RESEARCH ARTICLE OPEN ACCESS

Mobile C-Arm Prototyping and Repositioning using Motion Capture Guidance

Alireza Yazdanshenas*, Armin Yazdanshenas*, Chung-Hyun Goh*[1]
**Department of Mechanical Engineering, University of Texas at Tyler, Tyler, TX, 75799 USA*
[1]Corresponding Author : Chung-Hyun Goh (cgoh@uttyler.edu)

ABSTRACT
One of the most important surgical tools in spinal surgery is the C-Arm X-ray System (C-Arm). This system works fine for most surgical procedures but falls short when the C-Arm must be repositioned during complicated surgical procedures for validation X-rays. The aim of this paper is to develop an accurate repositioning method with the use of motion capture technology. This will be a novel approach to creating a repositioning integrated system. A virtual prototype and a virtual platform were developed that quantified the dynamics of the C-Arm maneuvering. Next, a complete kinematic model of the C-Arm was developed to relate the joint angles with the respective Cartesian coordinates for critical points on the C-Arm. A fully automatic robotic C-Arm prototype was designed and manufactured to replace the actual C-Arm. Finally, the robotic prototype, the virtual platform, and the kinematic model were all systematically integrated using Vicon motion capture system to perform the automatic repositioning. The C-Arm prototype was able to be repositioned using the Vicon system with an average position error value of less than ½ inches cubed. With the development of the new repositioning system of the C-Arm, it is expected that professional interest in motion capture-based repositioning grows.
Keywords–C-Arm, Virtual Prototype, Repositioning, Kinematic Model, Motion Capture

---

Date Of Submission: 30-08-2019 Date Of Acceptance: 16-09-2019

I. INTRODUCTION

One of the most important elements of surgery, especially spinal surgery, is radiography. In simpler surgeries, an X-ray or an MRI is usually taken before the surgical procedure to aid in the diagnosis. After the surgical procedure is completed, doctors might recommend further X-rays to ensure surgical success. However, in more complicated surgeries, such as spinal surgeries, X-rays must be taken during the procedure to ensure proper outcome. One of the few X-ray systems that allow for imaging during surgery is the C-Arm X-ray System (C-Arm) shown in Figure 1 below.

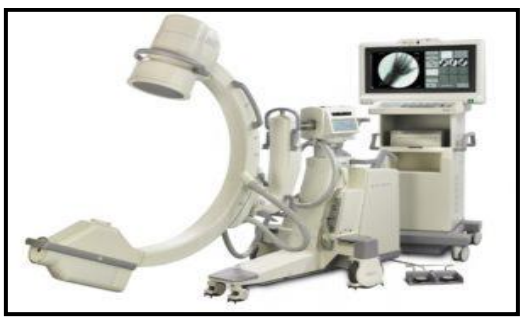

Figure 1. C-Arm X-ray System [1]

The C-Arm is a large 'C' shaped robotic arm with five degrees of freedom (DOF). The C-Arm can move up, down, back, and forth for two translational DOF. In addition, the C-Arm has three rotational DOF: tilt, wig-wag, and orbital movement. These DOF are visually represented in Figures 2-4.

This robot arm is moved through its ranges of motion manually by the X-ray technicians. Manual repositioning is currently one of the fastest ways of repositioning. New models of the C-Arm have been developed to allow X-ray technicians to move the C-Arm by motor control. Even when the movement of the C-Arm is motorized, the positions of each joint are still manually handled by the X-ray technicians. The C-Arm provides substantial advantages for the surgeons, allowing them to perform surgical maneuvers and to validate these maneuvers immediately. Some surgical procedures are more complicated and require the C-Arm to be moved or rolled completely away from the operating table. This is done to allow for more room for other equipment and procedures. Once those procedures are completed, the C-Arm is manually repositioned by the X-ray technicians. Manual repositioning of the C-Arm by the X-ray technician is usually not a problem. Although the manual repositioning is not exact, it gets the job done rapidly and with enough accuracy. In certain cases, the repositioning needs to be extremely accurate, and this is where the

limitation of the current models arise. In cases where repositioning accuracy is of the essence, much time and effort are spent trying to reposition the C-Arm so that the post-operation and pre-operation X-ray images are aligned and taken from the same references.

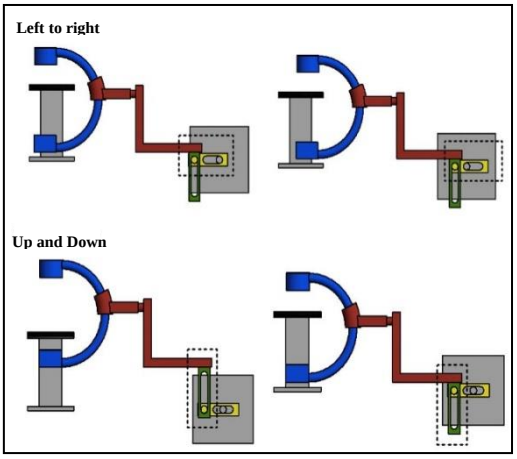

Figure 2. Translational Degrees of Freedom

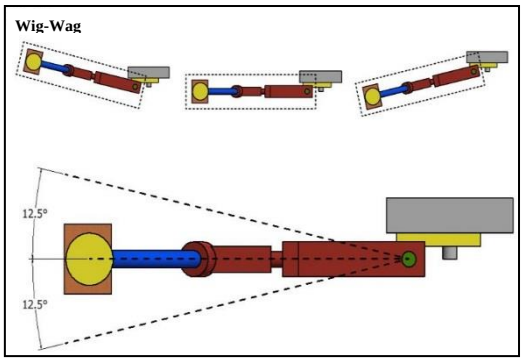

Figure 3. C-Arm Wig-Wag Movement

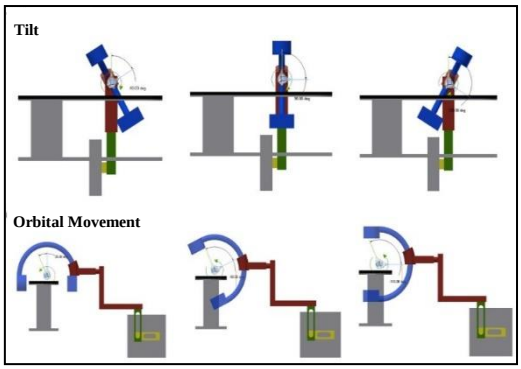

Figure 4. C-Arm Tilt and Orbital Movement

Because C-Arms are expensive instruments, any major or minor modification to the hardware is usually out of question. However, using motion capture as a sensing system for the C-Arm only requires a handful of reflective IR markers to be placed on the C-Arm. The key advantages that will make the integration of the C-Arm with a motion capture system marketable in the healthcare community are as follows: First, the proposed system does not need repeated calibration. While similar solutions have been proposed in the past, they have not been put into practice because of repeated calibration. When the Vicon motion capture (Vicon) system is calibrated once, it can work for days and hours without needing to be recalibrated. Although periodic recalibration is required to ensure quality, it is nothing like other propositions that need recalibration with every surgical procedure. Second, the reposition system developed here is modular, meaning that it can be applied to all C-Arms no matter the size or brand.

The tasks in this project aim to overcome these hurdles by creating an original kinematic model, developing a repositioning program that is accurate and versatile enough without needing constant calibration, and easily learned by the development of an intuitive Graphical User Interface (GUI). The original kinematic model will be a model that can be applied to any C-Arm no matter the size. This kinematic layout will allow offsets that can be changed based on the specific dimensions of the C-Arm in use. The kinematic model will then be programed to work in all scenarios of C-Arm use without needing constant calibration. The program should consider multiple scenarios or repositioning and be able to select the cored method. Finally, an intuitive GUI must be developed that allows for users with limited C-Arm experience to master the repositioning using the proposed method of this project. By creating an original kinematic model and a versatile program that is maneuvered with an intuitive GUI, this project will set itself apart from previously created methods.

II. LITERATURE REVIEW

Since the C-Arm made its debut in 1955, not much has changed. The simple-to-use physical design has stayed virtually the same except for the change in material selection and color. While most other equipment has been modernized with automatic moving joints, C Arm manufactures have not had much interest in integrating automatic positioning to the C-Arm. Because the C-Arm itself is expensive, additional automation is not desired since it marginally adds utility to the handling of the equipment. Only a small fraction of surgeries requires extreme accurate repositioning of the C-Arm. Therefore, surgeons who rely on accurate repositioning of the C-Arm have had to resort to aftermarket customization for reaching their goal of accuracy.

To achieve such an augmentation, a camera mirror system has been added onto the traditional C-Arm. Using the mirrors, the camera view is calibrated to exactly match the X-ray cone beam.

The live feed of the camera therefore allows surgeons to see from the view of the C-Arm. The modification for a camera augmented mobile C-Arm (CAMC) is shown in Figure 5 below.

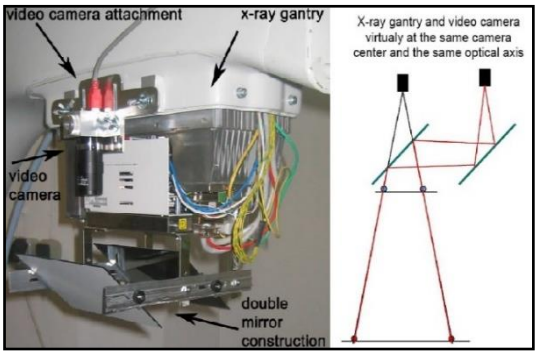

Figure 5. Double Mirror Modification for the CAMC [2]

In addition to camera augmentation, further modifications have been proposed and successfully implemented to enhance the augmented reality dynamics. For instance, RGB-D sensor, cameras that can perceive all color and depth, such as the Kinect sensors, have been added to further improve augmentation quality as shown in Figure 6. The usual overlay is based on 2D-3D vision calibration. The usual CAMC system is not able to tell the spatial relationships between surgeon's hands, tools and targets due to the lack of depth information in the image. As a result, surgeons' hands and tools are partly covered by the X-ray overlay. The Kinect sensor can then segment hands and tools according to depth data and create an enhanced X-ray overlay that will not block hands and tools. This would provide a more intuitive view of spatial relationships between targets, hands, and tools [3]. In other words, ideas of integrating the X-ray images with the body patient have been done successfully only if the C-Arm nor the patient are moving.

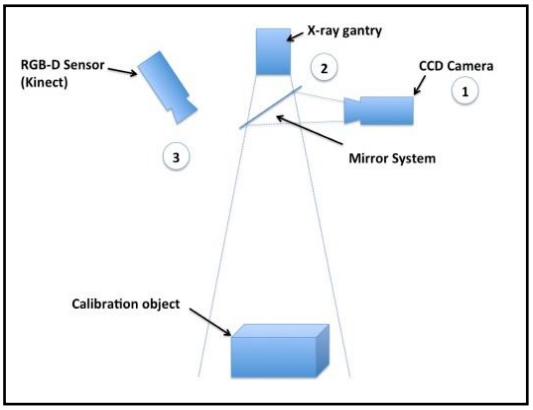

Figure 6. RGBD Integration with CAMC [3]

Naturally, after the progress of visual alignment with modified cameras reached the point of demising returns, the focus changed to finding the orientation and position of surgical tools in space. By the integration of optical tracker-based navigation systems, such as motion capture systems, reflective markers attached to surgical instruments became the next natural step in improvement of surgical utility. These tracker systems work by emitting IR light to spherical reflectors. The reflected light is than recorded by a set of fixed cameras, usually 6-12 cameras. Through careful image processing and calibration, the exact position of each spherical reflector can be known about the stationary cameras. Using such models, the starting and end points of lines can be projected very efficiently into the image plane, allowing real-time tracking of objects with relatively low computational effort [4]. Figure 7 shows the theory of acquiring the location of a reference point using a system of cameras.

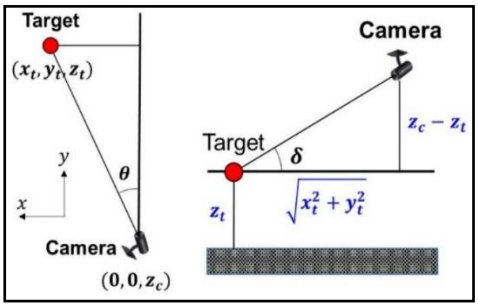

Figure 7. Schematic Representing the Detection of IR-Markers [5]

To manipulate and move robots, new and faster methods of kinematics had to be developed that would predict the robot's movement as accurately as possible. Robot kinematics is separated into two main sections, the forward kinematics and inverse kinematics. Forward kinematics, being the easier one of the two to calculate, concerns with the movements of the robotic joints to predict the exact location and movements in Cartesian space, whereas the inverse kinematics attempts to model the joint movements of the robot base on imported Cartesian values. Inverse kinematics problems are split into two branches, the ones that can be solved analytically, and those that need to be solved numerically [6]. The relationship between the forward and inverse kinematic modeling is represented in Figure 8.

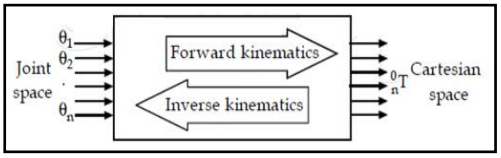

Figure 8. Relationship of Forward and Inverse Kinematics [6]

To further expand, each set of kinematics, forward or inverse, comes with its own specialized kinematic formulations. These formulations consist of the mathematical model, position kinematic model, differential kinematic model and the dynamic model. The mathematical models are subdivided into two further distinct groups called the geometric and the dynamic models. The geometric models are mainly concerned with positions of the robot, and dynamic models are concerned more with the movement vectors in space. Position kinematic models establish mathematical relation between joint angles and position whereas the differential kinematic models produce the velocity vectors of selected regions. Finally, dynamic modeling uses the position and the differential models to produce acceleration and force requirements for the selected robots. The dynamic models, which concern themselves with inertia and force, can be used to determine the optimal motor and power requirements for each robotic joint [7].

III. METHODOLOGY

3.1 Virtual Platform and Kinematic Model

For the case of the C-Arm, the virtual model became the first natural step in developing and proposing a better solution than current available models. The virtual model would provide an accurate description of the functions and demands of the C-Arm. These functions include the physical demands of joints and possible modeling for the control system.

First, using standard computer aided design (CAD) software, the general shape and functions of the C-Arm were developed. The virtual model consists of general and simple geometric shapes. Since the C-Arm Virtual Prototype (CVP) is not used for the actual manufacturing of the physical prototype, it is, therefore, beneficial to keep the components as simple as possible. Simple components are easier to draft in CAD programs and allow for faster processing time during virtual simulations, giving further reason in keeping the virtual model as simple as possible. The Autodesk Inventor assembly of the CVP can be seen in Figure 9. It is important to note that the CVP has as few components as possible This is also a key factor in the simplicity of the model. The total number of parts for this model is six components, and these six components allow for a total of five DOF. Two translational DOF and three rotational. All DOF were adapted from currently used C-Arms to provide realistic feedback of the mechanics and kinematics. The virtual model will be the key component in the extended development of the VP; therefore, the correct use of each joint and feedback are important.

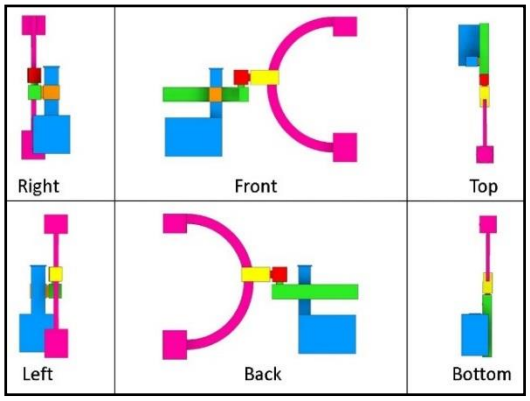

Figure 9. Different Views of the C-Arm Virtual Prototype

The VP is to represent how an actual C-Arm would behave in an operating room during surgery. The more realistic the VP, the more insight to the real scenario can be obtained, however, after a certain threshold the margin of returns is reached. This means that no significant gain will be provide after a certain level of detail is reached in the VP. For this project, it was decided that the VP should include a surgery room with a fixed operating table. In addition, a human manikin placed on the operating table will aid in how the real C-Arm maneuvers around a person in surgery. Lastly, six cameras are fixed around the surgery room to represent the aid of a motion capture system in the repositioning of the C-Arm. The full set up of the VP includes a physically accurately surgical environment and a fully mechanically functioning virtual C-Arm. With this setup, the surgical room and its components are completely stationary, and the CVP is a movable and dynamically realistic model.

To generate the VP, visual coding was used in MATLAB/Simulink. First, the CAD assembly of the surgery room and the CVP were given appropriate mating constraints that would be automatically recognize by Simulink. Then, by using Simscape Multibody, a MATLAB/Simulink extension package, the CAD assembly was converted into an Extensible Markup Language format (xml) file. This extension is a conversion of a CAD model to a mathematical/numeric matrix system, allowing for custom programming in MATLAB/Simulink. Finally, the xml file is uploaded into Simulink, allowing for a block diagram representation of the generated code and allowing for easy user input to change variables in the code. Below, in Figure 10 is the MATLAB/Simulink subsystem block network diagram of the virtual system and CAD model interaction.

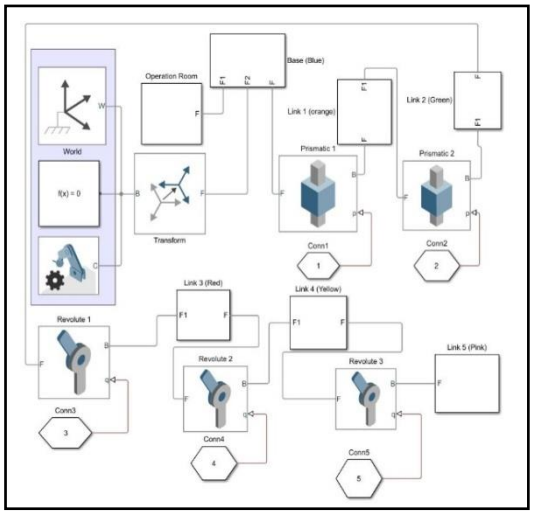

Figure 10. Simulink Block Diagram of the Visual System and CAD Model Interaction

In this visual code, the two translational linkages and the three rotational joints can be seen. The digital input to each joint is represented by five individual hexagonal blocks. The three blocks in the highlighted region represent the 3D world frame, MATLAB solver, and mechanical configuration block. All other blocks are block diagram representations of the six CAD components that make up the virtual C-Arm and the operation room. To acquire quality results from the VP, the C-Arm virtual model needs to be maneuvered by a human user. This led to the development of a GUI for the VP. The GUI will allow users to input desired translational locations and desired angular vectors to the virtual model. This input will allow the Simulation to react to the desired user input and maneuver the C-Arm virtual model through the VP.
The best way for a user to input their desired data is through manual sliders. These sliders will be actuated through the click and drag of the mouse with tick marks showing the graduated values that the user is selecting into the program. Once an input is made, the program runs and mechanically moves the virtual C-Arm through the VP with the appropriate force, acceleration and position predictions. These results can then be fed back to the user through the GUI. In the developed GUI for the VP the X, Y, and Z coordinates of the origin of the hypothetical X-ray cone beam and the coordinates of the region of interest are given. Figure 11 is a screenshot of the VP GUI with labeled sliders and coordinate outputs.

Finally, when the user maneuvers the GUI, Simulink produces a simulation of the C-Arm movement in real time. This simulation in addition to the virtual data allows for users, experienced or unexperienced, to understand what the strengths and weaknesses of the design are. Figure 12 shows the visual simulation results of the program according to the GUI input values. Four views are automatically generated to show in detail how the GUI input values move the CVP.

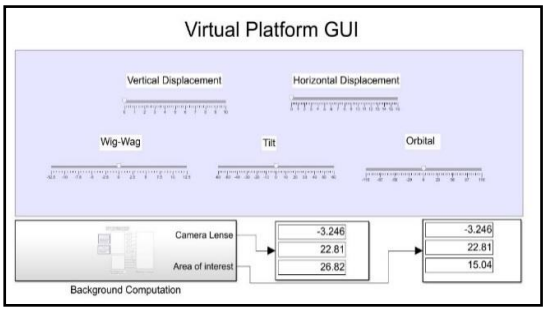

Figure 11. Virtual Platform GUI (all joints set to zero)

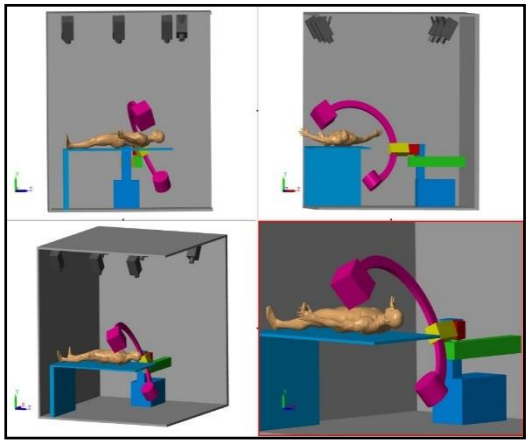

Figure 12. Simulink Virtual Platform Simulation

To reposition the C-Arm accurately, C-Arm movements in space must first be established. Because the C-Arm is essentially a robotic arm, the kinematics of the C-Arm can be generalized as the kinematics of a robotic arm. Kinematics is defined as the study of possible movement and configurations of a system [8]. The C-Arm only uses translational and rotational joints, something typical for most robotic arms. No matter how complicated the geometry of a robot, the links can always be generalized by the Denavit and Hartenberg (DH) link parameters. The DH link parameters and simple serial joints by applying a hand full of simple parameters. The first parameter is the angle of the joint of interest. This angle is always expressed about the local Z-axis or axis of rotation of the joint. The second parameter is determined by the length of the interested joint form its previous link along the Z-axis. Finally, the angle and length offsets about the respective X-axis determine the radius offset and altered orientation for the joint.

After developing the CVP and observing the movements in MATLAB Simulink, the forward kinematic model of the C-Arm was developed by applying the DH joint parameters. Thus, this

kinematic model enabled the C-Arm to simulate 5 DOF of its movement, which include two prismatic and three revolute joints. The prismatic joints move the C-Arm vertical and horizontally. The three revolute joints provide roll, pitch and yaw for the C-Arm camera X-ray source. In colloquial terms, those three revolute joints perform wig-way, tilt, and orbital rotations. Wig-wag corresponds to the yaw, tilt for the roll, and orbital movement for the pitch. To develop the kinematics, first a reference point was selected to be the origin for the C-Arm local coordinate system. From there on, all subsequent joints were modeled as shown in Figure 13.

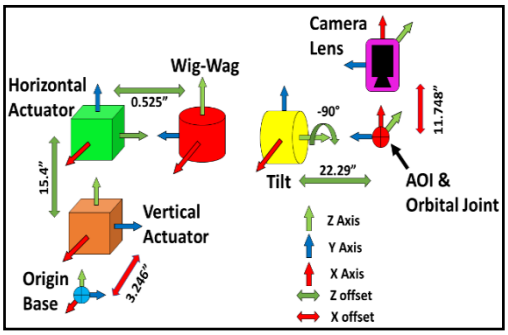

Figure 13. C-Arm Kinematic Model

Because access to a real C-Arm is extremely limited, the offset and length values used in the kinematic model are based on the C-Arm prototype (CAP). This prototype will be further introduced in the subsequent sections. To keep the kinematic model as intuitive as possible, the same colors as the CVP were used to represent each kinematic section. In the kinematic model depicted above, just like the traditional DH parameters, the Z-axis is the line of action ether by rotation or translation. The cubes represent prismatic joints and the cylinders represent revolute joints. It is important to note that the orbital joint does not exist in a physical manner and is represented by a hypothetical joint location at the center of the orbital movement [9]. The orbital joint is also naturally the center of the "C" shaped orbital link, making it the desired focal point for X-ray imaging or the Area of Interest (AOI). Lastly, the end effector of this robotic arm kinematic model is the tip of the beam cone.

At this point, it is good to remember that the whole purpose of the repositioning the C-Arm accurately is to take an image of the same point with the same orientation as the pre-procedural image. With this in mind and with a good look at the kinematic model, it can be concluded that the only two points that need to be accurately repositioned for the purpose of X-ray imaging are the AOI/orbital joint and the camera lens/ end effector. In other words, if the C-Arm camera lens and the AOI are properly positioned, a quality repositioned X-ray can be taken. This reasoning is also exactly why the GUI for the CVP only output the coordinates of the camera lens and the AOI.

Unlike forward kinematics, inverse kinematics does has not have general formulation/ method. Whereas in the forward kinematics the DH parameters and matrix can be used to model any robot, the inverse kinematics are still solved with more primitive mathematics [10]. Because of the lack of more ideal mathematical models, inverse kinematics are still solved by using a combination of simple algebra and simple trigonometry. This is also true for even the most versatile robotic arms and autonomous drone applications [11]. For example, even the best of industrial robotic arms still needs six joints to provide six DOF. Usually, the first three larger joints are responsible for providing the exact cartesian coordinates while the last three smaller joints provide the orientations with respect to each axis. However, if there existed better inverse kinematic model approaches, the position and orientation of the robotic end effects could be accomplished with far fewer joints. In general, the more joints a robot has, the more difficult the inverse kinematics will be.

To develop an inverse kinematic model for the C-Arm and the CAP, a similar approach to other robotic manipulators with five DOF was taken [12]. The inverse kinematics for the CAP was developed from the local origin on the base to the AOI, using the same elements as shown in Figure 13. A hierarchy of operation was set to acquire the correct joint angles based on the coordinate input of the AOI. This hierarchy consists of calculating the joint positions to reach the Z-coordinate first. Then, the program calculates the wig-wag angle needed to reach the expected X-coordinate value. Based on the proscribed joint positions to reach the Z- and X-coordinates, the joint positions to find the Y-components are derived and implemented.

To solve the inverse kinematic in the full robotic range, the inverse kinematic model had to be broken into four sperate sections. The sections were created based on the geometric limits of the physical protype. All four cases are shown in Figures 14-17.

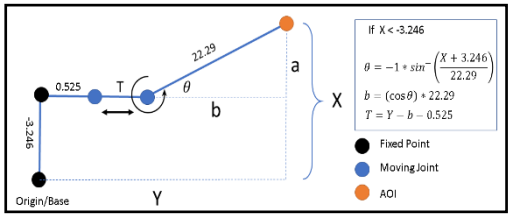

Figure14. Inverse Kinematics Model for Case 1

For the first case seen in Figure 14, the AOI is rotated into any positive wig-wag angle. This means that the X-components of the AOI is a sum of the X offset and the value of "a". Once the angle

theta is found, the length of "b" can be calculated, added to the Y offset, and subtracted from the total Y value, resulting in the horizontal translation of "T".

In the second case, the AOI lies in a negative wig-wag angle. However, the X-component is smaller than zero and yet larger than the X offset of -3.246. Therefore, the "a" value is the difference between the X-component and the total X offset. With this information, the angle theta and length "b" can be found. Finally, the horizontal offset "T" can be found as shown in Figure 15.

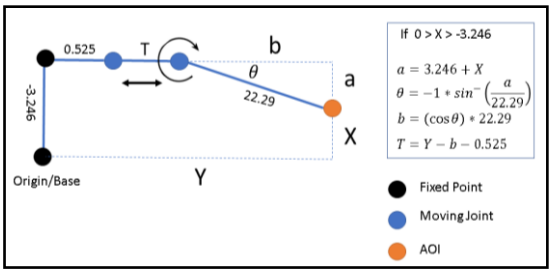

Figure 15. Inverse Kinematics Model Case 2

The third kinematic case (Figure 16) exists when the AOI has a positive X-component. This means that the value for "a" is a combination of the X-component and the X offset -3.246. Once again, after finding the value for "a" theta and "b" can be calculated. Finally, using "b" and the Y offset, the horizontal translation "T" can be found.

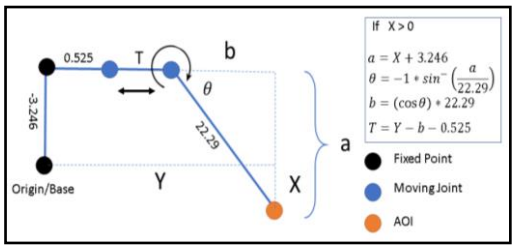

Figure 16. Inverse Kinematics Model Case 3

The fourth and final case for the inverse kinematics seen in Figure 17 is the simplest to solve because the X-component for the AOI is the same exact value as the -3.246 X offset. This leads to a theta of zero and an automatic "b" value of 22.29. With this information, the last possible value of "T" can be calculated and implemented.

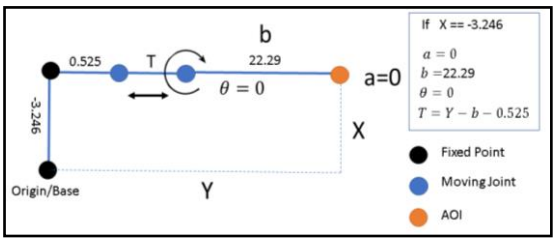

Figure 17. Inverse Kinematics Model Case 4

3.1.1 Physical Prototype Design and Manufacturing

The CAP was developed as a mixture based on the actual C-Arm and the CVP. The decision to do so was based on the fact that the CVP is much too simple of a design to be real-life applicable and manufacturable and that the actual C-Arm is way too complicated for the scope and purpose of this project. A comparison of all three C-Arm system can be seen in Figure 18.

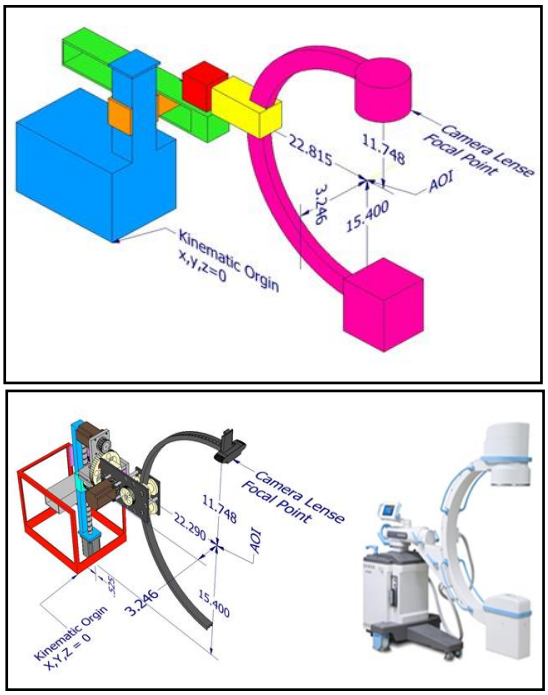

Figure 18. C-Arm Comparison: Virtual Prototype (up), Physical Prototype and Actual C-Arm (bottom)

To keep the cost of the CAP down, standard parts were used wherever possible. For example, the "C" shaped link is a modified 24" bicycle rim. For the translational joints, stepper motor driven linear actuators were use because of their proven reliability, modularity, and ease of access. The custom parts and mechanisms of the C-Arm prototype were developed with production steps in mind as the first criterion. The approach was to use readily available stock parts and make the least amount of modifications to them in order to develop custom parts. Each part was design such that the manufacturing would only require a drill press and a lathe for successful completion. A total of 3 aluminum stock types were used to manufacture all custom parts and sections. Load bearing link sections were built by machining 3" x 0.5" 6061 aluminum bars. Non-load bearing sections, mostly found in the orbital mechanism, were made from 3" x 0.25" 6061 aluminum bars. 6061 aluminum round stock of 2.25" was also used to machine out custom sprockets.

However, not all part productions could be limited to just using a drill press and lathe for manufacturing. A total of five parts had to rely on CNC machining for proper manufacturing. These parts included the base frame, the base plate, the tilt base, the large side plate, and the small side plate based. CNC code, known as G-code, was automatically developed using Computer Aided Machining (CAM) in Autodesk Inventor. A High-Speed Machining (HSM) plug-in was used to apply the CAM modeling to custom parts that required machining. All HSM procedure steps were simulated as shown in Figure 19 below.

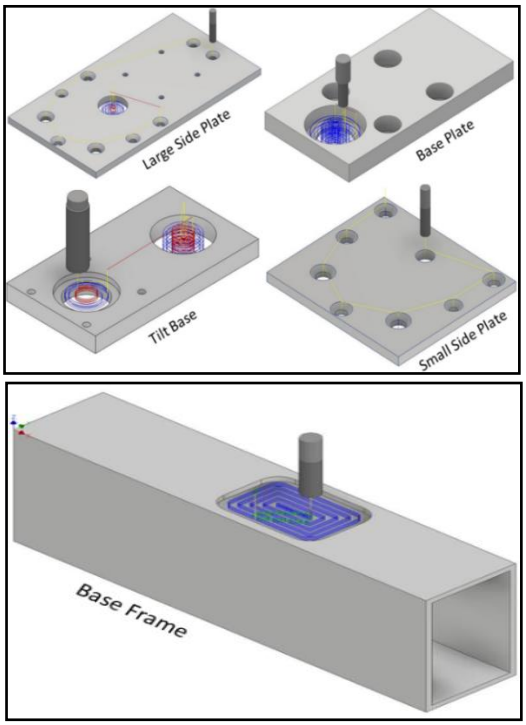

Figure 19. Computer Aided Marching using Inventor HSM

The lines in this figure show the tool path of the entire operation. All yellow line segments are rapid tool movements where the tool is not in contact with the stock. Blue line sections represent tool path sections where the tool is in contact with the stock and actual material is removed. Red lines usually represent plunge sections, where the tool lowers onto the stock to make the first cuts; otherwise, red lines represent tool collisions or tool paths that are prone to collision. Any green line is a path deviation to a new cut where stock and tool contact is maintained continuously. Post processing of the CAM model was converted for the use in Mach3 by Artsoft. Mach3 was chosen because this G-code language is quite universal for most CNC machines, allowing for replication of parts in most machine shops.

Motor selection for the CAP was a decision that needed to be considered carefully. There existed three main options for motor choice: DC motors, servo motors, and stepper motors. DC motors produce the most torque and are the most efficient, but because dc motors need additional sensors to detect angular position, the decision was made to exclude them for the design. Next, servo motors were considered as a viable option. Servo motors are relatively easy to program and provide a decent amount of torque compared to their weight. In addition, servo motors do not need additional sensors to detect their angular position. However, two factors led to the decision to move forward with other options. The first factor was the fact that servo motors are limited in revolution, and in order to use the linear actuators, a consistently rotating motor is needed. The second reason was the fact that most servo motors are limited in resolution to half of a degree. Such resolutions are usually enough for most applications but are not small enough to be trusted in the repositioning of the C-Arm prototype. Although stepper motors have the lowest torque to weight ratio of any motor, stepper motors are the most accurate motors available. The decision was made to move forward with a Nema 23. The low torque output of these motors had to be compensated with pully systems to allow for proper function. More information about the Nema 23 can be found in Appendix E. Finally, to avoid skipping steps, the motors were set at a speed of 1000 steps per second and a value of 800 steps per revolution. Figure 20 shows the manufactures torque curve in terms of torque vs. pulse per second (PPS) for the Nema 23.

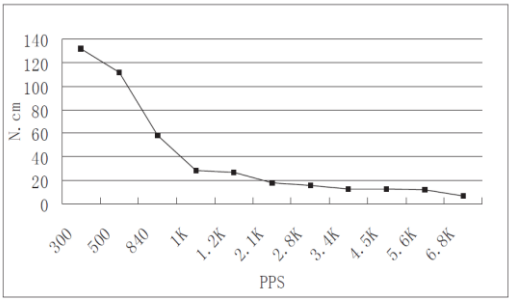

Figure 20. Nema 23 Manufacturer Torque Curve [13]

3.1.2 Motion Capture Integration with Mobile C-Arm for Repositioning

To validate and test the CAP and the developed kinematic model, Vicon was used [14]. Essentially, the C-Arm prototype's movement is objectively gathered and compared to the kinematic model. Because the translational joints of the C-Arm prototype are easily validated using traditional measuring methods, only the rotational joints of the C-Arm prototype were tested with Vicon. First,

spherical reflective markers were placed on the C-Arm. One marker was placed on the kinematic origin of the C-Arm prototype. This critical marker will be used to convert the global coordinate values of all markers placed on the C-Arm prototype to local coordinate values relative to the kinematic origin. Second, two other markers are placed on the base to establish a ridged body as the reference for all other movements. Next, a marker is placed on the camera lens to provide the exact coordinates of that point. In addition, a second marker is placed directly opposite of the camera lens onto the detection plate of the C-Arm prototype. Finally, two other markers are placed on the orbital link of the C-Arm prototype to model a ridged object. This set up can be seen in Figure 21. This figure shows an overlay of the detected markers onto the live video feed of the C-Arm prototype.

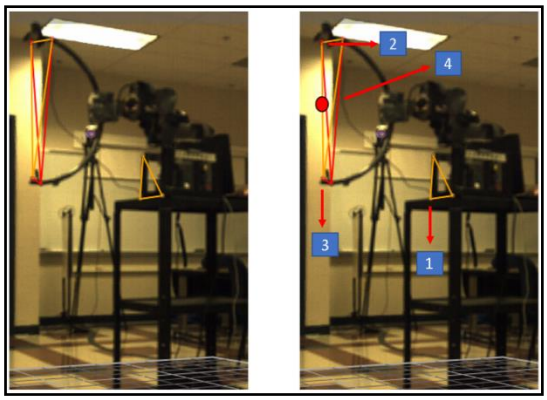

Figure 21. Vicon Video Overlay onto Detected Markers with Labels

To properly reposition any C-Arm for imaging, only two points of the C-Arm need to be properly repositioned as discussed in the Forward Kinematics subsection of this thesis. However, the AOI is not a physical existing point and cannot have a Vicon marker placed on it. To overcome this issue, the AOI point is calculated by applying the midpoint formula, in Equation (1), to the marker placed on the camera lens and the marker placed on the detection plate. Where Mid is the calculated midpoint and x, y, z represent the coordinates of the camera lenses and detection plate with the subscripts denoting the order.

$$Mid = \left(\frac{x_1 - x_2}{2}\right), \left(\frac{y_1 - y_2}{2}\right), \left(\frac{z_1 - z_2}{2}\right) \quad (1)$$

Although 7 markers were used in the Vicon model, only 3 markers are critical for the validation of the Kinematic model. Looking at Figure 21, marker number one is the local origin of the C-Arm prototype. Marker number two is the center point of the camera lens. Marker number three is the center point of the detection plate. Finally, point number four is the calculated AOI. With this set up, the C-Arm physical protype is fully integrated with the Vicon system.

IV. RESULTS AND DISCUSSION

During the Vicon testing procedure, the data acquired was converted to MATLAB arrays for mathematical and visual processing. The cartesian coordinates of the camera lens and the calculated AOI during each joint testing are graphed in the figures.

Figure 22 shows the path and positions the camera lens takes during the Vicon testing of the wig-wag joint. The wig-wag Joint was at zero degrees during the start position. When the joint was moved to a positive 10 degrees position, the first point was recorded. Lastly, the wig-wag joint was moved to -10 degrees resulting in the second point for this test. The shape of the path is also intuitive with the expected movements.

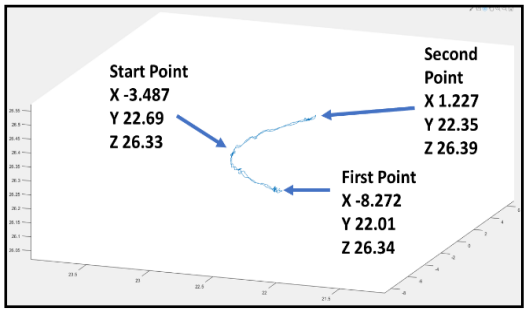

Figure 22. Wig-Wag Testing Coordinates of the Camera Lens

Figure 23 shows the interpolated AOI path and positions during the wig-wag testing of the C-Arm. The values of this point are interpolated by finding the midpoint between the camera lens and the detection plate. At zero degrees, the starting point was recorded. Then, at 10 degrees, wig-wag movement resulted in the first point. Finally, the second point corelated with a wig-wag angle of -10 degrees. The path that the AOI follows during the wig-wag testing is also intuitive with what is expected out of such movement.

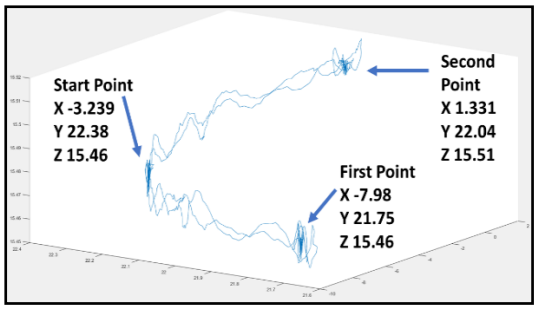

Figure 23. Wig-Wag Testing Coordinates of AOI

Figure 24 shows the path and position of the camera lens during the tilt testing procedure. At the beginning, the tilt angle of zero degrees resulted in the start point for this test. Then, the tilt angle was set to 40, resulting in the first point for the camera lens. Finally, the tilt angle was set to a complementary -40 degrees, providing a second point.

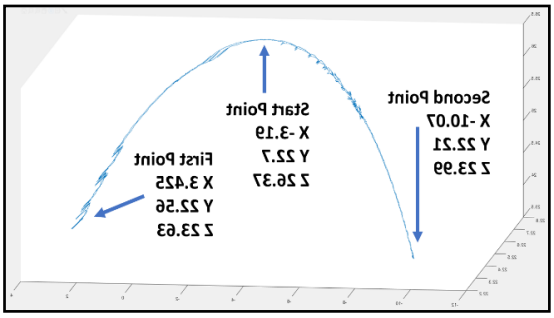

Figure 24. Tilt Testing Coordinates of Camera Lens

Figure 25 shows the path of the AOI during the tilt test. It is important to note, that if all Vicon markers were placed perfectly, and no mechanical error existed in the physical prototype, the AOI should be ideally stationary during tilt movements. However, because of vibration, mechanical inaccuracies, and Vicon marker errors, the AOI is not perfectly still during the movements from starting point to second point. The interpolated AOI path and position of however are in a small proximity and can be considered still.

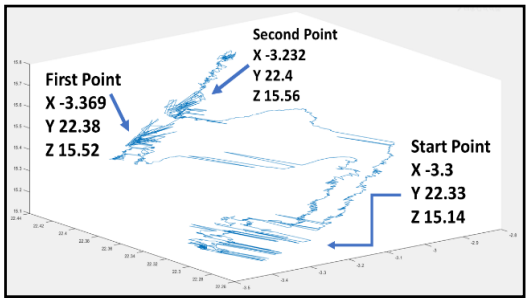

Figure 25. Tilt Test Coordinates of AOI

Figure 26 shows the camera lens movement and positons as the orbital angle was moved from the starting positon of zero degrees to a first position of 10 degrees. For the orbital movement, no second position was tested. The starting point and the first point are labled with their respective cartesian coordinants.

Ideally, during the orbital movement of the CAP, the AOI remains still and does not move. However, due to mechanical slack, less than perfect marker placement and vibrations, the AOI is not perfectly still as can be seen in Figure 27. Because the movement of the interpolated AOI is small, the point can be considered still for this practical application.

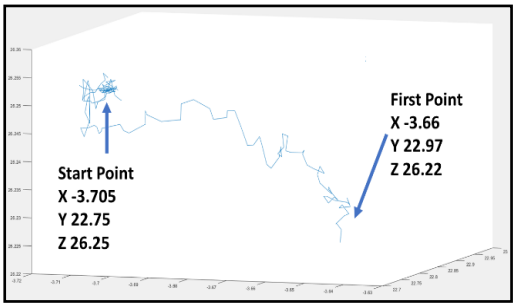

Figure 26. Orbital Test Coordinates of Camera Lens

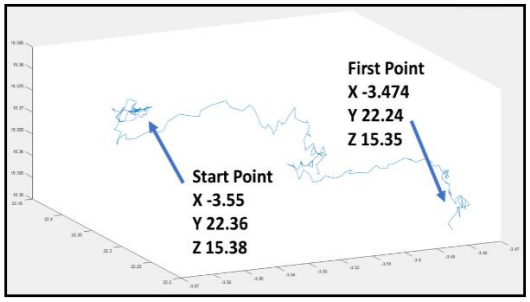

Figure 27. Orbital Test Coordinates of AOI

The CAP was integrated with Vicon with ease. Select markers were placed on critical points whose coordinates needed to be known for proper repositioning calculations. One marker was placed directly onto the kinematic origin of the CAP. This was done to convert the global coordinates of all markers to the local coordinates needed for proper repositioning and testing analysis. Additional markers were placed on the camera lens and the detection plate center. Using linear interpolation, the coordinates for the AOI were calculated. The integration model and integration procedures for the CAP can also be applied to other C-Arm models. The CAP provided very promising results. In order to acquire good results through Vicon testing, the GUI, C-Arm virtual model, kinematic model, and the CAP all had to work properly in synchronization. Therefore, by acquiring the expected results, we can be reassured that all four tasks/subsystems work with one another properly. The testing was done to one joint at a time to keep procedures as simple as possible. The positions of two points on the CAP, the camera lens and the AOI, were recorded during the testing procedures. These two recorded point coordinates were compared to the theoretical coordinates of the same points calculated by the kinematic model. The difference between theoretically calculated points and those provided by Vicon are on average 0.4 inches cubed. These results are reasonable and ensure that the entire integrated repositioning system works well.

V. CONCLUDING REMARKS

The C-Arm is one of the most important surgical tools because it provides surgeons the ability to take X-ray imaging during surgery on demand. In a handful of spinal surgical procedures, the C-Arm needs to be moved out of the way to make room for other equipment and maneuvers. To develop a simple repositioning system that works universally and without restriction, the C-Arm was integrated with Vicon motion capture system, allowing surgery personal to have feedback of current positions of the C-Arm relative to its previous positions. To complete the objective of automatic repositioning, first a set of tasks needed to be completed. These first included the development of a virtual prototype maneuvered by a GUI inside a VP. The second task consisted of developing a complete forward and inverse kinematic model of the C-Arm and integrating the kinematics into the GUI. Thirdly, a fully functioning robotic C-Arm prototype was developed that was programmed to be moved in synchronization with the VP. Finally, the fourth task was to integrate the C-Arm robotic prototype with Vicon to allow for the digital cartesian feedback into the kinematic program.

This novel approach manages to reach its objective of accurate repositioning the C-Arm and still manages to avoid all restrictions that have kept similar ideas from being implemented. By applying this system, we move significantly closer to the goal of autonomous X-ray imaging during surgery. It is expected that all, if not most, surgical centers that already use motion capture during surgery will implement and use the proposed methods of repositioning developed int this thesis. If successful in those cases, a larger spread of motion capture integration with medical imagining devices can also be expected. By doing so, surgeries that rely on accurate C-Arm repositioning will benefit from significantly shorter repositioning time and radiation exposure to patients, and surgical personnel will also decrease. This, in fact, will have the chain reaction of reducing surgery cost and surgery risk such as infections and anesthesia complications.

One of the next important steps for the continuation of the C-Arm repositioning project will be the development of obstacles in the repositioning program. For example, how can a robot arm move itself from one cartesian space point to another and still consider the physical limitation of its joints? In addition, no-go regions might also exist, where the robot arm must avoid a physical obstacle or barrier. In the case of the C-Arm, the obstacles are the patient and the operating table. These kinds of scenarios place challenges for path planning, where the route from one point to the next is never a straight line. Solutions to these kinds of problems have been proposed by J.T. Schwarz and M. Sharir, where attempts were made to find the shortest robotic path between two points while avoiding obstacles [15].

ACKNOWLEDGEMENT

This research was supported/partially supported by the Office of Research and Technology Transfer and The College of Nursing and Health Science at The University of Texas at Tyler. The significant contributions of Dr. Y.T. Wang and Dr. X. N. Dong are appreciated in the development of this paper.

REFERENCES

[1]. Bluestone Diagnostics, GE OEC 9800 PLUS MOBILE C-ARM, General Electric Company, 2019. [Online]. Available: https://bluestonediagnostics.com/product/ge-oec-9800-c-arm/. [Accessed 17 3 2019].

[2]. N. Navab, Camera Augmented Mobile C-Arm (CAMC): Callibration, Accuracy Study, and Clinical Applications, IEEE Transactions on Medical Imaging, 29 (7), 2010, 1414.

[3]. H. Xiao, N. Navab, B. Furest and J. Fotouhi, RGB-D Camera Integration into Camera Augmented Mobile C-arm (CamC), Computer Integrated Intervential Systems Laboratory, Johns Hopkins University Technical Report, 2015.

[4]. P. Azad, Combining Apperance-based and Model-based Methods for Real-Time Object Recognition and 6D Localization, International Conference of Intelligent Robotics and Systems, Beijing, China, 2006.

[5]. A.Yazdanshenas, E. Morrison, C.-H. Goh, J.K. Allen, and F. Mistree, Repsitioning Method of C-Arm Using Motion Capture on the Virtyal Reality Platform, IASTEM International, Seoul, South Korea, 2018.

[6]. S. Kucuk and Z. Bingul, Robot Kinematics: Forward and Inverse Kinematics, Industrial Robotics: Theory, Modeling and Control, IntechOpen, 2006.

[7]. A.C. Majarena, J. Santolaria, D. Samper, and J.J. Aguilar, An Overview of Kinematic and Calibration Models Using Internal/External Sensors or Constraints to Improve the Behavior of Spatial Parallel Mechanisms, Sensors, 10 (16), 2010, 10256-10297.

[8]. J. M. Selig, Introduciton to Robotics(Hemel Hampstead: Prentice Hall, 1992).

[9]. R. Vasilchenko, F. Simankin, G. Ziyakeav, and A. Simankin, Kinematics of the orbital movement of a digital X-ray scanner of annular pipe seams,MATEC Web of Conferences, 158, EDP Sciences, 2018, 10301.

[10]. S. Dereli and R. Köker, IW-PSO Aproach to the Inverse Kinematics Problem Solution of a 7-DOF Serial Robot Manipulator, Sigma Journal of Engineering and Natural Sciences, 36 (1),2018, 77-85.
[11]. A. Khalifa, M. Fanni, A. Ramadan, and A. Abo-Ismail, New Quadrotor Manipluation System: Inverse Kinematics, Identification and RC-Based Control, International Journal of Recent Advances in Mechanical Engineering, 4 (3),2015, 39-58.
[12]. D. Xu, C.A.A. Calderon, J.Q. Gan, H. Hu, and M. Tan, An Analyis of the Inverse Kinematics for a 5-DOF Manipulator, Inernational Journal of Automation and Computing, 2 (2), 2005, 114-124.
[13]. CNSOYO, Changzhou Songyang Machinery & Electronics Co., Ltd., 2014. [Online]. Available: http://www.cnsoyo.com/?gclid=CjwKCAjwwtTmBRBqEiwA-b6c_60yDw4Ox3-XRbRokI5SiC_mN7RLChTlxpsnVDPnAowJ2Q7lVN2sGBoCz1YQAvD_BwE. [Accessed 10 05 2019].
[14]. P. Eichelberger, M. Ferraro, U. Minder, T. Denton, A. Blasimann, F. Krause, and H. Baur, Analysis of Accuracy in Optical Motion Capture- AProtocol for Laboratory Setup Evaluation, Journal of Biomechanics, 49 (10), 2016, 2085-2088.
[15]. J. T. Schwartz and M. Sharir, Motion Planning and Related Geometric Algorithms in Robotics, International Congress of Mathematics, Berkley, California, 1986.

Alireza Yazdanshenas, Armin Yazdanshenas and Chung-Hyun Goh, "Mobile C-Arm Prototyping and Repositioning using Motion Capture Guidance," International Journal of Engineering Research and Applications (IJERA), Vol. 09, No. 09 (Series-II), 2019, pp. 32-43