Dental Assistant Robot for Surgical Procedures

Muhammad Kabir Hamzah, Diptan Regmi, Faisal Farhan

1. Introduction

Robots are devices designed to carry out automated physical tasks based on computer programming. Advancements in robotics and artificial intelligence have increasingly automated various tasks, especially those that are repetitive and time-consuming. Robotic automation and assistive technologies have a lot of potential in dentistry to raise the standard of care. Human resources can be freed up by robots to perform more important tasks, like interacting with patients or jobs requiring sophisticated cognitive skills. Dental professionals may experience physical and mental fatigue after performing lengthy and demanding procedures in physically uncomfortable positions, which may increase the likelihood of errors in oral examinations, disease diagnoses, and treatment planning. Daily routine tasks, such as cleaning instruments and surfaces in the dental clinic, also carry the risk of being performed with general negligence. Digital medicine and dentistry integrated with robotics can reduce errors while improving both the quality and efficiency of patient care. Over the years, researchers have developed numerous autonomous robots to aid the field of dentistry.

The objective of this project is to develop a dental assistant robot that can perform i) anesthesia and ii) suction for a dental patient using a Staubli robot. A wide variety of options are available to oral surgeons for preventing pain during procedures. Local anesthetics are used to numb the nerve endings in specific areas of the mouth, while general anesthesia can be administered to safely and comfortably keep the patient asleep throughout the surgery. These methods ensure that the patient remains comfortable as the procedure is carried out. Dental suction is also an important tool in oral surgery, utilized in various procedures such as filling teeth, draining abscesses, or performing complex operations like wisdom tooth extractions. Depending on the procedure, dentists and dental assistants employ different suction techniques to ensure effectiveness and precision. Automating these two tasks through a robot can help the dentist to work effectively and focus on more daunting tasks. Keeping all these things in mind we implemented these two functionalities in our dental assistant robot. The overall goal is to improve the efficiency of dental procedures, reduce surgery time, and provide consistent assistance to minimize dentist fatigue.

For this project we have used the Staubli TX60 robot manufactured by Staubli [ref]. It has a 6-axis robot arm that offers a 3.5kg payload and 600mm of reach. The Staubli TX60 robot offers a repeatability of ±0.02 mm and weighs approximately 51.4 kg. [ref: https://robodk.com/robot/Staubli/TX60]. We used the VAL 3 programming language to program our robot. For programming, simulation and monitoring our robot we used the Staubli Robotics suite.

With the Staubli Tx60 robot and an external syringe and suction tool we developed our dental assistant. A skeleton head is also used as the patient model. Based on the instructions of the user the robot can perform anesthesia or suction on the patient model. A interface is created in the robot to prompt the user for certain instructions such as choosing what procedures to be done and when to insert or remove a tool (syringe or suction). We also added some features such as flashing light indicators to indicate which buttons to press, progress bar and percentage indicators as well as a mechanical piston actuator that synchronizes with the progress bar and percentage indicators. Putting all these together completes the workflow of the dental assistant for both anesthesia injection and suctioning.

While the project successfully demonstrates these functionalities in a controlled environment, certain differences from real-life applications must be noted. The robot we use for simulation: Staubli TX60 is an industrial robot which lacks precision, safety, and adaptability to human biology as this robot is designed for automating high-speed, repetitive tasks and improving efficiency. The system was designed and tested on a static skeleton model, and actual patients would require dynamic adjustments based on real-time feedback. The use of a fixed syringe and suction tool also limits the system's adaptability compared to the varied tools used in dental practices. In simulation, the patient position/frame is fixed while in real applications, we need to map patient head anatomy and proceed with the application using real-time imaging. Here, we tested a static skeleton whereas in real life the robot must account for natural body movements like breathing or shifting of any position during the procedure. Error handling and safety features need to be added to cope with unexpected issues such as equipment malfunctions or robot malfunctions. These are required in a sterile environment with precise calibration. However, the project highlights the feasibility and potential of automating these essential dental procedures, offering a foundation for further development and real-world implementation.

2. Instruction Manual

2.1. Setting Up the Robot

Preparation Steps

i. **Prepare the Surgical Equipment**: Place all necessary tools (e.g., syringe, suction tools) within the surgical area for quick access.

ii. **Position the Patient**:

- Place the patient on the surgical bed, lying flat with their head facing upwards.
- Ensure the patient's mouth is open and accessible.

iii. **Power On the Robot**:

- Check that all connections are secure.
- Turn on the power button on the control panel of the robot.

2.2. Using the Control Panel

Control Panel Steps

1. **Load the Program**:
   - Access the interface and select the **Dental Assistance Program** from the menu.
2. **Run the Program**:
   - Once loaded, press the **Run** button to initialize the program.
3. **Activate the Robot**:
   - Press and hold the **Play** button. This activates the robot for operation.
4. **Initialization**:
   - The robot will automatically:
     - Check if it is in the **Home Position**. If not, it will move to the home position.
     - Move to the **Work Position** after completing the home position check.
5. **Task Selection**:
   - The robot will blink two buttons, prompting the surgeon to select the desired task:
     - **Button 1**: For **Anesthesia**.
     - **Button 2**: For **Suction**.

2.3. Surgery Procedures

A. Anesthesia Procedure

If **Button 1** is pressed, the robot will perform the following steps:

1. **Tool Attachment**:
   - The robot moves to the **tool position** (Gettool point) to allow the surgeon to attach the syringe.
   - The robot opens the syringe attachment point and displays a message: **"Attach the syringe and press Button 1."**
   - The surgeon should attach the syringe to the tool center point (TCP) and press the blinking/flashing **Button** to confirm the attachment.
2. **Move to Patient**:
   - After tool attachment, the robot moves to the **work area** and sequentially moves to:
     - **approxMouth**: An approximate location near the patient's mouth.
     - **transition**: A point midway into the patient's mouth.
     - **approxSkin**: Inside the mouth, touching the gum.

- **insideSkin**: Inside the gum for injection.

3. **Perform Injection**:

- The robot performs the injection with precision.
- A progress bar is displayed on the control panel, showing the status of the injection task.

4. **Return and Tool Detachment**:

- After completing the injection, the robot sequentially moves back through the following points:
  - **approxSkin → transition → approxMouth → tool position** (Gettool point).
- The surgeon should remove the syringe and press the blinking/flashing **Button** to confirm the detachment.

5. **Return to Home**:

- The robot moves back to the **work position** and then to the **home position**, ready for the next task.

B. Suction Procedure

If **Button 2** is pressed, the robot will perform the following steps:

1. **Tool Attachment**:

- The robot moves to the **tool position** (Gettool point) to allow the surgeon to attach the suction tool.
- The robot opens the suction attachment point and displays a message: **"Attach the suction tool and press the Button 1."**
- The surgeon should attach the suction tool to the TCP and press the blinking/flashing **Button** to confirm the attachment.

2. **Move to Patient**:

- After tool attachment, the robot moves to the **work area** and sequentially moves to:
  - **approxMouth**: An approximate location near the patient's mouth.
  - **approxMouthSuc**: A midway point into the patient's mouth.
  - **inMouthSuc**: The position for suctioning.

3. **Perform Suction**:

- The robot performs suction to clean the area.
- A progress bar is displayed on the control panel, showing the status of the suction task.

4. **Return and Tool Detachment**:

- After completing the suction task, the robot sequentially moves back through the following points:
  - **inMouthSuc → approxMouthSuc → approxMouth → tool position** (Gettool point).
- The surgeon should remove the suction tool and press the blinking/flashing **Button** to confirm the detachment.

5. **Return to Home**:

- The robot moves back to the **work position** and then to the **home position**, ready for the next task.

2.4. Completion of Procedure

- Once the procedure is completed, the robot will display a message: **“Program Finished”.**
- The surgeon can rerun the program to perform additional tasks as needed.

2.5. Important Notes

Speed Settings

- **Fast**: Used for non-critical movements (e.g., moving between home and work positions, getting tools).
- **Slow**: Used for precise movements near the patient to ensure safety and accuracy.

Attachments

- Ensure the correct tool (syringe or suction) is attached securely before starting the procedure.

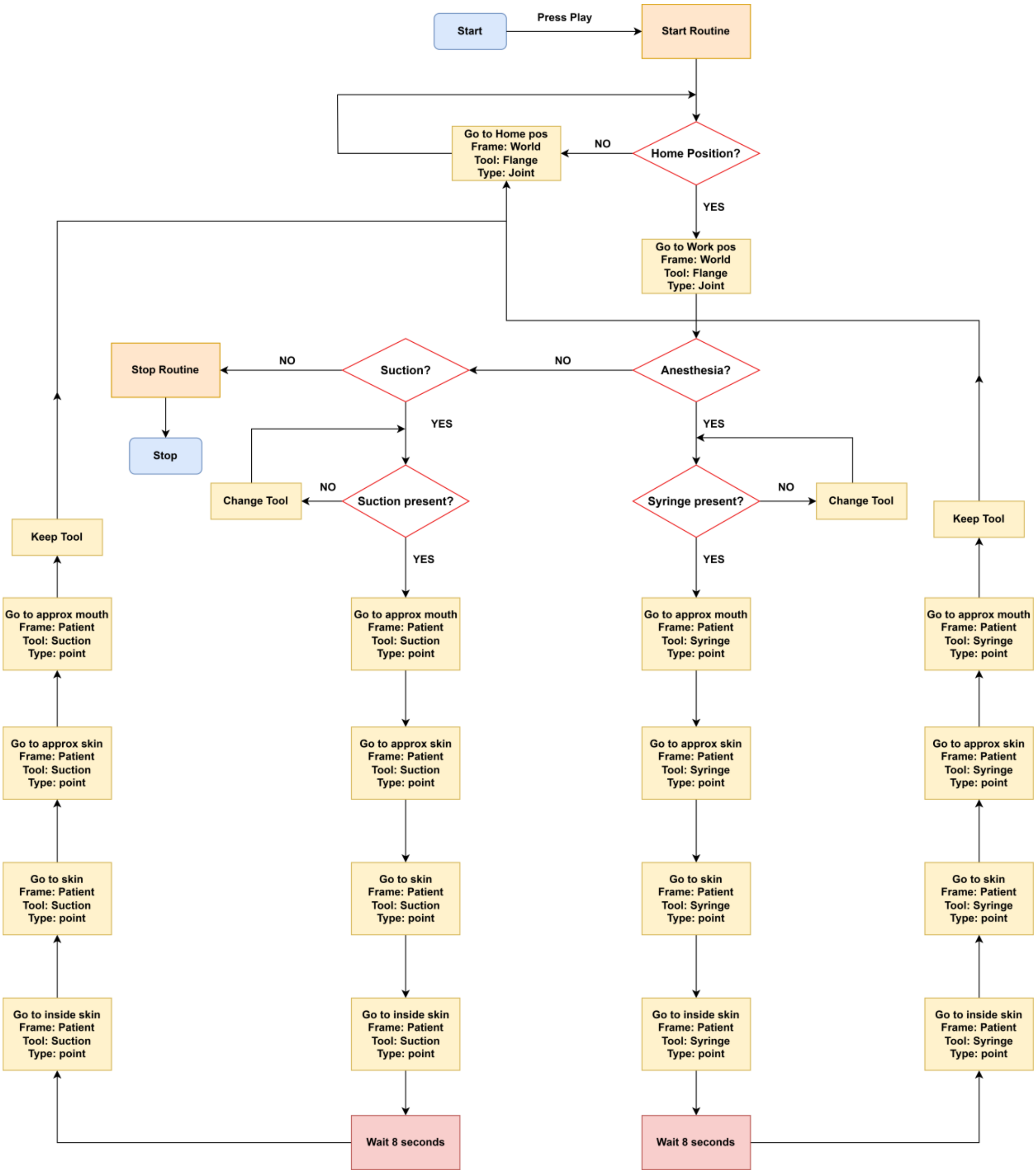
Safety Precautions

1. Always supervise the robot while it is in operation.
2. Maintain a safe distance from the robot when it is moving.
3. Follow all on-screen instructions on the control interface.

Safety Features

- **Indicators**: Flashing lights guide the surgeon during tool attachment and detachment.
- **Manual Override**: The robot can be stopped at any time by pressing the **Stop** button.

3. Block Diagram



4. Frame Definition

First, we define (by recording) home and work positions where all joints (j1 -j6) are set to [0, 0, 0, 0 ,0] and [-31.51138, 11.84158, 70.38717, 0.24327, 100.53681, 0.06167] for **Home[0] and Work[0]** respectively. The figures below show the robot in both home and work positions.

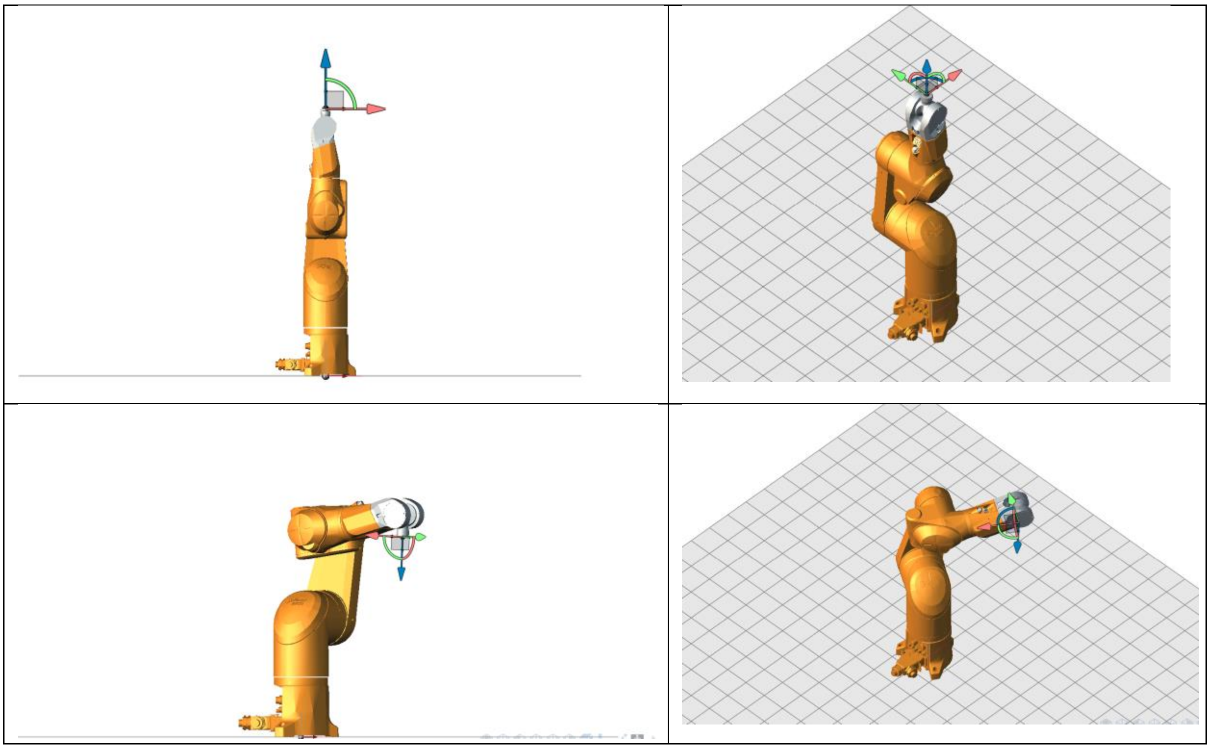


Fig.1. Robot at Home (top) and work (bottom) positions simulated in the staubli robotics suite.

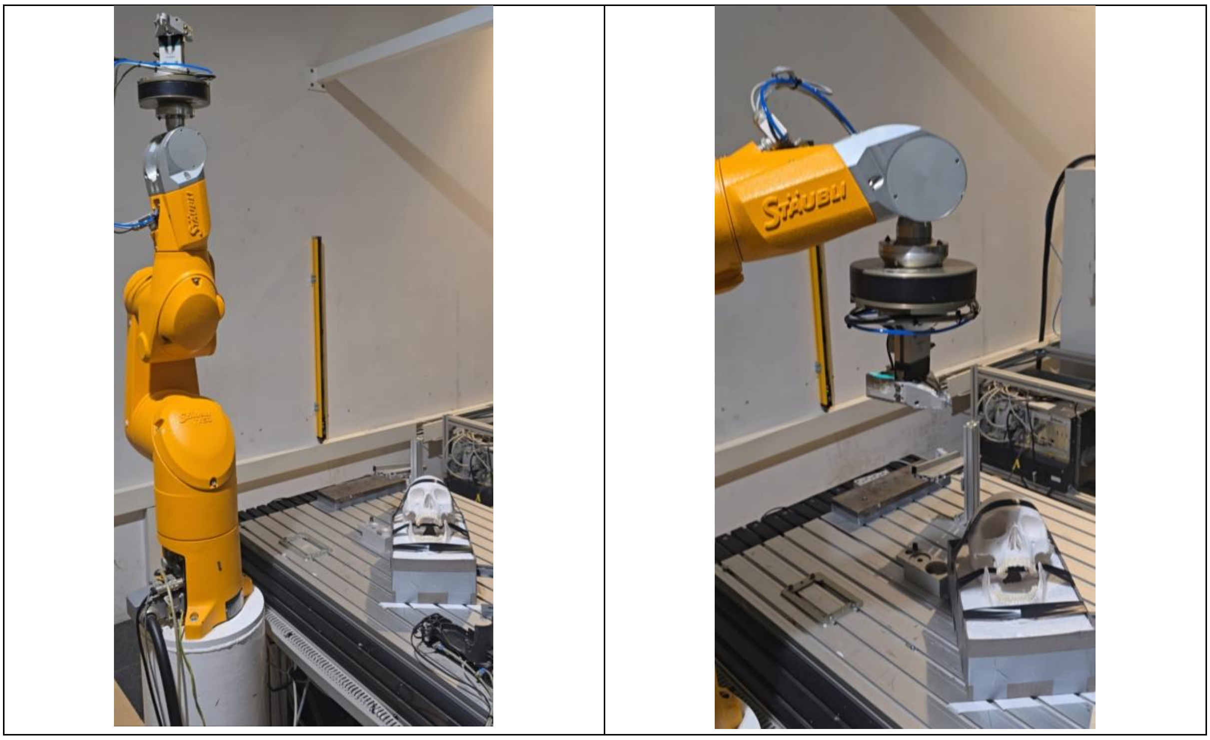


Fig.2. Actual robot at Home (left) and Work (right) Positions captured in the lab.

We also defined a patient frame (**patient[0]**) representing the frame with which all subsequent positions were recorded.

FrameName	X	Y	Z	Rx	Ry	Rz
patient[0]	477.28683	-197.60421	-268.00045	-23.85826	27.3008	56.49083

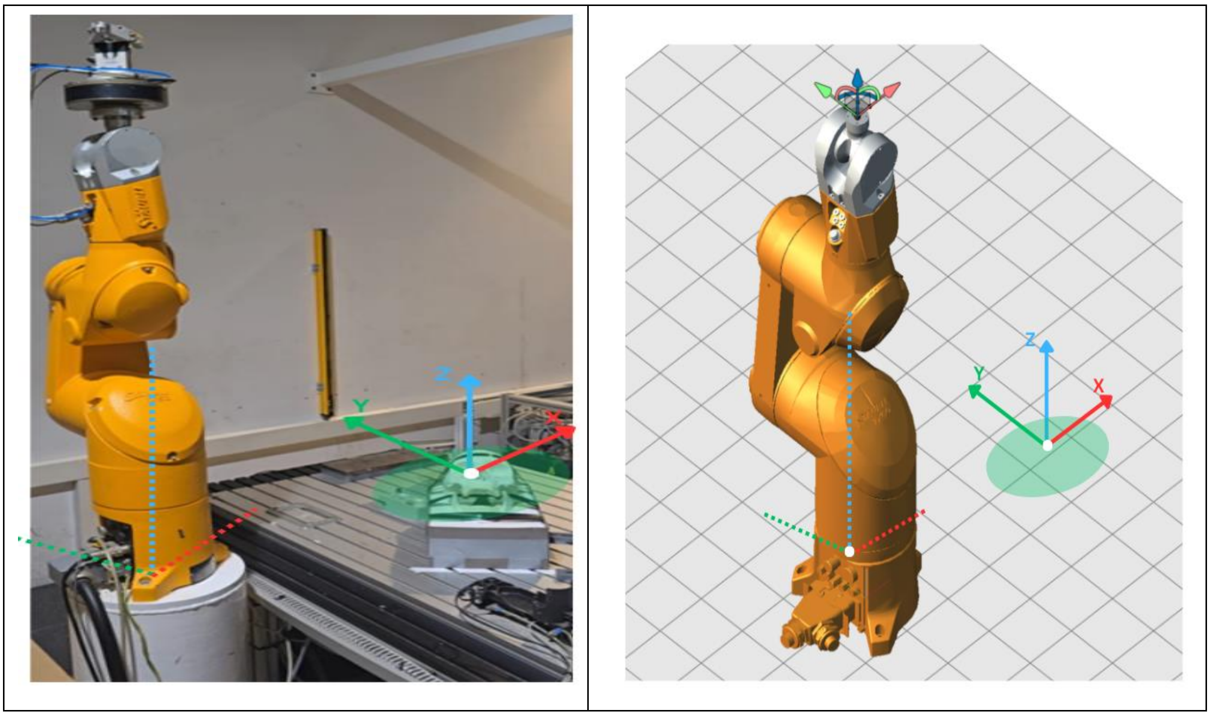


Fig.3. Definition of the patient frame. On the left side, the frame is overlaid on the actual robot, and on the right side, the frame and robot are simulated using the staubli robotics suite. The reference frame (Base frame) is illustrated with dashed lines in both images.

5. Tool Definition

The tools are defined as **syringe[0] and suction[0]**. Both tools as the names imply represent the tool used for either injecting anesthesia or suctioning out liquids from the mouth. We define both tools by measuring their length from the TCP and recording the precise coordinates with respect to the patient frame.

Tool Name	X	Y	Z	Rx	Ry	Rz
Syringe[0]	-30	-30	353	0	0	-134
Suction[0]	-30	-30	353	0	0	-134

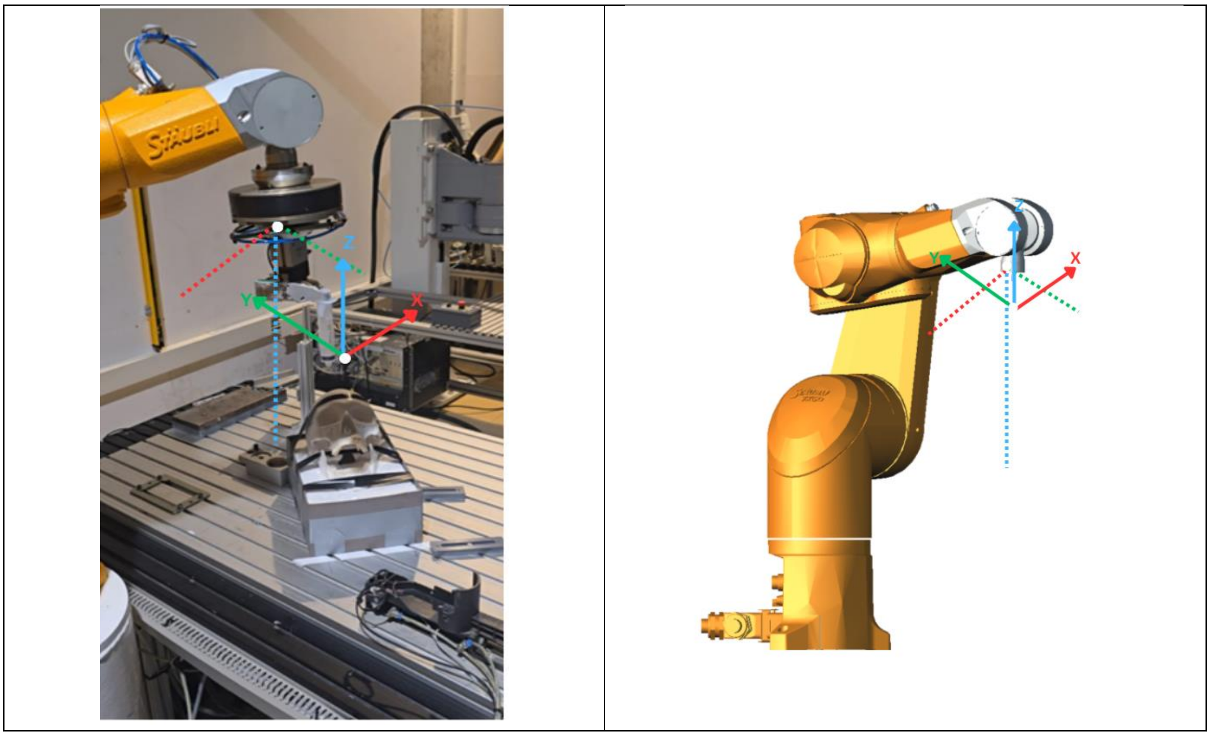


Fig.4. Definition of the Tool Center Point (TCP) frame. On the left side, the frame is overlaid on the actual robot and on the right side, the frame and robot are simulated using staubli robotics suite. The reference frame (Flange frame) is Illustrated with dashed lines in both images.

6. Variables

We recorded the point positions with each tool (syringe or suction) and with respect to the patient frame as well as custom speeds (fast and slow) for movement before and within the patient area. The variables recorded are as follows:

- **approxMouth**: Represents the point where the robot moves from the work area to an approximate location of the patient's mouth during both procedures.
- **transition**: Represents a transition point where the robot moves from the approxMouth position to a point midway into the patient's mouth during both procedures.
- **approxSkin**: Represents the point where the robot moves from the transition to an approximate location inside the patient's mouth touching the gum during the anesthesia procedure.
- **insideSkin**: Represents the point where the robot moves from the approxSkin into the gum and performs anesthesia injection.
- **approxMouthSuc**: Represents the point where the robot moves from the transition position midway into the patient's mouth during the suctioning procedure.
- **inMouthSuc**: Represents the point where the robot moves from the approxMouthSuc position to the point where it performs the suctioning procedure.
- **Gettool**: Represents the point where the robot arm moves to for the user to place the tools. It is located near the patient for convenience.
- And lastly four calibration points **cal[0-3].**

Variable Name	Type of variable	Parent	Default value
approxMouth	point	Patient[0]	(X, Y, Z: -2.33498, -9.06048, 15.57342) (Rx, Ry, Rz: -14.53001, 104.76999, -13.43998)
transition	point	Patient[0]	(X, Y, Z: 40.43454, -11.61247, 5.33501) (Rx, Ry, Rz: -14.52672, 104.76993, -13.443)
approxSkin	point	Patient[0]	(X, Y, Z: 47.25961, 15.29966, 5.6005) (Rx, Ry, Rz: 34.71573, 115.21271, -62.14874)
insideSkin	point	Patient[0]	(X, Y, Z: 57.89085, 18.16378, 1.48457) (Rx, Ry, Rz: 34.72284, 115.21515, -62.15506)
approxMouthSuc	point	Patient[0]	(X, Y, Z: 78.80726, -12.78095, 0.2459) (Rx, Ry, Rz: -12.5999, 104.36284, -13.68567)
inMouthSuc	point	Patient[0]	(X, Y, Z: 124.22832, -10.3642, -7.6128) (Rx, Ry, Rz: -13.68808, 104.07928, -12.62487)
Gettool	point	Patient[0]	(X, Y, Z: -261.66715, -334.54385, 71.47145) (Rx, Ry, Rz: -51.01638, 156.2786, -17.80273)

7. Code

Start()

```
begin
 // accesssory definition
 dioLink(button[0],io:bIn4)
 dioLink(button[1],io:bIn5)
 dioLink(button[2],io:bIn6)
 dioLink(button[3],io:bIn7)
 dioLink(indicator[0],io:bOut4)
 dioLink(indicator[1],io:bOut5)
 dioLink(indicator[2],io:bOut6)
 dioLink(indicator[3],io:bOut7)
 dioLink(piston_actuator[0],io:bOut10)
 dioLink(piston_sensor[0],io:bOut14)
 dioLink(piston_sensor[1],io:bOut15)

 userPage()
 cls()
 title("Dental Program")
 enablePower()
```

```
putln("Moving to Home")
movej(Home,flange,fast)
waitEndMove()
open(syringe)
delay(1)
putln("Arm at Home!")
putln("Moving to Work Area!")
movej(Work,flange,fast)
waitEndMove()
putln("Arm at Work Area!")
putln("Press Button")
putln("1 = Anesthesia and 2 = Suction")

// flash indicators prompting the buttons to press
counter=0
do
 delay(0.1)
 counter=counter+1
 if(counter<5)
  indicator[0]=true
  indicator[1]=false
 elseIf(counter>5)
  indicator[0]=false
  indicator[1]=true
 endIf
 if(counter==10)
  counter=0
 endIf
until(button[0]==true or button[1]==true)

// based on the user selection,
// run one of the programs (suction or Anesthesia)
if(button[0]==true)
 indicator[0]=false
 indicator[1]=false
```

```
  cls()
  putln("Anesthesia!!")
  putln("Moving to Place Tool")
  delay(2)
  call programAnes()
 elseIf(button[1]==true)
  indicator[0]=false
  indicator[1]=false
  cls()
  putln("Suction!!")
  putln("Moving to Place Tool")
  delay(2)
  call programSuc()
 endIf
end
```

Stop ()

```
begin
 movej(Home,syringe[0],fast)
 waitEndMove()
 putln("Application Stopped")
 popUpMsg("Program Finished")
end
```

programAnes ()

```
begin
 // change tool
 movel(gettool,syringe[0],slow)
 waitEndMove()
 putln("Attach Syringe and Press Button")
 piston_actuator[0]=false
 // Blink indicators until a button is clicked
 counter=0
 do
  delay(0.1)
  counter=counter+1
  if (counter<5)
   indicator[0]=true
  elseIf (counter>5)
   indicator[0]=false
  endIf
  if (counter==10)
   counter=0
  endIf
 until (button[0]==true)
 indicator[0]=false
 // main operation
 close(syringe)
 delay(1)
 putln("Moving back to Work")
 movej(Work,flange,slow)
 waitEndMove()
 putln("Moving to Patient!!")
 movel(approxMouth,syringe[0],slow)
 movel(transition,syringe[0],slow)
 movel(approxSkin,syringe[0],slow)
 movel(insideSkin,syringe[0],slow)
 waitEndMove()
 cls()
```

```
putln("Injecting ......")
piston_actuator[0]=true
counter=0
do
 gotoxy(15+counter,6)
 delay(0.5)
 putln("I")
 counter=counter+1
 gotoxy(17,7)
 put(20+counter*10)
 putln("%")
until (counter==8)
piston_actuator[0]=false
gotoxy(0,10)
putln("Finished Injecting!")
movel(approxSkin,syringe[0],slow)
movel(transition,syringe[0],slow)
movel(approxMouth,syringe[0],slow)
movej(Work,syringe[0],fast)
putln("Moving to Keep Tool")
movel(gettool,syringe[0],slow)
// keep tool
open(syringe)
putln("Remove Syringe and Press Button")
// Blink indicators until a button is clicked
counter=0
do
 delay(0.1)
 counter=counter+1
 if (counter<5)
  indicator[1]=true
 elseIf (counter>5)
  indicator[1]=false
 endIf
 if (counter==10)
```

```
    counter=0
  endIf
 until (button[1]==true)
 indicator[1]=false
 close(syringe)
 delay(1)
 putln("Syringe tool successfully returned.")
 movej(Work,syringe[0],slow)
end
```

programSuc()

```
begin
 // change tool
 movel(gettool,suction[0],slow)
 waitEndMove()
 putln("Attach Suction and Press Button")
 piston_actuator[0]=false
 // Blink indicators until a button is clicked
 counter=0
 do
  delay(0.1)
  counter=counter+1
  if (counter<5)
    indicator[0]=true
  elseIf (counter>5)
    indicator[0]=false
  endIf
  if (counter==10)
    counter=0
  endIf
 until (button[0]==true)
 indicator[0]=false
 // main operation
```

```
close(suction)
delay(1)
putln("Moving back to Work")
movej(Work,flange,slow)
waitEndMove()
putln("Moving to Patient!!")
movel(approxMouth,suction[0],slow)
movel(approxMouthSuc,suction[0],slow)
movel(inMouthSuc,suction[0],slow)
waitEndMove()
cls()
putln("Cleaning .....")
piston_actuator[0]=true
counter=0
do
 gotoxy(15+counter,6)
 delay(0.5)
 putln("I")
 counter=counter+1
 gotoxy(17,7)
 put(20+counter*10)
 putln("%")
until (counter==8)
piston_actuator[0]=false
gotoxy(0,10)
putln("Finished Suctioning!")
movel(approxMouthSuc,suction[0],slow)
movel(approxMouth,suction[0],slow)
movej(Work,suction[0],fast)
putln("Moving to Keep Tool")
movel(gettool,suction[0],slow)
// keep tool
open(suction)
putln("Remove Suction and Press Button")
// Blink indicators until a button is clicked
```

```
 counter=0
 do
  delay(0.1)
  counter=counter+1
  if (counter<5)
   indicator[1]=true
  elseIf (counter>5)
   indicator[1]=false
  endIf
  if (counter==10)
   counter=0
  endIf
 until (button[1]==true)
 indicator[1]=false
 close(suction)
 delay(1)
 putln("Suction tool successfully returned.")
 movej(Work,suction[0],slow)
end
```