}\}$ 
else
   $\tau_{r-1,stack} = \tau_{r,stack}$ 
end if
end for
return  $\tau_{1,stack}$ 

```

3.3 Saturation in Joint Space Algorithm

Assuming that in the stack is present a joint space task with a generic priority level r , T_{jr} , a similar reasoning can be applied considering joint constraints as:

$$\mathbf{q}_{min} \leq \mathbf{q} \leq \mathbf{q}_{max}, \quad \mathbf{v}_{min} \leq \dot{\mathbf{q}} \leq \mathbf{v}_{max}, \quad \mathbf{a}_{min} \leq \ddot{\mathbf{q}} \leq \mathbf{a}_{max}. \quad (13)$$

By following a procedure similar to (5), we get:

$$\begin{aligned} \ddot{\mathbf{q}}_{max} &= \min \left(\frac{\mathbf{q}_{max} - \mathbf{q} - \dot{\mathbf{q}}\delta t}{2\delta t^2}, \frac{\dot{\mathbf{q}}_{max} - \dot{\mathbf{q}}}{\delta t}, \mathbf{a}_{max} \right) \\ \ddot{\mathbf{q}}_{min} &= \max \left(\frac{\mathbf{q}_{min} - \mathbf{q} - \dot{\mathbf{q}}\delta t}{2\delta t^2}, \frac{\dot{\mathbf{q}}_{min} - \dot{\mathbf{q}}}{\delta t}, \mathbf{a}_{min} \right). \end{aligned} \quad (14)$$

To incorporate these time-varying constraints into the task hierarchy, similar steps to the SCS algorithm are applied, with two differences: the forward kinematics for the constraints is no longer required, and the Jacobian $\mathbf{J}_{r,lim}$ of the SCS algorithm becomes a diagonal square activation matrix, that, for parallelism, we will keep on denoting as $\mathbf{J}_{r,lim}$. It is computed in the following way:

$$\mathbf{J}_{r,lim}(i, i) = \begin{cases} 1, & \text{if } |\ddot{q}(i)| > |\ddot{q}_{lim}(i)| \\ 0, & \text{otherwise} \end{cases}. \quad (15)$$

The algorithm pseudocode is presented in Alg.2.

Table 1: Task hierarchy with associated controllers and their priorities (*highest=1*)

Priority	Controller / Algorithm	Associated Task
1	SJS Algorithm	Joint Limits Task
2	SCS-Elbow Algorithm	Elbow Task
3	SCS-EE Algorithm	TCP Task
4	Impedance Control	Handguiding Task
5	Oscillations Suppression	Joint Damping Task

4 Experimental Validation

The proposed algorithms are tested in the MATLAB environment and implemented on a KUKA LBR MED robot equipped with an high-speed rotary cutting tool to simulate a laminotomy on a lumbar spine phantom model. The implemented stack of tasks, shown in Table 1 and also outlined in Sect.1, generates the following final torque command:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{jl} + \mathbf{N}_{jl}\{\boldsymbol{\tau}_{el} + \mathbf{N}_{el}\{\boldsymbol{\tau}_{ee} + \mathbf{N}_{ee}\{\boldsymbol{\tau}_{mg} + \mathbf{N}_{mg}\{\boldsymbol{\tau}_{jd}\}\}\}\}, \quad (16)$$

where the subscripts jl , el , ee , mg and jd denote joint limits, elbow limits, EE limits, manual guidance (impedance), and joint damping tasks, respectively. As the task T_{mg} has no constraint, $\boldsymbol{\tau}_{jd}$ is directly projected in the Null-Space $\mathbf{N}_{mg}$ and computed as $\boldsymbol{\tau}_{jd} = -\mathbf{D}\mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$, where $\mathbf{D}$ is an additive damping gain matrix.

4.1 Simulation Results

This Section presents the results of a simulated bone resection implemented on a simulated KUKA LBR Med robot. The same task hierarchy as in Table 1 is defined. In the impedance control loop, the desired inertia matrix $\mathbf{A}_d$ is set to the identity matrix $\mathbf{I}$. The simulated scenario, illustrated in Fig. 2, considers the robot's EE initially positioned at the center of a circle with a radius of 0.20 m. The EE first moves along the x -axis toward the circumference, then follows a circular trajectory in the xy -plane while maintaining its initial orientation. Bounding limits of ± 0.15 m along the x and y axes are imposed relative to the EE's initial pose. These limits define a rectangular workspace enclosed within the circular path, preventing exact trajectory tracking. The elbow joint is constrained within limits of ± 0.1 m. No external forces are applied in this scenario, effectively reducing the manual guidance task to a trajectory tracking under impedance control. Fig. 3a shows

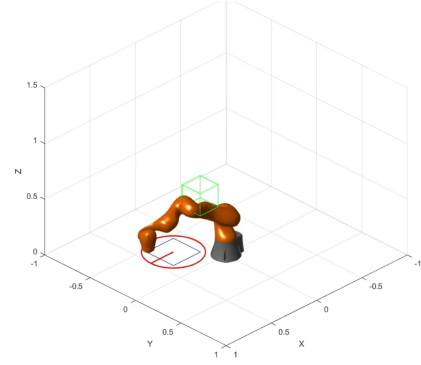


Fig. 2: Bone Resection Algorithm implementation in MATLAB: the circular trajectory to track (*red*), the EE bounding box (*blue*), the elbow bounding box (*green*)

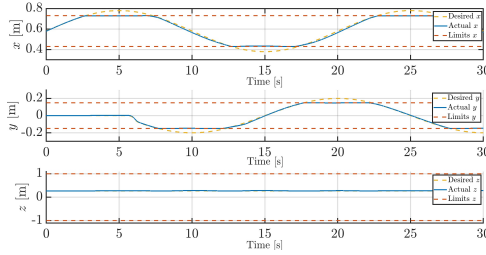


Fig. 3a : Comparison of Actual and Desired Robot EE Position with Limits: in the desired trajectory (*yellow*), actual EE position (*blue*), the maximum and minimum allowable limits (*red*)

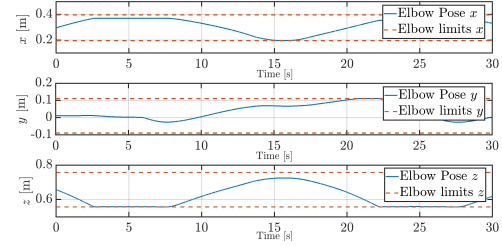


Fig. 3b : Comparison of Actual Elbow Position with Max and Min Limits: the actual elbow trajectory (*blue*), the upper and lower bounds (*red*)

the comparison between the desired and actual EE position over time. The actual trajectory follows the desired path closely, demonstrating effective tracking, until the predefined spatial limits are reached. Upon reaching these limits, the accelerations along the x and y axes saturate, as a result of the Saturation in Cartesian Space (SCS) algorithm over the nominal Cartesian task. Fig. 3b illustrates the evolution of the elbow joint position relative to its predefined limits, showing that it remains within these bounds throughout the experiment.

4.2 Real Hardware Tests

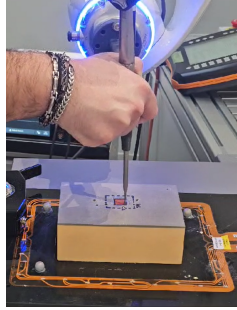


Fig. 4: Experimental setup

In the experiments, a sawbone block [5] is employed, with a cutting area designed to replicate typical laminotomy dimensions. The cutting region is defined with a length and width of 1 cm and a depth of 0.3 cm, serving as the constraints for the EE task T_{ee} . For the elbow task T_{el} , a cubic bounding box with an edge length of 10 cm is defined, constraining the elbow joint's initial pose. The experimental setup is shown in Fig. 4. The proposed algorithm is tested under two conditions: one that neglects external forces $\mathbf{h}_A$ in the impedance control formulation, and another that incorporates them.

No External Torque Compensation Without external torque compensation, the TCP remains within the Cartesian boundaries even during rapid movements. However, if a sufficiently high external force is applied via hand-guiding, the robot may exceed these limits. This behavior is due to the coupled dynamics in equation (4). As a result, the constraint acts as a soft stop, and an additional hard limit is defined by the K gain in (6). Within the region between these two limits, the TCP behavior can be approximated as that of a virtual spring. By adjusting the stiffness of this virtual spring, the size of the hard limit region can be increased or decreased. However, the two limits cannot overlap, as doing so may lead to system instability. While the possibility of exceeding the limits remains inherent in the algorithm, this characteristic is not necessarily undesirable. Expert surgeons may prefer the ability to approach critical areas with flexibility while remaining within safe bounds, relying on haptic feedback when limits are surpassed. The evolution of the EE position over time is shown in Fig. 5a.

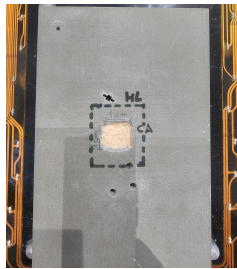


Fig. 6: Results of a simulated laminotomy

External Torque Compensation In this case, the hand-guided system consistently remains within its predefined hard limits, even under the application of high external forces. As shown in Fig. 5b, the deviation of the TCP from these limits remains within 4 mm, which is due to sensor noise and control update rate of 5 ms. Moreover, the experimental results indicate that the cutting process for a single-level laminotomy using this system takes approximately 2 minutes, compared to an average of 10 minutes for traditional manual laminotomy. The results of the experimental bone resection procedure on the sawbone block are shown in Fig. 6.

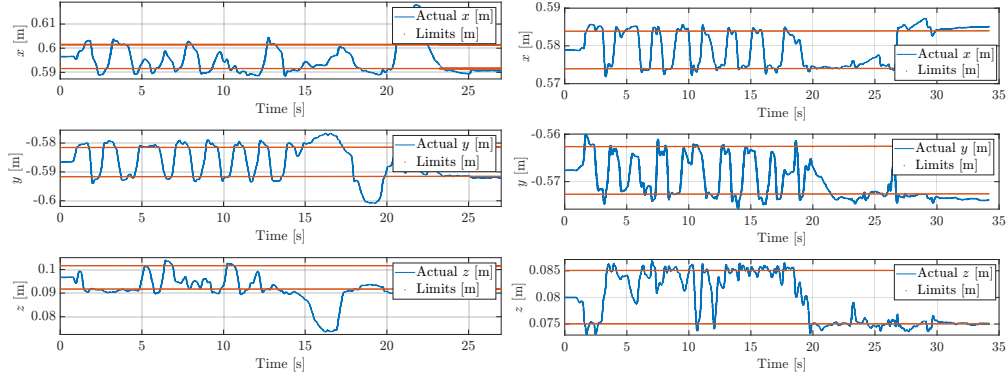


Fig. 5a : Experiment with no compensation of external forces: EE position over time (*blue*) with maximum and minimum limits (*red*)

Fig. 5b : Experiment with compensated external forces: EE position over time (*blue*) with maximum and minimum limits (*red*)

5 Main Experimental Insights

Comparing the presented experiments—one with external force compensation and the other without—yields important insights. Both strategies offer distinct advantages and can be selectively deployed depending on the surgeon’s experience and the specific surgical scenario. For experienced surgeons, the increased freedom of motion provided by the strategy without external force compensation could represent a significant advancement in enhancing surgeon–robot interaction. In contrast, less experienced surgeons may prefer to adopt the more conservative strategy, including external force compensation, as it enforces strict adherence to safety limits. Over time, they could gradually transition to the strategy without external force compensation during training sessions, thereby gaining confidence and expanding their operational capabilities.

6 Conclusion

This work introduced a novel robot-assisted framework for spinal surgery bone resection, combining high-speed burr navigation with robotic guidance. Validation through simulations and experiments demonstrated accurate and smooth tool motion along planned cutting paths. The proposed impedance control framework—leveraging SCS and SJS algorithms with dynamic acceleration constraints—effectively ensured tool confinement within safe boundaries, safeguarding patient tissue. Future research will advance in two directions: control and clinical. On the control side, efforts will focus on developing alternative (variable) impedance control strategies for model decoupling without external force measurements as in [12] and possibly based on machine learning as in [13], task conflict resolution for vibration mitigation, RGB camera integration, and autonomous cutting strategies for enhanced precision. Clinically, the system will

undergo evaluation with surgeons in testing scenarios, collecting feedback to refine control strategies.

7 Acknowledgements

The research leading to these results has been partially supported by the BRIEF “Biorobotics Research and Innovation Engineering Facilities” project (Project identification code IR0000036) funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 3.1 of Italian Ministry of University and Research funded by the European Union – NextGenerationEU, partially by European Union - ERC-2023-SyG Endotheranostics (n. 101118626) and partially by the TI-RED “Towards Intelligent Robotic Endoscopic Dissection” project, PRIN 2022- PNRR, identification code Prot. 20229N5YW8. This research was also partially supported by a Master Thesis Project conducted at Stryker Corporation.

References

1. Chiaverini, S.: Redundant Robots. Springer, Encyclopedia of Systems and Control (2019)
2. Flacco, F., et al.: Prioritized Multi-Task Motion Control of Redundant Robots under Hard Joint Constraints. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2012)
3. Ghogawala, Z., et al.: Comparative Outcomes of Laminectomy for Spinal Stenosis- A Systematic Review. Neurosurgery (2016)
4. Han, H., and Park, J.: Robot control near singularity and joint limit using a continuous task transition algorithm. International Journal of Advanced (2013)
5. Hetaimish, B.: Sawbones laboratory in orthopedic surgical training. Saudi Medical Journal (2016)
6. Kim, H.J., and Lenke, L. G.: Vertebroplasty and Kyphoplasty: Innovations and Controversies Neurosurgical Focus (2012)
7. Siciliano, B., et al.: Springer handbook of robotics. Springer (2008)
8. Lachner, J., Schettino, V. et al.: The influence of coordinates in robotic manipulability analysis. Mechanism and machine theory (2020)
9. Malham, G. M., and Wells-Quinn, T.: What should my hospital buy next?—guidelines for the acquisition and application of imaging, navigation, and robotics for spine surgery. Journal of Spine Surgery (2019)
10. Osorio, J. D. M.: A Hierarchical Framework for Physical Human-Robot Interaction. PhD thesis (2022)
11. Schettino, V., et al.: Geometrical Interpretation and Detection of Multiple Task Conflicts using a Coordinate Invariant Index. International Conference on Intelligent Robots and Systems (2020)
12. F. Ficuciello, L. Villani, B. Siciliano: Variable Impedance Control of Redundant Manipulators for Intuitive Human–Robot Physical Interaction doi: 10.1109/TRO.2015.2430053 31 (4), 850-863
13. P. De Risi, F. Amadio, G. Garofalo, F. Ficuciello and P. Falco, “Variable Impedance Control Combining Reinforcement Learning and Gaussian Process Regression,” 2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids), Nancy, France, 2024, pp. 343-350,