



## Paper

## Active constraint control for the surgical robotic platform with concentric connector joints



Samir Morad<sup>a,\*</sup>, Christian Ulbricht<sup>b</sup>, Paul Harkin<sup>c</sup>, Justin Chan<sup>c</sup>, Kim Parker<sup>c</sup>, Ravi Vaidyanathan<sup>c</sup>

<sup>a</sup> The University of East London, University Way, London, E16 2RD, UK

<sup>b</sup> Charing Cross Hospital, London, W6 8RF, UK

<sup>c</sup> Imperial College London, London SW7 2BX, UK

## ARTICLE INFO

**Keywords:**

Surgical robots  
Haptic feedback  
Active constraint  
Control system

## ABSTRACT

Robotic minimally invasive surgery (MIS) has changed numerous surgical techniques in the past few years and enhanced their results. Haptic feedback is integrated into robotic surgical systems to restore the surgeon's perception of forces in response to interaction with objects in the surgical environment. The ideal exact emulation of the robot's interaction with its physical environment in free space is a very challenging problem to solve completely. Previously, we introduced the surgical robotic platform (SRP) with a novel concentric connector joint (CCJ). This study aims to develop a haptic control system that integrates an active constraint controller into a surgical robot platform. We have successfully established haptic feedback control for the surgical robot using constraint control and inverse kinematic relationships integrated into the overall positioning structure. A preliminary feasibility study, modelling, and simulation were presented.

## 1. Introduction

## 1.1. Surgical robotics

In recent years, there has been a significant increase in interest in the field of robotic surgery, both in terms of financial investment and research. Robots have been created for a variety of surgical procedures, and they are more accurate overall than traditional techniques because of their various sizes and cutting abilities [1].

Robotic surgery promises greater dexterity and greater accuracy by minimizing human error and incorporating myriad other surgical methods that can be performed with the help of a computer. Surgical robots are designed to improve surgical success by increasing fidelity and thus minimizing patient trauma, but a lack of power and tactile feedback appears to limit clinical success [2]. It has been found that surgeons using robotic surgical probes find that the lack of feedback eliminates tactile cues, masks force cues, and in some cases increases intraoperative injury [3,4]. Widespread adoption of surgical robots is therefore unlikely unless a way is found to address arguably the biggest

problem in surgical robotics, the lack of touch.

## 1.2. Haptic technology

In general, haptics refers to the simulated interactions of humans, robots, and simulated or remote environments [5]. The biggest obstacle to the widespread use of the surgical robot is the inability of the surgeon to feel the surgical environment through the use of robots. Haptic technology aims to solve this problem by providing cutaneous (tactile) and kinesthetic (force) feedback to the user.

Using kinesthetic haptics, a robot simulates touch interaction with a patient and transmits important information about the operating environment to the surgeon. As a result, kinesthetic haptic positional feedback control is used to assist the user in safely and successfully performing certain operations using a robotic manipulator.

The integration of haptic feedback into teleoperated robotic surgical systems poses a notable difficulty since system communication time delays induce a trade-off between transparency and stability. By integrating an environment estimation and force prediction methodology

**Place of work:** The Department of Mechanical Engineering, Imperial College London, South Kensington Campus, London SW7 2BX, United Kingdom.

\* Corresponding author at: The Department of Engineering, The University of East London, University Way, London, E16 2RD, UK.

E-mail addresses: [s.morad@uel.ac.uk](mailto:s.morad@uel.ac.uk) (S. Morad), [Christian.Ulbricht@imperial.nhs.uk](mailto:Christian.Ulbricht@imperial.nhs.uk) (C. Ulbricht), [paul.harkin@formchangingstructures.com](mailto:paul.harkin@formchangingstructures.com) (P. Harkin), [justin\\_chy@hotmail.com](mailto:justin_chy@hotmail.com) (J. Chan), [k.parker@imperial.ac.uk](mailto:k.parker@imperial.ac.uk) (K. Parker), [r.vaidyanathan@imperial.ac.uk](mailto:r.vaidyanathan@imperial.ac.uk) (R. Vaidyanathan).

<https://doi.org/10.1016/j.medengphy.2024.104236>

Received 29 December 2023; Received in revised form 28 August 2024; Accepted 2 September 2024

Available online 2 September 2024

1350-4533/© 2024 The Author(s). Published by Elsevier Ltd on behalf of IPEM. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

into an experimental robotic minimally invasive surgical system, these time delays are reduced [6].

The particular instance of laser microsurgery assisted by a robot is examined by Olivieri et al. [7]. Here, stereoscopic visualisation and 3D reconstruction are used to create a fictional force feedback that enables the surgeon to manipulate a non-contact surgical laser while haptically sensing the operative area.

In their study to investigate the effect of haptic feedback on suturing accuracy and performance, Ehrampoush et al. [8] presented a new force-sensing and semi-automated robotic needle driver for teleoperated MIS suturing activities. The proposed methodology offers the surgeon with two types of force information via the haptic feedback control architecture: virtual fixture force and needle-tissue contact force.

Souipas et al. [9] demonstrated the use of SimPS-Net, a monocular RGB (Red-Green-Blue) camera-based network, that can recognise and locate surgical instruments in three dimensions, to generate and enforce active constraints [10]. Real-time network deployment was done in order to define boundaries. Later on, this boundary was employed to test constraint enforcement. Two distinct active limitations were tested with the robot: a restricted region and a safe region.

In addition to measuring conventional grabbing forces, an actuated modular force feedback-enabled laparoscopic tool was proposed by Dalvand et al. [11] that can monitor the forces of tip-tissue lateral interaction. Additionally, the device can change the direction in which it grasps the patient's body.

A master-slave robotic system for needle indentation and insertion is described by Shin et al. [12]. This robotic device can characterise touch interactions with soft tissue. For robotic indentation and needle insertion, a bilateral controller with a linear motor is used. A new nonlinear state observer is created by combining unscented Kalman filter (UKF) with the Hunt-Crossley (H—C) model to estimate the contact interaction with soft tissue in real time.

A new technique is proposed by Abeywardena et al. [13] for estimating the tool-tissue force interaction in robot-assisted minimally invasive surgery that does not rely on external force measurement. To estimate the force interaction, the suggested technique uses the current from the surgical instrument's motors as well as neural network algorithms. Offline and online testing are carried out to determine the practicality of the created algorithm. The authors demonstrated that the devised approach shows promise in allowing for a live estimate of tool-tissue force, potentially enabling haptic feedback in robotic surgery.

Wearable Haptic Devices (WHDs) such as Hand Exoskeletons (HE) have emerged as the future focus for haptic devices. WHDs such as wearable Fingertip Haptic Device (FHD) [14–16] can provide more freedom of movement, mimic the hand movements of the operator and potentially remove the cognitive gap in tele-operation.

In robotic-assisted MIS, real-time identification of soft tissues is important to provide accurate force feedback during robot's surgical activities. In their study, Zhu et al. introduces an online approach for soft tissue characterisation. Online soft tissue parameter identification is created using the extended Kalman filter (EKF). It based on the nonlinear Hunt-Crossley (H—C) contact model being dynamically linearized with respect to the system state [17,18]. A technique for realistic modelling of human soft tissue, incorporating dynamic soft tissue characterisation was introduced by Song et al. [19]. It uses maximum likelihood theory, nonlinear filtering, and the nonlinear finite element technique (NFEM) to describe nonlinear tissue deformation behaviour. This method helps identify mechanical characteristics and estimate soft tissue deformation behavior, creating a nonlinear state-space equation based on hyperelasticity.

In comparisons to the above existing haptic technology, guidance constraints and regional constraints are considered in this paper. Guidance constraints are used to guide the user through a path. This technique has many limitations when used to guide robotic surgical probes along convoluted pathways inside the human body. The probe

can only operate in a specific active space. It is important to keep the probe's degree of motion within the surgical environment. The paper is intended to set out the initial results of this control system, to set the scene for further analysis, simulation and testing.

### 1.3. Active constraints

It is typically assumed when examining the active constraint definition process that the constraint geometry has either been defined a priori or can be produced using a generated point-cloud [20]. Creating and combining primitive shapes (such as spheres, cylinders, etc.) is another common application. Then, by combining these basic forms, a more intricate active constraint geometry can be created [21]. However, it should be highlighted that creating the constraint geometry by itself is insufficient when investigating the idea of active constraint definition. It is necessary to "anchor" the created active constraint border to the tissue of the operated patient. By doing this, the produced geometry can be localised by any registered robotic system.

During the development of a prostatectomy robot, an active constraint theory was put into practice [22]. The mechanical constraint was further developed to be used in knee replacement surgery.

Three zones were defined for bone removal. A robot in a safe zone I can be moved by an operator. The power of the motors is turned on when entering transition zone II and moving towards the cutting boundary. The stationary position command is set in the direction of the boundary in accordance with the active constraint. The cutter is prevented from entering the forbidden zone III by using this control concept.

An illustrative example of constrained control is provided [23]. Staying on a point, avoiding plane penetration, moving along a line and rotating around a line are some of the geometric constraints that are relevant to surgical robots. M. Scheint et al. [24] studied trajectory supervision and rendering with invariance control. An example of regional constraint implementation is presented.

The laser interferometry-based sensing and measuring (LISM) approach was initially studied for dynamic measurements of the end effector of a robot manipulator in motion. This approach may give dynamic position measurements in real time while maintaining high precision, a vast working space, a high sample rate, and autonomous target tracking [25]. An approach based on the LISM technique is suggested to provide laser interferometry-based guiding (LIG) for precise placement of a robot manipulator in high precision manufacturing processes. The LISM equipment is used in the approach to guide the robot's end effector to a desired place or along a desired path by guiding the robot to follow the laser beam-mapped trajectory [26].

In their work, Clark et al. [27] describes a novel approach for sensing position and orientation in micro/nano positioning stages with three planar degrees of freedom (DOF). To prevent misalignment, the mechanism is counter-rotated utilizing a laser interferometry-based sensing approach. A position-sensitive diode sensing approach identifies misalignment, while a feed-forward feedback compound controller minimizes misalignment errors. The approach was experimentally validated on a three DOF planar micro/nano positioning stage, confirming its capacity to give both linear and angular measurements.

The purpose of this paper is to explore the design of a working positional control structure incorporating a robotic framework, developed for use in spine surgical procedures, which serves as the guide and moving mount. In order to simplify the mechanical design of the framework. The parallel manipulator uses novel concentric joints [28]. The design of a surgical tool that can remove tissue from both the anterior and the posterior sides of the spine will be developed in a separate paper. This paper is part of a PhD thesis defended by the author [29].

## 2. Control system

### 2.1. System definition

The SRP has six prismatic actuators that actuate an end-effector independently and separately. In this study, six extensible legs were controlled by direct current (DC) motors to move an articulating surgical probe in 6DOF: three linear movements (latitude, longitudinal and vertical) and three rotations (pitch  $\Psi$ , roll  $\theta$  and yaw  $\Phi$ ). The SRP was mounted on a cuboid frame, inverted to mimic the way it would be positioned in the operation theatre, above the patient [28].

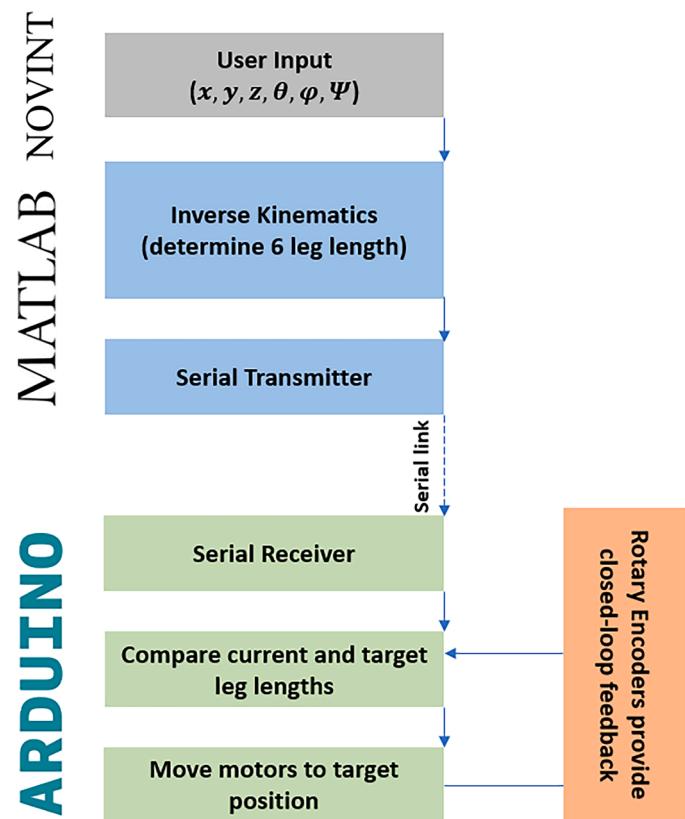
The mobile plate of the SRP has a surgical probe attached to it. It has a high-pressure water jet tool that can be used to remove human tissue, and is intended to be introduced in detail in a separate paper.

The user-side controller generates kinesthetic haptics (force feedback) through the implementation of the 3DOF Novint Falcon haptic enabled controller (NF) to control the three translational inputs [30]. A graphical user interface (GUI) was generated to control the three rotational input values.

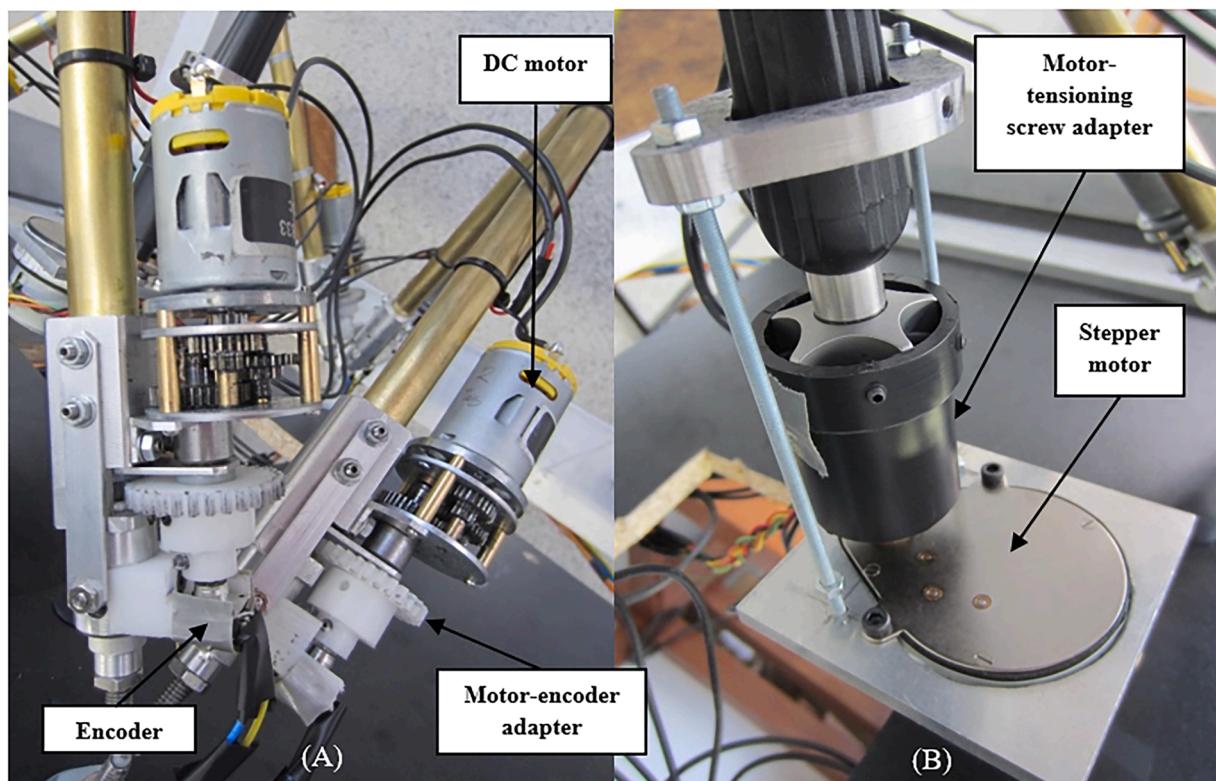
Each of the six DC motors was connected to a rotary encoder via a custom-made adapter to provide positional feedback regarding each leg (Fig. 1-A). A unipolar stepper motor was controlled by a single ULN2003a transistor array, which was connected to the flexible probe (Fig. 1-B).

### 2.2. Position control of the SRP

With the NF, user inputs of up to 3DOF can be made and mechanical force feedback can be passed back to the user. Fig. 2 shows the structure of the code used to translate this user input into meaningful motion for the platform. Programming begins with the definition of global variables, inputs, and outputs. In the main loop, the program continues to read the positional input from the NF and controls the motors, adjusting the six-leg lengths as necessary, to achieve the desired position for the



**Fig. 2.** Code structure used to process the user input (NF) into the meaningful motion of the robot platform. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 1.** The installed motor-encoder units on the platform legs via motor-encoder adapter (A). The stepper motor attached to the probe tensioning screw via a motor-tensioning screw adapter (B). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

end effector.

The program calculates each leg's target length using inverse kinematics, using a rotational matrix and positional matrix. Using six encoders, one on each leg, the position control creates a feedback loop. At the beginning of the program, the legs are datumed with a home flag, which indicates the base's initial position. The encoder readings in relation to this home position value represent the length of each of the six legs of the platform.

In Fig. 3, the electronics used as well as the associated software that implements full control of the position of the platform are shown. The figure clearly illustrates how the electronic components are connected, powered and controlled from the input side to the output side.

### 2.3. Inverse kinematics

The input controller specifies the position of the end-effector in the positional control loop. In this scenario, the user specifies the orientation and position of the flexible probe in 6DOF. To impose the desired end-effector orientation on the robot platform, each of the six legs of the platform needs to be calculated to be the required length. In their paper [28], the authors introduce the full details of the robot platform's kinematics, and they formulate the inverse kinematic relationships for a 3–3 surgical robot platform.

### 2.4. Simulation

As a validation of the inverse kinematics derived in [28], a MATLAB simulation was created of a robot model receiving inputs from an NF to model the movement of a 6DOF model. The desired position ( $x, y, z$ ) and orientation ( $\varphi, \theta, \Psi$ ) of the end-effector was specified and the nominal leg lengths of each of the six platform's legs were calculated. The leg lengths were plotted in a real-time 3-D model (Fig. 4).

Simulation is structured into three main subsections: the model construction, the GUI initialization, and the draw function. In model

construct script, the figure is initialized and created. A fixed coordinate system is used to define the manipulator geometry including the tool, base and legs. In the next script, the GUI is initialized. A menu bar, labels, and position values indicating the orientation and location of the end effector are created in the on-screen user interface. Finally, MATLAB reads the input from the NF and performs the inverse kinematics to get the six-leg lengths. In the draw function, the robot platform figure is continuously updated in real-time as the simulation continues.

Regional constraints were implemented by generating force feedback when an inadmissible position was reached. Due to the physical limitations of the robot's actuation and/or geometry, some robot manipulators have a fixed workspace beyond which they will not be able to reach. Therefore, kinesthetic haptics, or active constraints, are used to keep the user within a certain predefined spatial area, namely the robot platform's workspace.

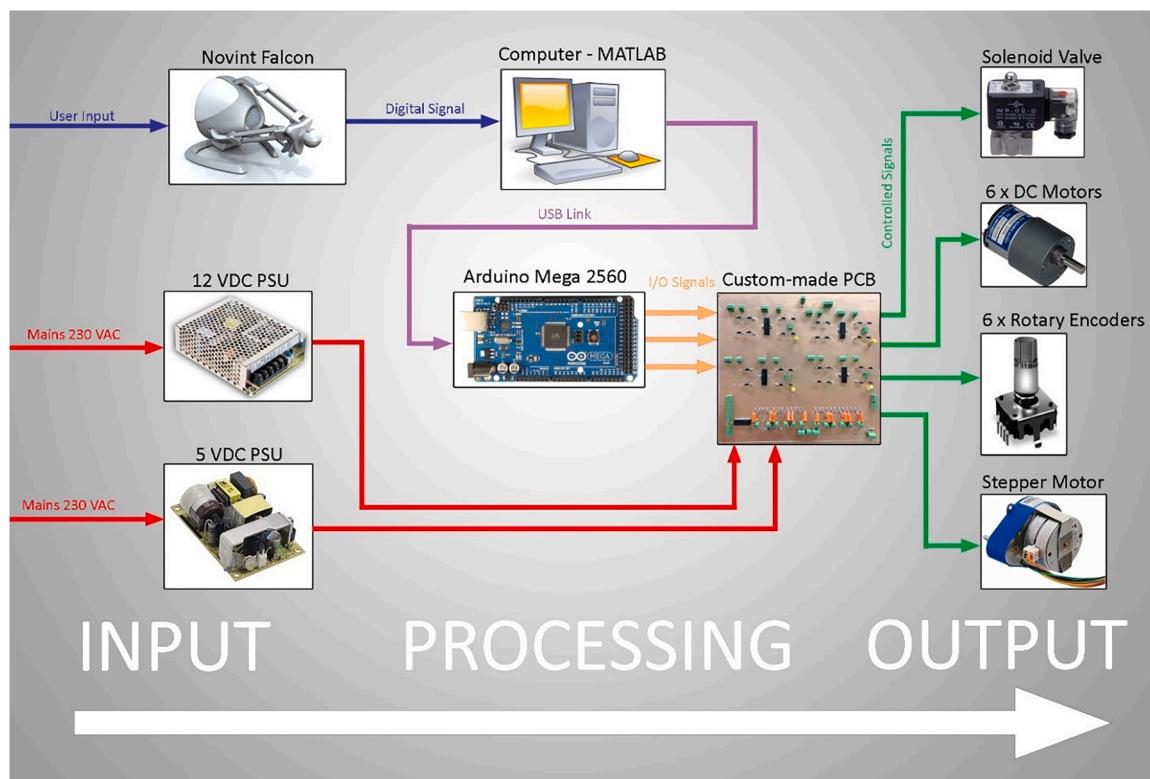
## 3. Haptic modelling

### 3.1. Haptic rendering

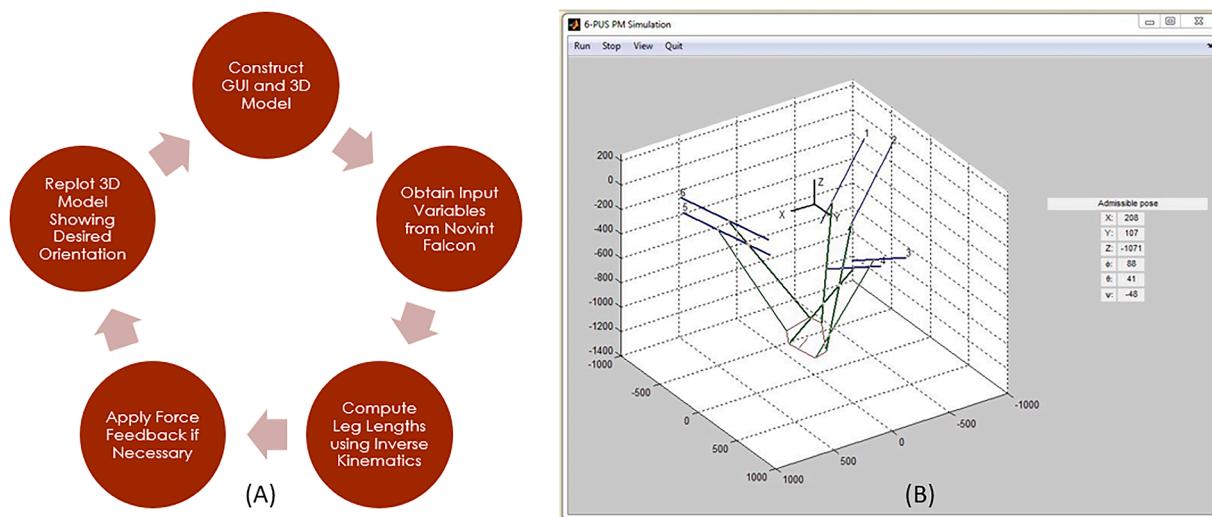
Haptic rendering enables the surgeon to perceive forces in the surgical environment in response to interaction with objects. From a control theory perspective, haptic rendering is a problem with constrained control, since rigid walls are rendered. A set of constraints and boundaries that are impermissible or inadmissible to the robot are specified. Fig. 5 shows the basic structure of the control loop used for haptic rendering.

#### 3.1.1. Control loop structure

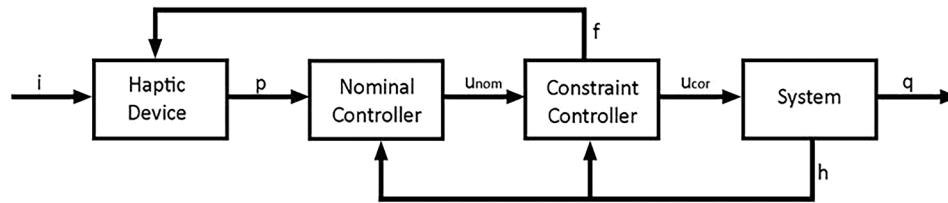
A position and orientation of the end-effector are specified by an input set ( $p$ ) to the nominal controller (NC). Inverse kinematics and all necessary calculations are performed by the NC to determine the nominal control signals ( $u_{nom}$ ) for the system to achieve the desired position without taking into account any constraints. The constraint controller



**Fig. 3.** The component layout of the surgical robot platform (SRP) system. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 4.** Logic flow chart of the simulation (A) and the resultant simulation of the robot platform displayed in MATLAB GUI (B). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 5.** Control loop structure for haptic rendering. User input information ( $p$ ) is sent to the nominal controller (NC) to provide signals ( $u_{nom}$ ) to control the system's output within permissible regions. Otherwise, the constraint controller is involved to generate forces ( $f$ ) that keep the system's output within the permissible region.

(CC) monitors the state of the system and only activates when the system approaches the boundaries of a forbidden region. By modifying  $u_{nom}$  and generating a corrected control signal ( $u_{cor}$ ), the CC prevents the system from entering the forbidden area. In addition, it computes feedback force whose magnitude is proportional to the severity of the constraint exceeded.

In addition to determining the appropriate direction of force insertion, the CC also decides in which direction this force should be applied, which is in the opposite direction of the direction of error, normal to the boundary at which the constraint is applied. The force vector  $f$  is then sent to the haptic device to provide corrective kinaesthetic haptic feedback to the user. A closed-loop feedback control structure of the NC and CC monitors the real-time state of the system, in this case, the orientation of the robot platform.

### 3.1.2. Constraint set

To test the control loop, a virtual environment is created with basic constraints. Through modifying the constraint set by which the CC samples, the output of the haptic device and the robot platform can be controlled. In [23], the methods of superposition and linear combination can be used to define any boundary system. There are five constraints: point limit, cuboid constraint, plane constraint, cylindrical constraint, and spherical constraint (Table 1).

$$f_i = -(i - LB_i), u_{icor} = LB_i, \text{ for } i < LB_i \quad (1)$$

$$f_i = -(i - UB_i), u_{icor} = UB_i, \text{ for } i > UB_i \quad (2)$$

$$f_i = 0, u_{icor} = i$$

A haptic device produces force feedback in three translational

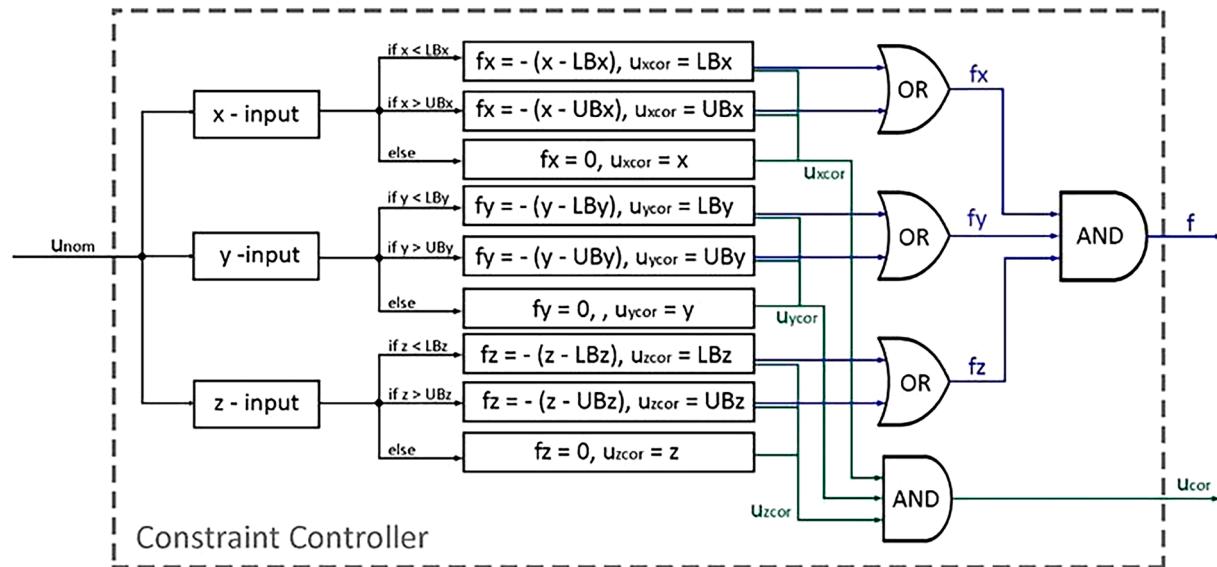
**Table 1**  
The five basic types of constraints.

Constraint type	Description	Example of physical uses
Point limit	Limits the robot end effector to a particular point	A set of point limits can be superimposed to create a 3D path
Cuboid Constraint	Restricts the robot to motion within a cuboid-shaped region	Used to define the robot's workspace
Plane Constraint	Prevent the robot from penetrating a plane	A combination of planes can be generated to recreate a particular environment
Cylindrical Constraint	Confines the robot to a cylindrical region	Linear combination with point limit to give hard and soft constraint for motion along a path
Spherical Constraint	Defines a critical spherical region	Limiting the robot's functional space to a 3D curved surface

coordinates (x, y and z), so there are 3D boundary conditions for each constraint. The CC block in Fig. 5 is expanded to show how the code within this program determines the appropriate force output  $f$  and the  $u_{cor}$  to the output. Fig. 6 shows the control logic.

Eqs. (1) and (2) are the mathematical equations governing the constraint controller, where  $UB$  and  $LB$  are the upper and lower boundaries respectively,  $f$  is force magnitude, and  $i$  the position along each of the Cartesian axis x, y, z.

As an example of a geometric constraint, cylindrical constraints can be defined using 2D polar coordinates (Fig. 7). The associated equations for a cylinder with radius  $r$  centered about the origin O are:

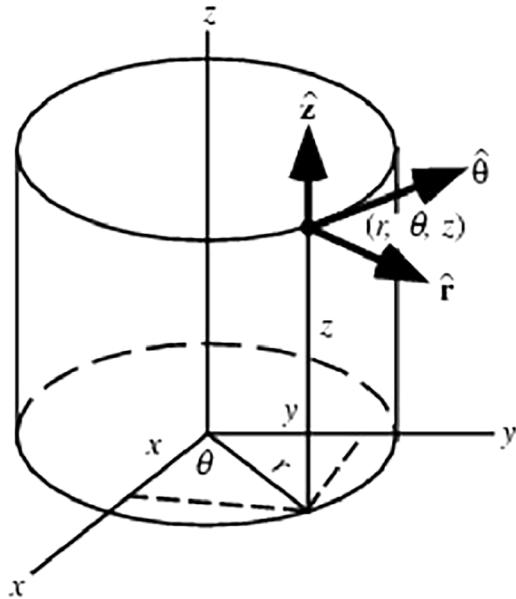


**Fig. 6.** Constraint controller control logic. The (NC) signals are used to control the output of the system within the pre-defined boundaries. The constraint controller becomes active at the boundaries of the constraint region, generating forces that keep the system within pre-defined boundaries.

$$r = \sqrt{x^2 + y^2} \quad (3)$$

$$\theta = \tan^{-1} \left( \frac{y}{x} \right) \quad (4)$$

**Table 2** present the constraints of the cylinder, including LB and UB of the Cartesian coordinates ( $x$ ,  $y$  and  $z$ ). From this, the direction of the force vector depends on which of the four quadrants the robot is in, so as to act to force the user inwards towards the centre of the cylinder. By inverting the constraint set, the constraint controller will keep the robot outside the boundaries.



**Fig. 7.** A cylinder with radius  $r$  centred about the origin O.

### 3.2. Trajectory supervision

Since the robot's trajectory is not designed to run autonomously from one point to another, there is no need to model the robot's trajectories. Due to the real-time nature of the robot's application, it is sufficient to ensure that the end effector's position is within an allowable threshold of the input position. Since the input control signals are monitored by the constraint controller; The robot end effector's trajectory or path will never enter an inadmissible region as long as the end effector's real-time position is always equal to the input. In other words, if the real-time position of the robot end effector is always equal to the corrected control position signal, the end effector will never enter a restricted area because the corrected control position signal will never output an inadmissible signal.

Thus, the robot's trajectory monitoring is achieved by incorporating a closed-loop feedback mechanism. In this case, a proportional-integral-derivative (PID) controller was implemented to ensure that the robotic platform's leg lengths increase to the desired length in the optimal time. This is illustrated by signal  $h$  in Fig. 5. The constraint controller contains two additional constraint sets: the first, a speed limit set, and the second, an allowable threshold set.

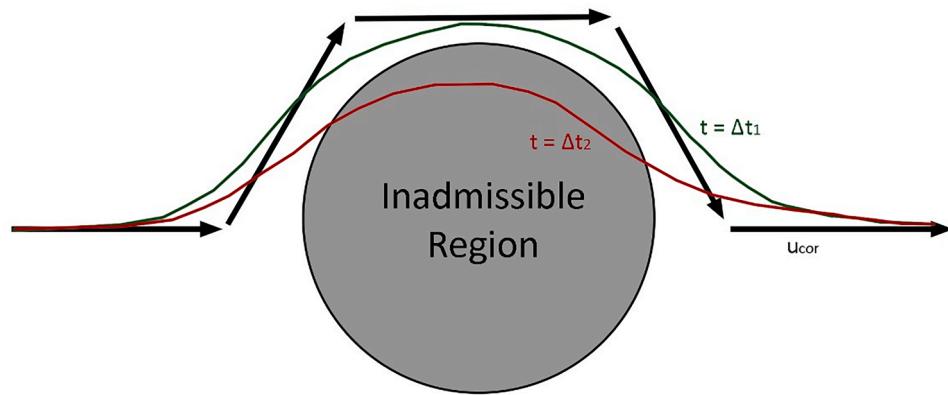
#### 3.2.1. Velocity limit

The system needs time to accelerate and decelerate to achieve target alignment. If the rate of change of the entered targets is critically high, the system will not have enough time to acceptably respond to the target orientations.

In Fig. 8, the black arrows show the same path inputs from  $u_{cor}$ . Due to the constraint control, the arrows will never enter the forbidden area.

**Table 2**  
Constraint set for cylindrical constraint.

Axis	Region	Value
x	Upper boundary, $UBx$	$r \times \cos \theta$
	Lower boundary, $LBx$	$-r \times \cos \theta$
y	Upper boundary, $UBy$	$r \times \sin \theta$
	Lower boundary, $LBy$	$-r \times \sin \theta$
z	Upper boundary, $UBz$	$z_5$
	Lower boundary, $LBz$	$z_4$

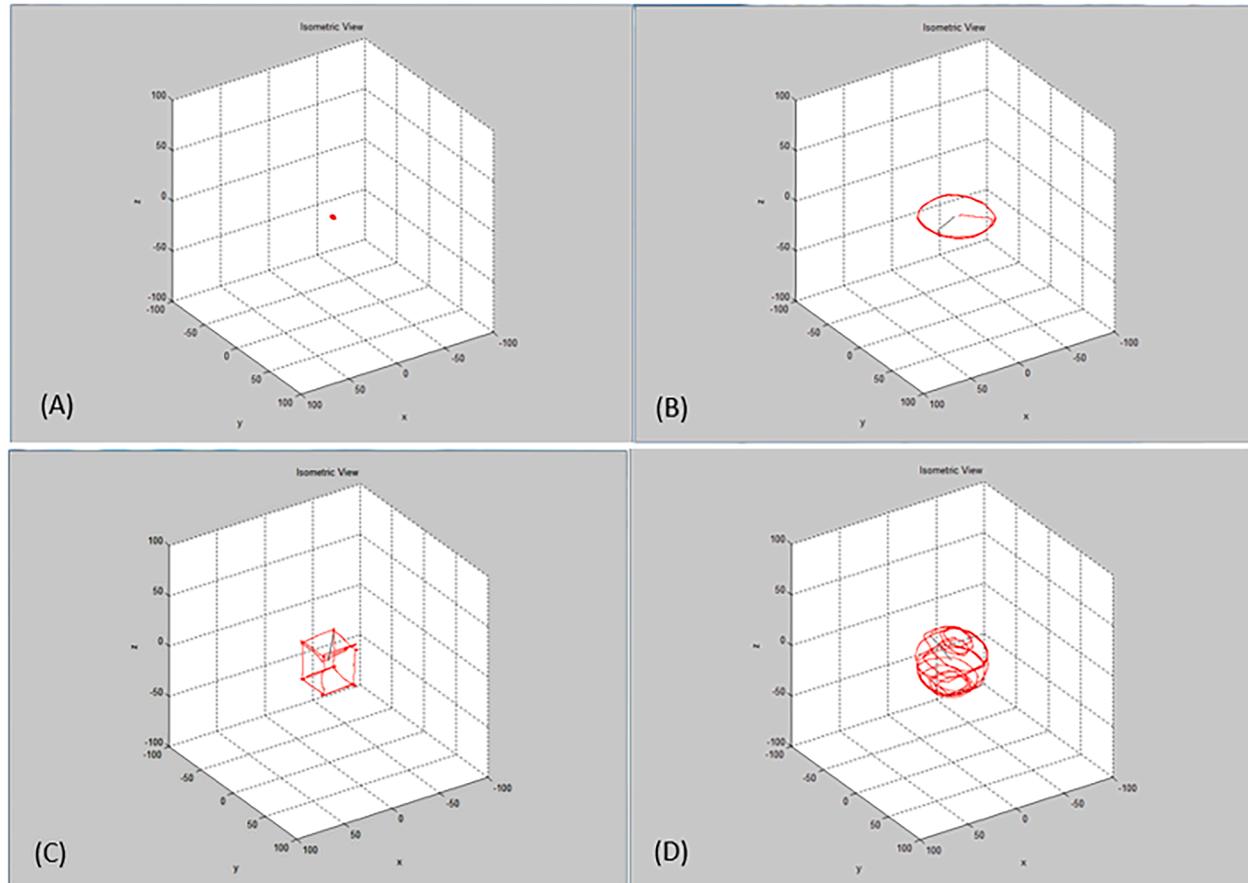


**Fig. 8.** Illustrative diagram showing the effects of input signal velocity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In the figure, two different instances of the system response trajectory are shown. The green line trajectory corresponds to input speeds below the critical speed. In this case, the PID controller maintains the robot's trajectory within a threshold acceptable to the input signal in real time, allowing the system to react in time. Notice what happens when the input speed is increased to a speed higher than the critical speed. As a result, a red line represents the robotic end effector's path through the prohibited area. To avoid this, the input velocity must be limited to be below the critical value at all times. This is done by incorporating a velocity limit constraint that generates a haptic force that slows the user down if the critical velocity is exceeded.

### 3.2.2. Permissible threshold

Fig. 8 shows that the robot's real-time trajectory does not exactly match the control signal trajectory  $u_{cor}$ . By reducing the input velocity,  $\Delta t$  will increase, and the robot's trajectory will closely follow the input signal. Since the input signals are constantly changing, the robot's path cannot correspond to them. Consequently, the system will always have an error as long as the inputs change, since the system requires time to react. In order to correct the motion, a suitable acceptable error threshold must be established, beyond which the constraint controller must intervene. Through empirical testing integrated with MATLAB GUI demonstrating a geometrical constraint of various shapes (Fig. 9), 10 mm has been found to be an acceptable value for this threshold. The NF



**Fig. 9.** MATLAB demonstration of various geometrical constraints. (A) point, (B) circle, (C) box, and (D) sphere. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

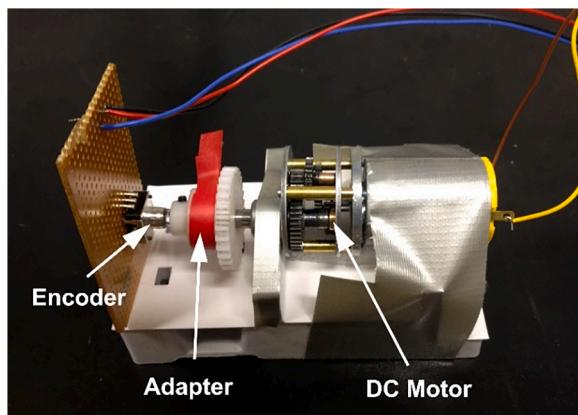
has 3 actuators which work together to exert a push/pull force in 3D space. In essence the 3 curved arms of NF have tendons connecting them to 3 motors (1 each). The software receives inputs of force vectors and magnitudes in the form of a 3 by 1 array (e.g.  $[F_x \ F_y \ F_z]$ ) and applies its proprietary calculations to generate the appropriate actuation of the 3 motors to generate the desired force. This is felt by the user as a pushing/pulling force. The controller is designed to output force feedback that is proportional to the error hence, if the user moves the controller to outside the permissible zone, the force will be large and the vector will act to force (push/pull) the user's hand back into the permissible zone. As the user approaches the boundary the force decreases gradually since the magnitude of error decreases as well, until the user is on the boundary and the error is reduced to 0 where the force vector also becomes 0 and the controller ceases to exert a force. This ensures an optimal balance between the allowable speed and precision of the end effector.

#### 4. Motor-Encoder testing and results

Rotary encoders are notoriously susceptible to signal noise. When the quadrature encoder is rotated, it acts like two switches producing cyclical outputs operating at  $90^\circ$  out of phase with one another. It is this phase shift that allows the software to determine if the shaft is rotating clockwise or anticlockwise. These switches produce noise when they latch on and latch off. The extent of the noise is a property of the particular encoder in use. In our case, Tyco Electronics DPL12 Rotary Encoder produces chattering noise and rotational noise. These signal artifacts, if not dealt with, falsely trigger the software interrupts which lead to highly inaccurate encoder readings. There are two ways of eliminating the signal noise: hardware filter, also known as debouncer circuits, and software filter. Both methods were employed in the control structure to ultimately produce a robust and accurate encoder. The software debouncer was implemented by introducing a simple timer script that waits 3.0 ms every time the interrupt is called before checking the signal again to ascertain if the signal was noise; disregarding it if it is. Individual functional cells were tested for accuracy and robustness before being assembled into the robot. There are, in total, six motor-encoder functional cells incorporated in the robot. Fig. 10 shows the test cell setup, which consists of a DC motor, rotary encoder, and adapter. From the encoder datasheet, one revolution of the encoder produces 96 pulses.

##### 4.1. Motor-Encoder cell accuracy with filter circuit

To test the accuracy of the function cell, a series of target encoder pulses were sent to the DC motor. The response variable in this experiment is the speed of the DC motor. The motor speed was held constant at



**Fig. 10.** Motor-encoder test cell. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

60 rpm, which is the rated speed of the quadrature encoder. The results (Fig. 11) clearly show the effectiveness of the filter circuit; the resulting errors with the filters are reduced by over 90 %.

##### 4.2. Effects of increasing motor speed

The encoder errors were also tested as a function of motor speed measured in revolutions per second (rpm). To control the motor's speed, pulse width modulation was used. The results for three different motor speeds are shown in Fig. 12. The error increases by approximately 0.1 % when the speed is increased from 60 rpm to more than three times its nominal speed, 200 rpm. Due to the encoder running at more than three times its rated speed, there is a large increase in error. At 400 rpm, the increase is slightly smaller, about 0.03 %. Accordingly, the errors caused by the encoder running above its nominal speed are relatively constant at higher speeds.

#### 5. Discussion

For minimally invasive procedures, a surgical robotic platform with a novel concentric joint had previously been introduced. The purpose of this study was to develop and test a working position control structure for a surgical robot that incorporates haptic technology. The inverse kinematic equations were validated using a computer simulation model developed in MATLAB. A position control code with haptic rendering and trajectory monitoring was successfully implemented using the required hardware, software, and electronics. Tests were conducted to determine the accuracy and robustness of the system.

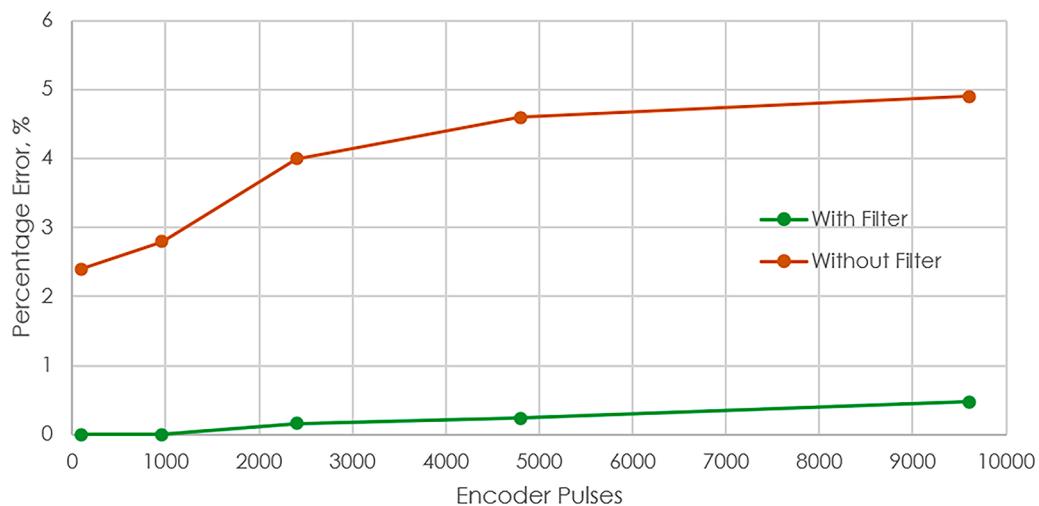
During the robot test, it was found that the probe's end effector was very sensitive to the movements of the input device, the NF. This problem could be solved by introducing a scaling system into the control scheme [31]. This allows the surgeon to control the robotic end effector at the microscopic level by scaling down the macroscopic movements of the system input device. An example of what the logic of the scaling code would look like is shown in Fig. 13.

The scaling controller sits between the input devices and the nominal controller. Signal  $p$  is modified to produce signal  $\hat{p}$ , which is the scaled control output signal. For macro-scale,  $\hat{p} = p$ , and for micro-scale,  $\hat{p} = p/100$ . Any level of control accuracy can be achieved by adjusting this scaling factor. Scaling control improves the surgeon's dexterity to previously unattainable levels of precision.

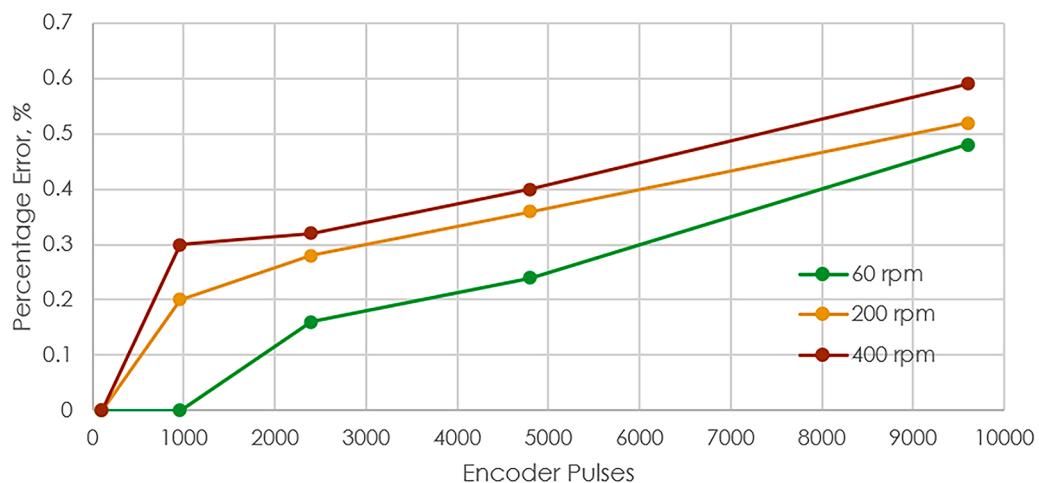
The NF can only input 3DOF. This does not provide a complete definition of the robot's end effector orientation. A MATLAB GUI was created in this study to specify the remaining 3DOF, partially resolving the problem. Future versions of this device should use a 6DOF haptic input device, such as Phantom Omni, to replace the NF and GUI. As a result, controlling the robot becomes more intuitive and reduces cognitive load for the surgeon.

It was determined that a single motor revolution corresponds to a 3.1 mm change in leg length based on the dimensions of the gears in the robot's leg. It would take 59.68 turns (5730 encoder pulses) to span the entire leg range since the leg range is ( $L_{max} - L_{min} = 491 - 306 = 185$  mm). As a result of the filter circuit, a percentage error of approximately 0.2 % is expected. The pulses correspond to 11.46 revolutions, or 0.37 mm, or 0.11937 pulses. For this study, this is an acceptable level of accuracy with a very small error. The observed rotation values were always less than or equal to the expected values. In this case, noise is the main cause of the error.

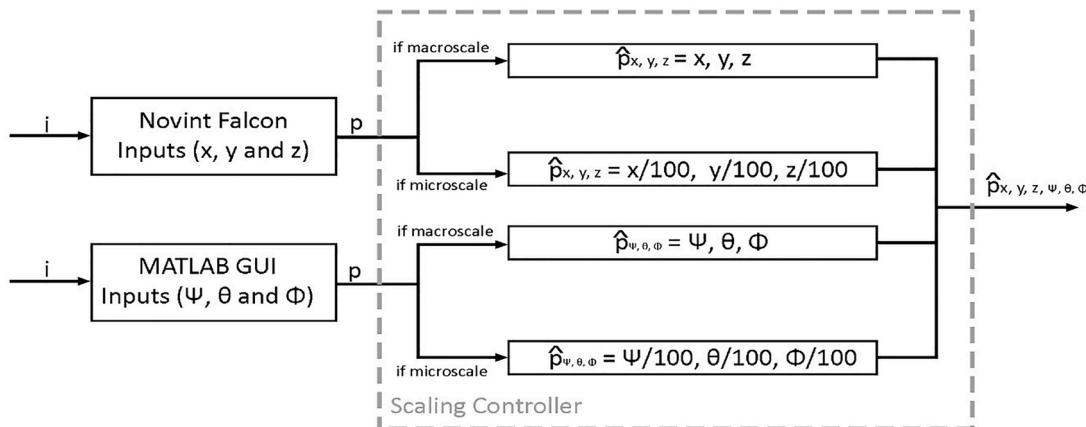
The motor should run at as high a speed as possible. With a higher motor speed, the leg would reach a target extension or retraction in less time, thus decreasing the system's response time. The error introduced by a DC motor speeding at 400 rpm is 0.45 % per leg span (185 mm) or 0.832 mm. Compared to an encoder operating at a nominal 60 rpm, this represents a 125 % increase in linear position error. Although the percentage is large, the absolute value of the error is minimal. It is on the



**Fig. 11.** Graph of percentage errors against the number of encoder pulses for motor test cell with filter and without a filter. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 12.** Graph of percentage error of rotation against encoder pulses for different motor speeds. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 13.** Scaling controller logic flow chart. Position and orientation information from the user interface (P) will be scaled down by the scaling controller before entering the nominal controller.

order of  $10^{-4}$  m. It is therefore acceptable to run the motor at maximum speed.

For extending and retracting its legs, the robotic platform was equipped with DC geared motors. Despite being inexpensive, DC motors cannot instantly adjust the length of the legs. As a result, tracking the trajectory was particularly difficult. Trajectory path control is achieved by limiting input speed to a certain level. The system then has sufficient time to respond to the input signals. Due to the slow leg extension caused by DC motors, the threshold was very low. As a result, the input signals were also limited to a very slow rate. It is also possible to use pneumatic or linear drives instead of DC motors. In the surgical application at hand, these actuators provide sufficient accuracy and holding torque to provide desirable response times.

## 6. Conclusion

We present the design and development of a working position control structure that integrates haptic technology into the surgical robotic platform on the basis of our previous work on a surgical robotic platform with a concentric connecting joint. A surgical robot's position control is integrated with constraint control and inverse kinematics, resulting in haptic feedback control. We developed a computer simulation model in MATLAB and validated the inverse kinematic equations based on it. A position control code incorporating haptic rendering and trajectory monitoring was successfully implemented using the required hardware, software, and electronics. Tests were conducted to determine the accuracy and robustness of the system. Future work will improve the design and further investigate the implementation of haptic technology in surgical robots based on the findings of this study.

Future work will involve designing and developing a flexible robotic surgical device that includes a water jet tool and an articulated distal tip (ADT).

## Declaration of competing interest

Authors declare that they have no conflict of interest.

## Acknowledgements

This is a summary of independent research funded by the UK Engineering and Physical Sciences Research Council (EPSRC).

## Funding

The UK Engineering and Physical Sciences Research Council (EPSRC) – the United Kingdom, funded this study (Funder's Grant Number: ME-ME F46009).

## Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

## References

- [1] Walgrave S, Oussedik S. Comparative assessment of current robotic-assisted systems in primary total knee arthroplasty. *Bone Jt Open* 2023;4:13–8. <https://doi.org/10.1302/2633-1462.41>.
- [2] Okamura AM. Haptic feedback in robot-assisted minimally invasive surgery. *Curr Opin Urol* 2009;19:102–7. <https://doi.org/10.1097/MOU.0b013e32831a478c>.
- [3] Chand M, Bhoday J, Brown G, Moran B, Parvaiz A. Laparoscopic surgery for rectal cancer. *J R Soc Med* 2012;105:429–35. <https://doi.org/10.1258/jrsm.2012.120070>.
- [4] Aly EH. Laparoscopic surgery for rectal cancer: approaches, challenges and outcome. editor. In: Ho Y-H, editor. *Contemporary issues in colorectal surgical practice*. Rijeka: InTech; 2012. p. 57–72. <https://doi.org/10.5772/33501>.
- [5] Hannaford B, Okamura AM. *Haptics*. In: Siciliano B, Khatib O, editors. *Springer handbook of robotics*. Berlin: Springer; 2008. p. 719–39.
- [6] Batty T, Ehrampoush A, Shirinzadeh B, Zhong Y, Smith J. A transparent teleoperated robotic surgical system with predictive haptic feedback and force modelling. *Sensors* 2022;22:9770. <https://doi.org/10.3390/s22249770>.
- [7] Olivieri E, Barresi G, Caldwell DG, Mattos LS. Haptic Feedback for control and active constraints in contactless laser surgery: concept, implementation, and evaluation. *IEEE Trans Haptics* 2018;11:241–54. <https://doi.org/10.1109/TOH.2017.2786243>.
- [8] Ehrampoush A, Shirinzadeh B, Pinskier J, Smith J, Moshinsky R, Zhong Y. A force-feedback methodology for teleoperated suturing task in robotic-assisted minimally invasive surgery. *Sensors* 2022;22:7829. <https://doi.org/10.3390/s22207829>.
- [9] Souipas S, Nguyen A, Laws SG, Davies BL, Baena FR. Real-time active constraint generation and enforcement for surgical tools using 3D detection and localisation network. *Front Robot AI* 2024;11:1365632. <https://doi.org/10.3389/frobt.2024.1365632>.
- [10] Souipas S, Nguyen A, Laws SG, Davies BL, Baena FR. SimPS-Net: simultaneous pose and segmentation network of surgical tools. *IEEE Trans Med Robot Bionics* 2023;5:614–22. <https://doi.org/10.1109/TMRB.2023.3291022>.
- [11] Dalvand MM, Shirinzadeh B, Shamdani AH, Smith J, Zhong Y. An actuated force feedback-enabled laparoscopic instrument for robotic-assisted surgery. *Int J Med Robot* 2014;10:11–21. <https://doi.org/10.1002/rco.1503>.
- [12] Shin J, Zhong Y, Gu C. Master-slave robotic system for needle indentation and insertion. *Comput Assist Surg (Abingdon)* 2017;22:100–5. <https://doi.org/10.1080/24699322.2017.1379236>.
- [13] Abeywardena S, Yuan Q, Tzemanaki A, Psomopoulou E, Droukas L, Melhuish C, Dogramadzi S. Estimation of tool-tissue forces in robot-assisted minimally invasive surgery using neural networks. *Front Robot AI* 2019;6:56. <https://doi.org/10.3389/frobt.2019.00056>.
- [14] Tzemanaki A, Al GA, Melhuish C, Dogramadzi S. Design of a wearable fingertip haptic device for remote palpation: characterisation and interface with a virtual environment. *Front Robot AI* 2018;5:311495. <https://doi.org/10.3389/frobt.2018.00062>.
- [15] Morad S, Jaffer Z, Dogramadzi S. Design of a wearable fingertip haptic device: investigating materials of varying stiffness for mapping the variable compliance platform. *J Med Robot Res* 2021;06:2150005. <https://doi.org/10.1142/S2424905x21500057>.
- [16] Morad S, Jaffer Z, Dogramadzi S. A novel mechanical design of a wearable fingertip haptic device for remote meniscus palpation. *J Med Robot Res* 2023;08:2350001. <https://doi.org/10.1142/S2424905x23500010>.
- [17] Zhu X, Gao B, Zhong Y, Gu C, Choi KS. Extended Kalman filter for online soft tissue characterization based on Hunt-Crossley contact model. *J Mech Behav Biomed Mater* 2021;123:104667.
- [18] Zhu X, Li J, Zhong Y, Choi KS, Shirinzadeh B, Smith J, Gu C. Iterative Kalman filter for biological tissue identification. *Int J Robust Nonlinear Control* 2023;1–13. <https://doi.org/10.1002/rnc.6742>.
- [19] Song J, Xie H, Zhong Y, Gu C, Choi KS. Maximum likelihood-based extended Kalman filter for soft tissue modelling. *J Mech Behav Biomed Mater* 2023;137:105553. <https://doi.org/10.1016/j.jmbbm.2022.105553>.
- [20] Sharp A, Pryor M. Virtual fixture generation for task planning with complex geometriesASME. *J Comput Inf Sci Eng* 2021;21. <https://doi.org/10.1115/1.4049993>.
- [21] Bettini A, Marayong P, Lang S, Okamura AM, Hager GD. Vision-assisted control for manipulation using virtual fixtures. *IEEE Trans Robot* 2004;20:953–66. <https://doi.org/10.1109/TRO.2004.829483>.
- [22] Yen P-L, Davies BL. Active constraint control for image-guided robotic surgery. *J Eng Med* 2010;224:623–31. <https://doi.org/10.1243/09544119JEM606>.
- [23] Kapoor A, Li M, Taylor RH. Constrained control for surgical assistant robots. *Proceedings of the international conference on robotics and automation*. Florida: IEEE; 2006. p. 231–6.
- [24] Scheient M, Wolff J, Buss M. Invariance control in robotic applications: trajectory supervision and haptic rendering. Seattle: American Control Conference; 2008. p. 1436–42.
- [25] Shirinzadeh B. Laser-interferometry-based tracking for dynamic measurements. *Indust Robot* 1998;25:35–41. <https://doi.org/10.1108/01439919810196946>.
- [26] Shirinzadeh B, Teoh PL, Tian Y, Dalvand MM, Zhong Y, Liaw HC. Laser interferometry-based guidance methodology for high precision positioning of mechanisms and robots. *Robot Comput Integr Manuf* 2010;26:74–82. <https://doi.org/10.1016/j.rcim.2009.04.002>.
- [27] Clark L, Shirinzadeh B, Tian Y, Oetomo D. Laser-based sensing, measurement, and misalignment control of coupled linear and angular motion for ultrahigh precision movement. *IEEE/ASME Trans Mech* 2015;20:84–92. <https://doi.org/10.1109/TMECH.2014.230182414361442>.
- [28] Morad S, Ulbricht C, Harkin P, Chan J, Parker K, Vaidyanathan R. Surgical robot platform with a novel concentric joint for minimally invasive procedures. *J Med Robot Res* 2021;5:e2050001. <https://doi.org/10.1142/S2424905x20500014>.
- [29] Morad S. Flexible robotic device for spinal surgery. London, UK: Imperial College London; 2015. <https://doi.org/10.25560/26878>. PhD thesis.
- [30] Block DJ, Michelotti MB, Sreenivas RS. Application of the Novint Falcon haptic device as an actuator in real-time control. *J Behav Robot* 2013;4:182–93. <https://doi.org/10.2478/pjbr-2013-0017>.
- [31] Bethia BT, Okamura AM, Kitagawa M, Fitton TP, Cattaneo SM, Gott VL, Baumgartner WA, Yuh DD. Application of haptic feedback to robotic surgery. *J Laparoendosc Adv Surg Tech A* 2004;14:191–5. <https://doi.org/10.1089/1092642041255441>.