Bilateral Control Robotic Arm

Mostafa Elmasry
ESM, FH Dortmund
Dortmund, Germany
7207073

Philip Okonkwo
ESM, FH Dortmund
Dortmund, Germany
7207097

Swathi Sudeendra Rao
ESM, FH Dortmund
Dortmund, Germany
7207100

Santhosh Kumar Ramareddy
ESM, FH Dortmund
Dortmund, Germany
7206929

Elis Gjana
ESM, FH Dortmund
Dortmund, Germany
7203373

***Abstract* — This paper presents the utilization of the control theory approach "Bilateral control" for robot manipulation in medical field specifically in surgical operations. This application like any other in these modern technological days will rely on an embedded system for implementation. In order to deliver such complex system accurately and efficiently, we had to model and design the system to know all the requirements, make the system more understandable and ensure maintainability.**

Keywords—Control, robotics, bilateral control, medical robots

I. Introduction

An embedded system is a combination of a computer processor, computer memory, and input/output peripheral devices that has a dedicated function within a larger mechanical or electrical system. It is embedded as part of a complete device often including electrical or electronic hardware and mechanical parts. Because an embedded system typically controls physical operations of the machine that it is embedded within, it often has real-time computing constraints. Embedded systems control many devices in common use today.[1] Ninety-eight percent of all microprocessors manufactured are used in embedded systems.[2]

A. Embedded and robotics

Almost every device that we use in our day to day life depends on embedded systems. Same goes to robotics. Controlling a robotic system can be described in two parts, a part uses a modeling process of the mechanical features of the robot and the other part uses an embedded system as a brain of the operation to deliver the desired functionality; actuating the links, utility of sensors or monitoring performance. With the enhancement of the embedded field also comes the significant increase of speed, efficiency and accuracy of the designed robot.

B. Bilateral Control

Bilateral control is a remote-control approach which is used to perform control and manipulation from a far distance[3] Bilateral control system has been developed as a method to transmit a tactile sense. It enables human operators to feel as if they were really touching the environment.[4] This technique is used in several applications like trajectory tracking, compliance control, interaction force control are scientific to maintain desired system configuration. [5]

II. Methodology

Nowadays robotic systems are used in various fields as a substitution for human force. The reason behind it is to generate the best possible output from a system by increasing efficiency and decreasing the cost and human risk. As an example, industrial robots perform several processes faster, more accurate and concurrent. They even eliminate the risk of fatality. Another use of robotic system that is our case, the medical field. In surgical operations, a major constriction come in the situation, which is the geographical location of the patient and his doctor.

Surgical robotic systems use bilateral control to make enable expert doctors perform critical surgeries and operations that are in their area of knowledge, this way an increase in the possibility of the success of the operations is guaranteed.

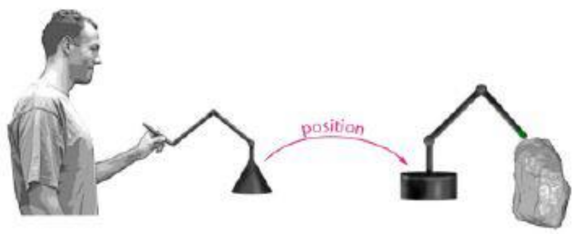


Figure 1 Bilateral control of a robotic arm

The figure above is a representation of the implementation of bilateral control on a robotic arm. The user moves a robot "Master robot" as he desires and this movement is read as data in respect of coordinates, forces and angles to be transmitted to the required robot "Slave robot". The role of the system is to ensure that the slave robot mimics efficiently the movement of the master robot. This situation can be translated in the surgical operations where the doctor is the user of the master robot and the slave robot operates on the patient. Also, the slave robot sends it's interaction with the environment to make the doctor feel the surrounding objects as if he is in the operation's location. This is to prevent any accidents while the operation is in place.

III. System characteristics

A. Reactiveness

a. The Robot is continuously reactive with the environment and forces.

b. It does not terminate unless the user exits the system.

c. Stimulus/response as it senses drag forces and stiffness from the environment and sends the feedback to the master robot thus the user.

B. Hard Real-Time

Hard real-time system as the robot should immediately synchronize with the users movement with the least minimal delay

C. Continuous

In order to control the robot, differential equations and continuous time functions are used

D. Embedded System

a. Sensors are used for the master robot to measure the required coordinates and angles to send them to the system controlling the slave robot (End Effector).
b. Sensors used for the slave robot to sense and react with the environment and send feedback to the master robot.
c. Both sensors read analogue measurements and ADC are used for computation.
d. Servo Motors with good torque are used.
e. Micro-processors for controlling the motors.

IV. USE CASE DIAGRAM

Use case modeling is the view of the user to the system's behavior. A use case model distributes system functionality into transactions ('use cases') that are meaningful to users ('actors').

Figure 2 Use Case Diagram - Bilateral Robot Control

The above figure represents our Use case diagram of the system, where the only two actors are the Doctor who is the main actor and user and the Hospital as the reactor.

The use cases represents the user's interaction with the system and how he affects it. By moving the arm with the desired movement, use cases are constructed for a better understanding. In our system, the doctor has the following roles: Verification on the system, manipulating the arm and monitoring using the camera and GUI. The hospital's affection comes only with the verification in that case.

V. BLOCK DIAGRAM

Blocks are used to specify hierarchies and interconnection. Block definition diagrams describe the relationship among various blocks (e.g., composition, association, specialization). Internal block diagrams describe the internal structure of blocks in terms of properties and connectors. Behavior can be allocated to blocks.

Block Definition Diagram(BDD) are used to define the features of a block and any form of relationships between blocks such as associations, generalizations, and dependencies, in terms of properties, operations, and relationships. Block Definition Diagrams are based on UML class diagrams and include restrictions and extensions as defined by SysML.

The model elements that you display on BDDs—blocks, actors, value types, constraint blocks, flow specifications, and interfaces—serve as types for the other model elements that appear on the other eight kinds of SysML diagrams. Elements that appear on BDDs are referred to as elements of definition. Elements of definition, in a real sense, form the foundation for everything else in the system model. [7]

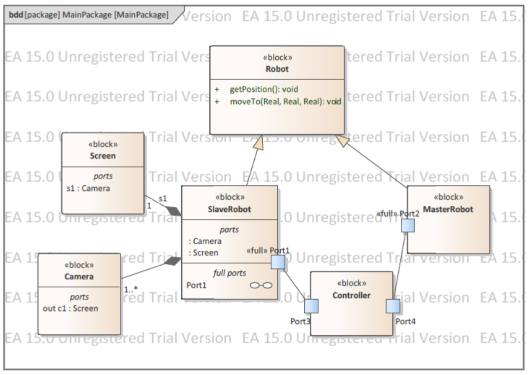


Figure 3 Block Diagram

The block diagram in the figure shows how the system should work. It contains about 6 blocks. The MasterRobot and SlaveRobot block inherits from the Robot block. The screen and Camera blocks are part of the SlaveRobot block since they will aid the doctor or operator of the system to be able to discern what is happening on the slave end of the robot. A controller transfers what happens on the master robot to the slave robot.

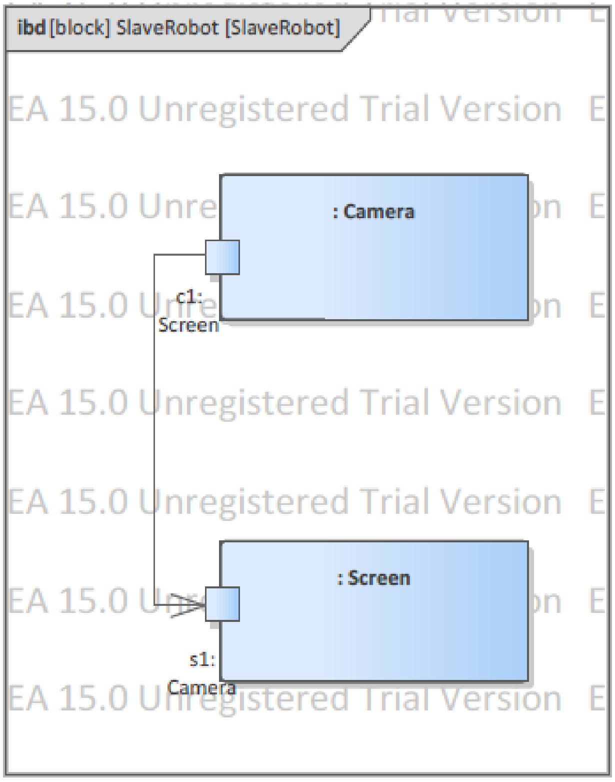


Figure 4 Internal Block Diagram - SlaveRobot

The internal block diagram or "ibd" resembles a traditional system block diagram and shows the connections between parts of a block [8]. As can be seen from the diagram, signal from the Camera is sent to Screen. This is what helps the operator of the master robot see whats going on at the slave robot end.

VI. SEQUENCE DIAGRAM

The Sequence diagram in gerneral way, shows the order in which the objects or classes of a program communicate with each other as time passes by. The events are enlisted in the order or sequence with which they occur. The flow of data and the flow of control are the major ideologies that are depicted through Sequence diagram.

This topic has been entitled to throw light upon the sequence of events that occur when the Control System ought to check for the movements of Master and Slave robots and make sure that their actions synchronize amongst the pair.

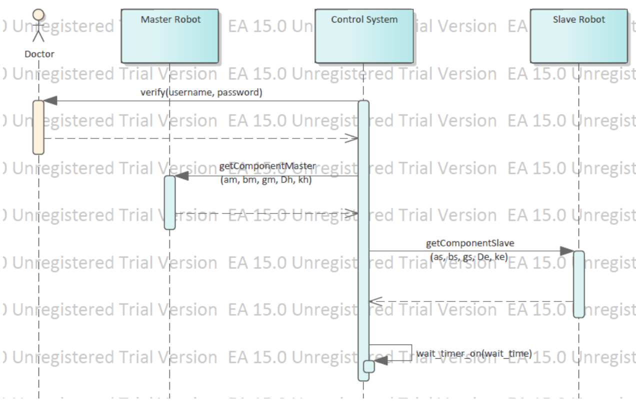


Figure 5 Sequence Diagram-Part 1

The Control System first authenticates for the doctor's credential as shown in Fig., 1. The username and password are passed as arguments. Once the data is authenticated, the component data (Inertia matrix a, Nonlinear frictional force coefficient b, gravitational force coefficient g, Damping coefficient D, stiffness k) of master robot and then the slave robot, are fetched. These components remain almost constant throughout the operation, once the sequence of events start. Hence, these are fetched in hand.

Once these data are fetched, the timer is started and the motion of the master robot is awaited. The events that occur following the start of timer are described by Fig., 2.

There are two ways to continue at this point of time:

A. The loop of continuously calculating the torques, sending them to the other bot and fetching live camera updates happen only if the master robot does not remain idle for more than 60 seconds.

B. Once the master robot is idle, the alternative case is used and the next set of events take place.

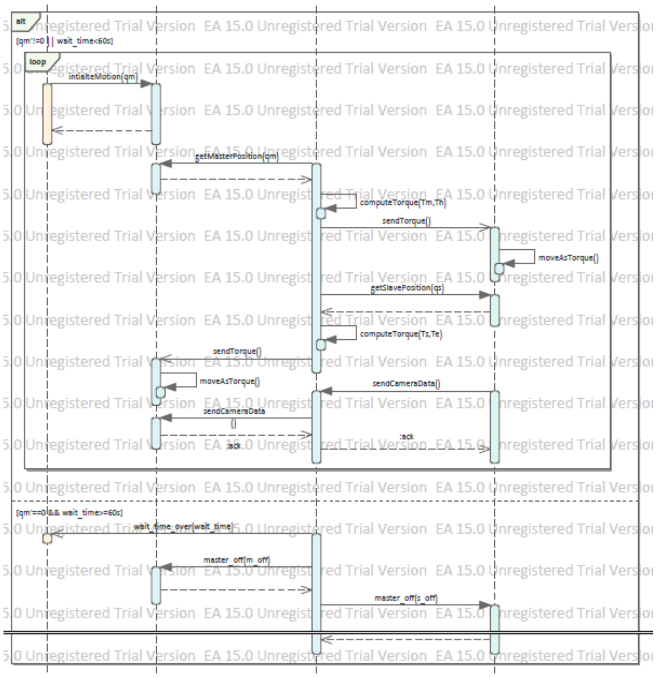


Figure 6 Sequence Diagram-Part 2

The count on timer is used to decide which of the parts are executed as shown in Fig., 6

A. The first part of the diagram is executed if the motion q is detected at master robot's end by the control. Once the motion is detected, the torque is calculated by the control system using Eq., (1) in the following set of equations:

$$D_h\,(q_m{'}-q_h{'}) + k_h\,(q_m-q_h) = \tau_h \tag{1}$$

$$D_e\,(q_s{'}-q_e{'}) + k_e\,(q_s-q_e) = \tau_e \tag{2}$$

$$a_m\,(q_m)\,q_m{''} + b_m\,(q_m\,,q_m{'}) + g_m\,(q_m) = \tau_m + \tau_h \tag{3}$$

$$a_s\,(q_s)\,q_s{''} + b_s\,(q_s\,,q_s{'}) + g_s\,(q_s) = \tau_s + \tau_e \tag{4}$$

The τ_h and τ_m are computed and sent to the slave robot for its motion as per this computed torque. τ_h and τ_m are External torque applied by Operator hand to move master robot to position q_m and torque applied by motors of the master to hold its structure, respectively.

Once the slave robot moves as per the computed torque, the reactive motion of the slave robot is sent to the control system. It again computes the τ_s and τ_e, by using eq., (2) to send it to the mater robot. τ_s and τ_e are torque applied by motors of the slave to hold its structure and resisting torque from the environment resisting motion of the slave robot to desired position q_s, respectively. Thus, the computed torque at the end of slave robot is sent to the master robot by sending τ_s and τ_e.

This happens until and unless the motion is being detected on master's side and the master is not idle for more than a minute (60 seconds)

Also, in the same loop, the camera data is continuously sent as the live update of what motion the slave robot is making after each and every motion by the master robot.

B. If the master bot is found to be idle for more than 60 seconds, the doctor is notified the same. Bothe bots are to be automatically switched off when there is no motion detected for more than a minute.

Also, the live data by the camera is no more sent to the master robot.

VII. Activity Diagram

Illustrate the flow of control in a system. There are different ways of control flows including parallel, concurrent, single and branched. They are intended to model both computational and organizational processes starting from the initial state till the final state.

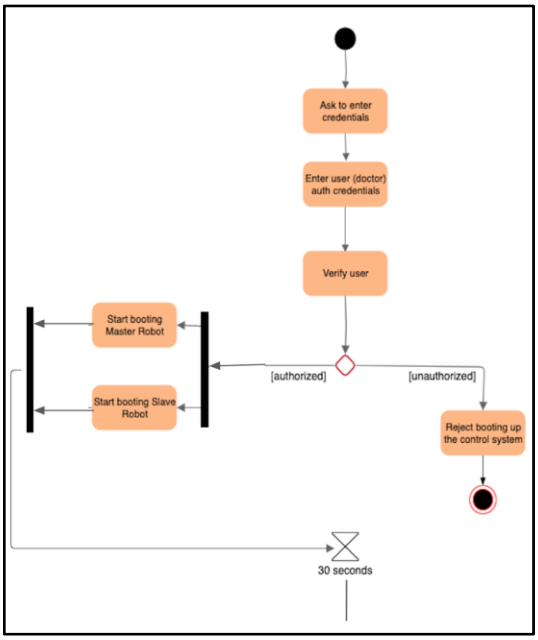


Figure 7 Authentication Activity Diagram

In the activity diagram above it is described the authentication part of the Bilateral Robotic System in which, only authorized people (specialized doctors), can log in and use the system. The system starts by asking the user (doctor) to enter his credentials and after the input data, it verifies if the credentials are valid or not. If the credentials are verified and the user has the "authorized" status, both Master and Slave Robots start booting at the same time parallely. After that, the system waits 30 seconds to continue operating in normal conditions. On the other hand, if the user is unauthorized, the system will reject booting up the control system.

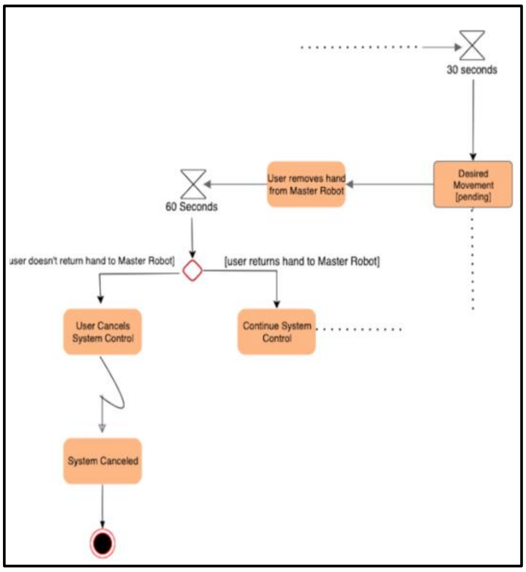


Figure 8 User stops using the system

After the 30 second waiting time, the desired movement remains in the pending mode and the user has two options: to continue the operation, or to take a break and stop using the system. The second situation is described in the activity diagram in Figure 2, in which the user removes the hand from the master robot and this way, the system waits in an idle state for a reaction from the user. If the user doesn't return his hand in the master robot in 60 seconds, the system control is going to be cancelled. Otherwise, if the user returns his hand back to the robot, it will continue the system control.

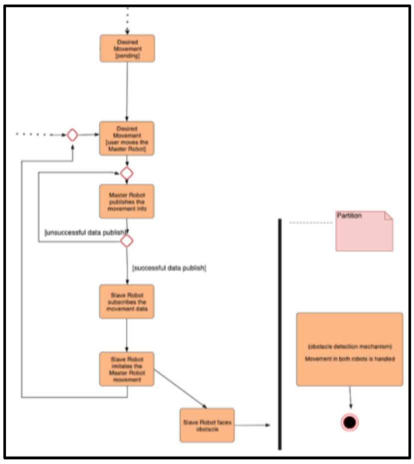


Figure 9 User continues using the system

In Figure 3 it is displayed the other part of the activity diagram, in the scenario the user decides to continue with the system operation by moving the master robot. This way, the master robot publishes its movement data including the coordinates and angle. If the data is published successfully, the slave robot will subscribe data in order to imitate the movement of the master robot. If the data is not published successfully, the master robot has to republish the data again. In the special cases where the slave robot faces some obstacles (bones), an obstacle detection mechanism will handle the movement of both robots. All the time the system is operating, a periodic system update will be executed every 0.1 seconds, in which the system gets the robot's coordinates and angles, and saves these data in the database for further synchronisation. This action is described in the activity diagram below:

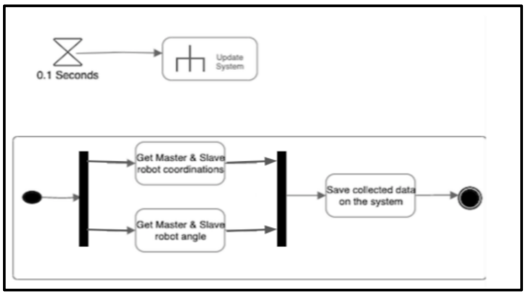


Figure 10 . Update System Activity Diagram

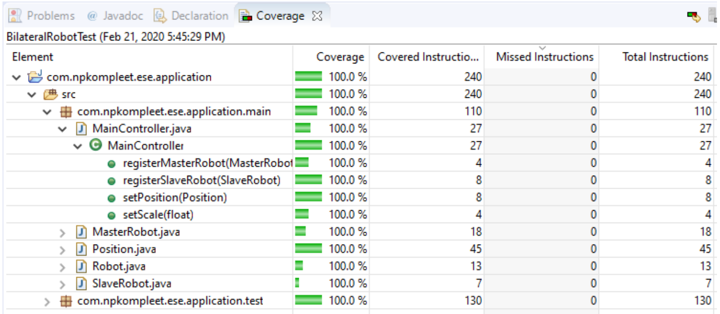
VIII. Verification and Validation

Test-driven development is a development practice that increases developer confidence in the written code by advocating tests for all software requirements. It makes developers to work in short incremental development cycles, thus providing quick feedback on our progress. TDD involves writing a failing test before writing the production code. The complete process is as follows:

- Create a test for the new functionality or method.
- Run the test and see it fail.
- Write the code to make the test pass.
- Make sure all the previous tests pass as well.
- Repeat until code is completed

A good use of TDD is to avoid regression errors. "The most obvious reason to follow TDD is to detect regression errors in an automated manner. Teams that follow TDD have the freedom to change and add new features without worrying about introducing regressions. If all the previous test cases ran fine, then we can be sure that we have not introduced any regression error" [3]

The prototype implementation for the project was implemented in Java programming language. It was developed using the TDD principle and the unit tests were also written with JUnit.



Problems | Javadoc | Declaration | Coverage

BilateralRobotTest (Feb 21, 2020 5:45:29 PM)

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
com.npkompleet.ese.application	100.0 %	240	0	240
src	100.0 %	240	0	240
com.npkompleet.ese.application.main	100.0 %	110	0	110
MainController.java	100.0 %	27	0	27
MainController	100.0 %	27	0	27
registerMasterRobot(MasterRobot	100.0 %	4	0	4
registerSlaveRobot(SlaveRobot)	100.0 %	8	0	8
setPosition(Position)	100.0 %	8	0	8
setScale(float)	100.0 %	4	0	4
MasterRobot.java	100.0 %	18	0	18
Position.java	100.0 %	45	0	45
Robot.java	100.0 %	13	0	13
SlaveRobot.java	100.0 %	7	0	7
com.npkompleet.ese.application.test	100.0 %	130	0	130

Figure 11. Code coverage

IX. State Machine Diagram

Shows discrete behavior of a part of designed system through finite state transitions. Typically they are used to describe state-dependent behavior for an object. In Figure 5 it is described the authentication state machine diagram, in which the system begins as unauthenticated and if the credentials entered by the user are correct, then the state changes to authenticated. On the other side, the state can change from authenticated to unauthenticated, if the idle time of waiting is greater than 60 seconds, or the user decides to logout and abandon the system.

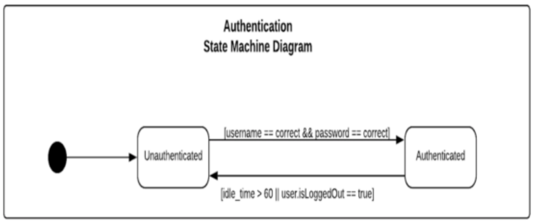


Figure 12 . Authentication State Machine Diagram

One important state machine diagram, is the one in Figure 6 in which describes the three states of the Bilateral Robotic System which are:

- State ON
- State IDLE
- State OFF

The state can change easily from ON to IDLE if the user removes the hand from the master robot and can change back to ON, if the hand is returned to the robot. If the user does not return his hand to the master robot in 60 seconds, the state will automatically change to OFF. Using the "Power On" switch, the user can change the state from OFF to ON, and also using the corresponding “Power Off” switch performs the inverse operation. The picture below describes the state transitions:

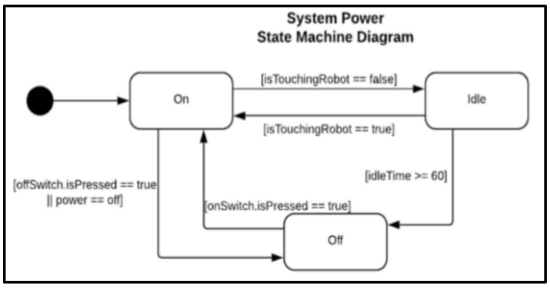


Figure 13 System State Machine

Another important functionality of the system involves the cameras implemented in the slave robot, using which, the doctor performs the entire surgery. These cameras have only two states: ON and OFF and the condition that makes it possible to switch from one state to another, is whether or not the system is idle. If the system is idle, the cameras automatically change the state from ON to OFF. On the other side, if the system is not idle anymore, the system switches back to the initial ON state (Figure 7).

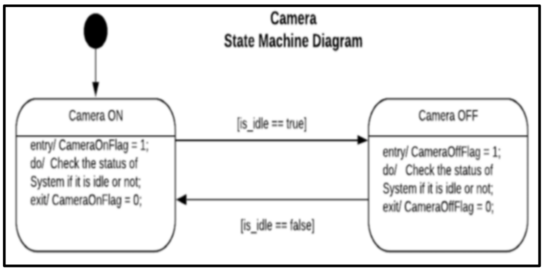


Figure 14 Camera State Machine Diagram

X. Requirements Diagram

A requirement is a need or a capability that should be provided by the system. There are several types of requirement, quality, functional or safety requirements. These requirements are documented as a guideline to ensure that the system will have the specifications that meets the customer/market needs, guaranteeing effectiveness and preventing extra cost.

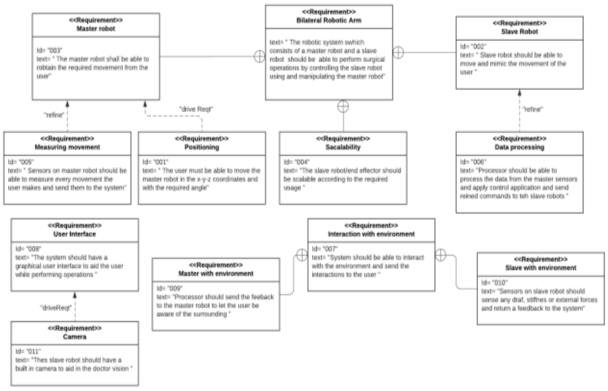


Figure 15Requirement Diagram for Bilateral Control Robotic System

The figure above represents the complete functional requirements of the system that enables the product to perform its functionality efficiently. The major requirement of the system is that the system should be able perform the surgical operations from distance using the bilateral control. Then this major requirement is divided into smaller and much more specific requirements.

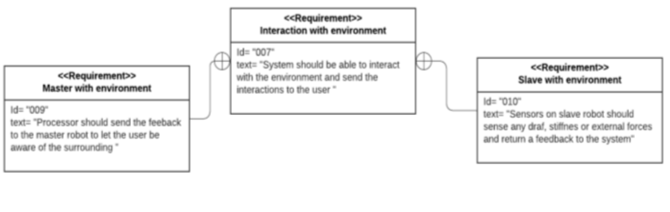


Figure 16 Requirement for environment interaction

A major important part of the system is the environment interaction that takes place when the slave robot senses external forces due to stiffness, weights or drags. The system in that case should transmit these senses to the master robot and thus to the doctor.

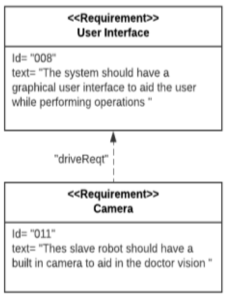


Figure 17 Requirements for User interface with the system

XI. Constraint Diagram

A constraint diagram will contain the hurdles of a system that would affect it by physical means. It is a better approach to enhance the understanding of individual components that affect the flawless and effective working of a system.

A. The constraint diagram in fig., 3 has the mathematical formula that describes the hurdle of the torque affecting the master’s torque due to the External torque applied by Operator hand. As it is a torque that is the effect of the motion of the master robot, is thus another input to compute effective torque produced by the master robot.

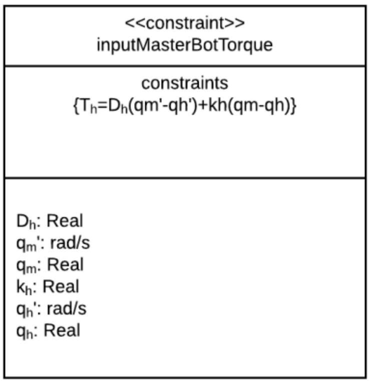


Figure 18 Constraint Diagram 1

B. The constraint diagram in fig., 4 has the mathematical formula that describes the hurdle of the total torque affected due to the resisting torque from the environment. Similar to the constrain affecting the master robot, constrain affecting the slave robot, the torque thus obtained is another input to compute effective torque produced by the slave robot.

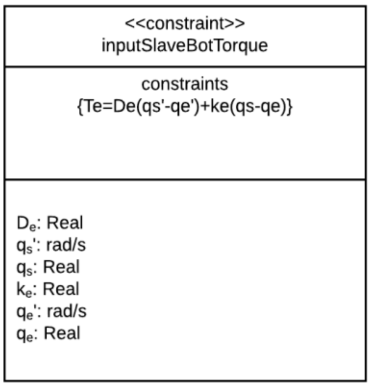


Figure 19 . Constraint Diagram 2

Various parameters of the system are described in the third partition of the system in fig., 18 and 19.

XII. Parametric Constraint Diagram

A Parametric constraint diagram will be used to describe the effective physical performance of the system wherein the physical characteristics that pose an intriguing threat to the system is evaluated using constrain diagram.

Thus, the constraint diagram blocks have been used as the building blocks and critical data of the mathematical computations that have been used there are used to construct the Parametric constrain diagrams here.

The equations (1) and (2) are the mathematical way of representing these parametric constraints that has been described in the previous section.

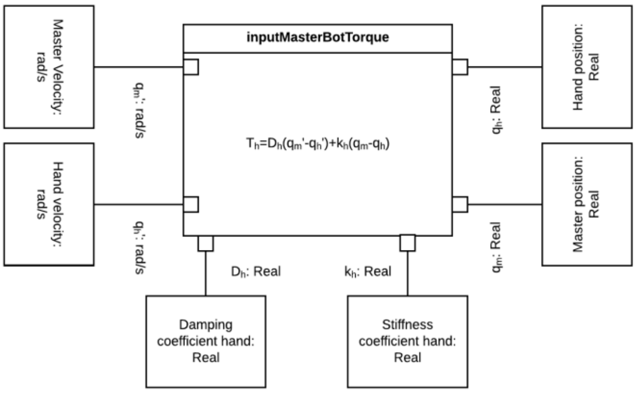


Figure 20 Parametric Constraint Diagram 1

A. Above Parametric constraint diagram, Fig., 20 describes factors affecting working of a Master robot.

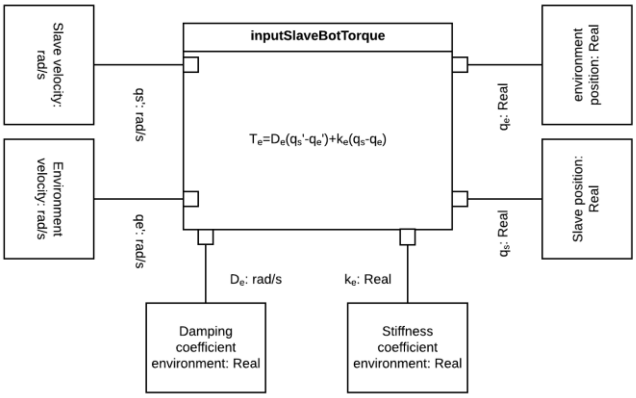


Figure 21 Parametric Constraint Diagram 2

B. Above Parametric constraint diagram, Fig.,21 describes factors affecting working of a Slave robot.

For the user to able to perform any operation, an aid of vision should be provided to him by the system. It is consists of two parts, the first part is a camera installed on the slave robot for view of the operation and the second part is a graphical user interface to enable an easy interaction between the user and the system.

Acknowledgment *(Heading 5)*

Contributions:

- Mostafa Elmasry, Use case, requirements and methodology **20%**
- Swathi Sudeendra Rao, Sequence diagram, constraint and parametric constraing diagrams **20%**
- Philip Okonkwo, Block diagram and V&V, **20%**
- Santhosh Kumar Ramareddy, Allocation diagram and systems characteristics **20%**
- Elis Gjana, Activity diagram and state machine diagram, **20%**

References

[1] Michael Barr; Anthony J. Massa (2006). "Introduction". Programming embedded systems: with C and GNU development tools. O'Reilly. pp. 1–2. ISBN 978-0-596-00983-0.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] Hasegawa, Y., Kitamura, T., Sakaino, S., & Tsuji, T. (2018, October). Bilateral Control of Two Finger Joints Using Functional Electrical

[4] Stimulation. In IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society (pp. 5433-5438). IEEE.

[5] Kitamura, Kazuya & Yashiro, Daisuke & Ohnishi, Kouhei. (2009). Bilateral control using estimated environmental stiffness as the master position gain

[6] Elitas, Meltem & Khan, Shahzad & Nergiz, Ahmet & Sabanovic, Asif. (2011). Task Based Bilateral Control for Microsystems Application. Automatika. 52. 107-117. 10.1080/00051144.2011.11828410

[7] Lenny Delligatti, “SysML Distilled A Brief Guide to the Systems Modeling Language”, Pearson Education Inc, 2014, pp 23-25.

[8] S. Friedenthal, A. Moore, and R. Steiner, “A Practical Guide to SysML The Systems Modeling Language ”, Elsevier Inc, 2012, pp 121.

[9] OMG Unified Modeling Language TM (OMG UML) Version 2.5
http://www.omg.org/spec/UML/2.5.

[10] A Practical Guide to SysML 3rd Edition by Sanford Friedenthal Alan Moore Rick Steiner

[11] Sparx Systems and Official Website :
www.sparxsystems.com