

DEVELOPMENT OF A VIRTUAL D.O.F. MOTION PLATFORM FOR SIMULATION AND RAPID SYNTHESIS

K.F. Hulme*
State University of New York at Buffalo
NYSCEDII
5 Norton Hall
Buffalo, New York 14260-1810
hulme@buffalo.edu

A. Pancotti†
University Park Campus
University of Southern California
Department of Aerospace Engineering - Astronautics
Los Angeles, CA 90089
pancotti@email.usc.edu

Keywords: virtual prototyping, computational methods, motion simulation, heuristic error minimization, rapid synthesis, risk reduction, scientific visualization

Abstract

This paper describes the design, development, and utility of a fully operational computer simulation of a 6 degree-of-freedom motion platform. Such devices are commonly used for a wide variety of flight simulation, driving simulation, location-based entertainment, and training applications worldwide. Such a “virtual motion platform” will provide a fast, easy, and safe mechanism by which new candidate motion excursions can be tested and refined. As such, the integrity of the expensive and delicate physical hardware need not be compromised by motion patterns that might be detrimental to the limits of the hardware. The development of the NYSCEDII Virtual Motion Platform (NVMP) began with the design of a highly detailed CAD model of our existing Moog 2000E motion platform. Its development will be outlined in brief. With the static CAD model developed, the authors were then able to convert the model into a dynamic entity by using an inverse kinematical approach. The utility of these previously published algorithms will be presented, and their practical performance discussed. The NVMP has been designed to be highly user-interactive so as to allow for human “in the loop” intervention and control, and numerous of these features will be outlined and their utility demonstrated. Quantitative and qualitative comparisons between the NVMP and the physical motion platform will be discussed. Finally, the current state of the NVMP, as well as future planned features for the simulator (such as the incorporation of sophisticated washout algorithms and I/O plotting utilities) will be presented and discussed at the closure of this document.

* Member AIAA, Research Scientist

† Student member AIAA, Research Assistant

Background

A *motion platform, parallel manipulator*¹⁸, *motion base, or Stewart Platform*^{30,6} is a powered, mechanical, self-contained motion system. Such systems are commonly used for motion simulation and training activities, and common applications include: flight simulation, driving simulation, industrial equipment training, and location-based entertainment (i.e. amusement rides and simulation), to name a few.



Figure 1: Moog 2000E motion platform

Moog, Inc. of East Aurora, New York donated a 6 degree-of-freedom (D.O.F) 2000E model motion platform to the NYSCEDII research facility for which the proposed research will be performed. Refer to Figure 1. The 6 D.O.F's present on the Moog motion system are commonly referred to with the following nomenclature. Translation is defined as “heave” (up and down – *Z axis*), “surge” (forwards and backwards – *X axis*), and “sway” (side to side – *Y axis*). Rotation is defined as “roll” (tilt about the *X-axis*), “pitch” (incline about the *Y-axis*), and “yaw” (twist about the *Z-axis*). Refer to Figure 2 for a detailed illustration of the 2000E model reference system.^{24,26}

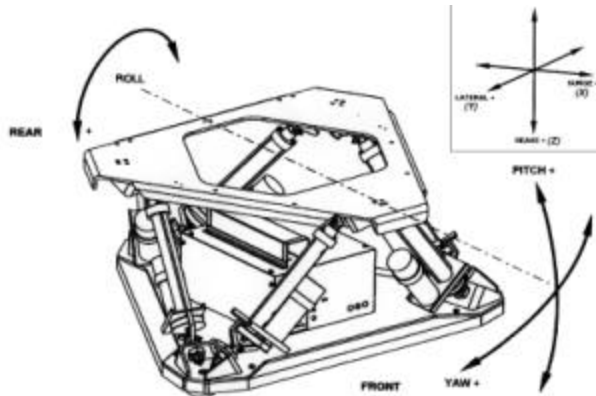


Figure 2: hardware reference frame

Motivation

Software programmers and hardware technicians who design input ride profiles for these motion systems described previously often encounter difficulties while testing with the physical system. The hardware itself is financially costly, and mechanically intricate and delicate. To realistically emulate the motions of a fast moving ground vehicle, a fighter plane executing a complicated maneuver, or an amusement park ride simulation attempting to thrill its passengers, the motion system is often subjected to severe, potentially (mechanically) harmful motions. Not only might the physical system be compromised by a newly developed motion profile (during testing), but passenger safety will certainly be a concern once the motion designers deem the ride profile “safe” for humans to ride. It would seem imperative that realistic computer simulation methods be developed that allow designers to more safely and accurately assess the response of the motion system to hypothesized external inputs within a controlled, realistic, and visual testing environment. Clearly, this is the primary motivation for this research.

One can see that the physical hardware associated with a motion platform is large, bulky, and cumbersome. The 2000E model Moog motion base is the smallest platform they manufacture, and is intended for small payloads and relatively small-scale motion tasks. Most practical motion platforms (used for flight simulators, industrial equipment training, truck driving simulation, etc.) would require even larger motion systems. Clearly, a space-efficient virtual simulation would be beneficial for obvious reasons. A simulation-based motion platform would thus serve as a useful, compact simulation resource for large sessions of developers as an aide for training. It would be impractical and non-productive to attempt to train, hands-on, a group of developers in a room in front of a single motion platform, and unrealistic to conduct a large-scale

training session such that each trainee has access to his/her own physical motion platform simultaneously. With a virtual motion platform, a practical, hands-on training session could instead be achieved such that each trainee has access to his/her own “hardware” in virtual space, and the trainer could supplement discussions in the virtual world with references to the physical hardware itself, which could perhaps be on display in the training room.

Clearly, the motion platform is a classical example of a spatial mechanism. Complex kinematics are required to maneuver the platform, and to determine the “output” of the platform (i.e. the values of the 6 degrees of freedom) with respect to the “input” (i.e. the positions of the moving pistons and cylinders), and kinematics required to solve the inverse problem. As such, a properly functioning and user definable virtual motion platform could serve as an exemplary tool for undergraduate/graduate education for students studying system dynamics, machine mechanism design, and/or control system theory. For example, a digital simulator of this nature could be used to study the global effects of each of the six “legs” of the motion base – i.e. what would happen to the total degrees of freedom of the entire system if one of the “legs” were removed? How would a comparable motion output be generated with a 3 D.O.F. simulator instead of a 6 D.O.F.?

With these project motivations stated formally, we hereby propose the design and development of a computational virtual motion platform. Through this form of virtual prototyping, such a toolkit will certainly assist with hardware/passenger risk reduction much earlier in the process of candidate motion design, and provide a safe avenue for testing, verifying, and calibrating parallel manipulators. Through the use of such virtual simulation, motion platform developers and scientists will be able to more rapidly and easily synthesize potential motion excursion paths. Clearly, this computer simulator will provide the end user with the ability to analyze a greater number of candidate “design configurations” per unit time. This, for example, will greatly assist a developer in synchronizing the motion platform dynamics to a corresponding predefined “scene graph” ride file. The proposed research also incorporates a numerical optimization component, as mapping (infinite) real world motions to a (finite) motion platform movement space involves some degree of input scaling and “compromise” by way of error minimization and motion “washout”^{27,10,1,12}

The next section presents a brief survey of literature relevant to this research project, and further justifies the need for a virtual motion platform simulation.

Literature Review

Numerous related and relevant research efforts have both motivated and influenced the work proposed here. A small sampling of this past literature is presented as follows, all of which fall under the general category of computer-assisted simulation pertaining to the design and practical application of parallel manipulators and spatial mechanisms.

The COPRIN research team is composed of researchers working in the field of constraint programming and others working on interval analysis. Their shared competencies are used for solving systems with an emphasis on geometrical problems and on mechanism theory. Their software utilities include: *FK-stewart*, which enables users to compute the solutions of the forward kinematics of a Stewart platform parallel robot¹⁵. *xworkspace* deals with workspace related problems (i.e. the limited range of the linear actuators, the mechanical limits of the passive joints, and linkage interference) for Gough-type platforms – i.e., two rigid bodies, the base and the moving platform, connected by 6 articulated links. This program is able to: compute the workspace of a robot when the orientation of the moving platform is fixed or when the position of the moving platform center is fixed, verify that the straight line trajectory between two postures of the moving platform lie within the workspace, and find a path within the workspace for joining two positions of the center of the moving platform^{20,21}. *visu_robot* is a tool that enables users to visualize various types of parallel robots, in various positions, and *design* is a software intended for computing the geometry of all the parallel manipulators whose workspace include a specified work region²².

Lee et al. designed a parallel link manipulator that has been designed for usage as a robot wrist. In computer simulation, it was shown that this manipulator can be used to perform tasks such as position control, path tracing, and force control¹⁴. Gosselin et al. executed the comparison of various spatial D.O.F. architectures are performed using SIMPA, a specialized CAD tool developed for the kinematic analysis and optimization of parallel mechanisms. The architectures were compared on the basis of their kinematic properties, such as workspace and rigidity, and in terms of their ability to simulate particular aircraft maneuvers⁸. Lu designed a novel computer aided geometric approach for designing the computer simulation mechanisms of spatial parallel manipulators with 3-6 driving limbs. Several new spatial parallel manipulators were synthesized, and some common computer aided geometry constraints and definitions for designing the simulation mechanisms were presented¹⁷. Yia and

Leinonen used a computer simulation approach based on Stewart's parallel mechanism, to design a virtual 3-D orthogonal 6rod machine tool with 6 degrees of freedom. Using reverse motion analysis, the kinematic curves of the 6 rods and the feeding process were subsequently simulated and analyzed³³.

Bonev and Ryu presented a geometrical method for the workspace analysis of 6-PRRS-type parallel manipulators. A geometric algorithm was described for the computation of the constant-orientation workspace (implemented in CATIA), and the influence of the different design parameters on the workspace, as well as on other properties of the manipulator, was discussed³. Zhao and Sun described an image-based robot animation technique. The technique utilizes the eigenspace method to generate compact representations of images from a set of reference images. A neural network is used to learn the relationship between robot joints and image representations³⁴. A medical/surgical application of similar relevance⁵ presented a simulation system based on novel algorithms for animating instruments interacting with deformable tissue in real-time. The focus was on computing the deformation of a tissue subject to external forces, and detecting collisions among deformable and rigid objects. Lastly, Besseling and Gong² have developed a finite element method that relates the nodal coordinates to either a) the element deformations or b) the relative motion of elements with generalized strain functions. Their method was successfully used for the kinematic analysis, dynamic analysis and simulation of spatial mechanisms and manipulators.

Despite the innovation of the above-described research efforts, the authors have not identified any existing work, with sufficient visual realism and detail, that accomplishes the research goals discussed in the "Motivation" chapter. In response to this apparent deficiency, the next section presents a detailed description of the development of the NYSCEDII Virtual Motion Platform, or "NVMP".

The NYSCEDII Virtual Motion Platform (NVMP)

With background on hardware motion platforms, and the motivation to have a fully computer-based visual simulation of this type of hardware, this section proposes the design and development of the NVMP. Initially, **i)**, a brief discussion of the detailed CAD model is presented. Note that this CAD model is vital in that it serves as the visual groundwork for the NVMP, and further serves as the basis for all of the proposed research that follows. This discussion is followed by **ii)**, a presentation of the proposed features and operability of the eventual NVMP toolkit itself.

Development of the detailed CAD model

The ProEngineer utility³¹ has been used to create a detailed model of all solid objects that comprise the CAD model. Refer to Figure 3. Both the flying frame and base plate were modeled directly from the blueprint schematics provided by Moog²⁵. The pistons/cylinders and braces were measured by hand using a ruler and protractor, as exact dimensions are proprietary and were not made available to us. Modeling consisted of a series of standard protrusions, sweeps and cuts. Each of the major components were modeled separately, and then later combined in the “Assembly Modeler”. The motion platform itself was exported from the Assembler in four separate parts: a.) the base plate, which consisted of the bottom plate and the six lower braces, b.) the outer cylinders, c.) the inner pistons, and d.) the flying frame, which included the six upper braces. Each of these components was individually exported as a ProEngineer Render File (.slp). In this way, the component or group of components could be selected from the final assembly model and exported as triangular polygonal surfaces. A third party program, NuGraf (www.okino.com/nrs/nrs.htm), was then used to convert these ProEngineer render files into C++ code¹³ – specifically – a sequence of 3-D vertices and outward normal coordinates.

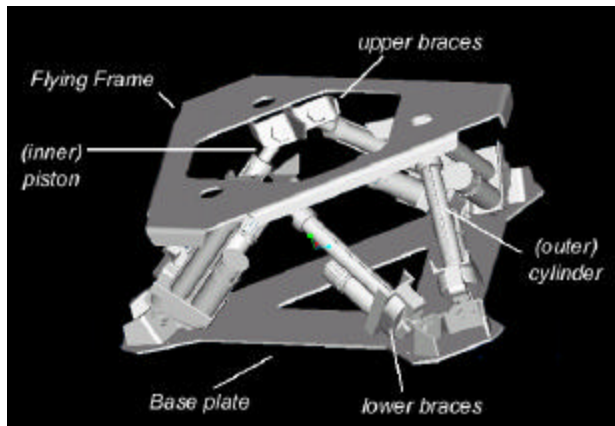


Figure 3: Detailed CAD model of the NVMP

By selecting the proper coordinate system for each of the components, the exported objects were all properly oriented – there was no need for auxiliary scaling or translation, once imported (as binary data) into the NVMP OpenGL³² simulation code.

With an understanding of how the detailed model of the virtual motion platform was constructed, the next subsection will discuss the proposed features and operability of the NVMP simulator, which makes extensive, time-varying use of this CAD model.

Features and Operability of the NVMP

Display and Navigation

As with most PC-based software utilities, the NVMP is a highly mouse-menu and keyboard-driven utility. The NVMP simulation contains numerous baseline *navigational* features that one would expect from any virtual simulator, namely rotational orbiting using the mouse and translation of the virtual platform using the mouse keys. The default view (Figure 4) is a top view of the NVMP, with the front of the platform pointing towards the right of the screen. To re-attain this view at any time, the user can invoke the “camera home” display option.

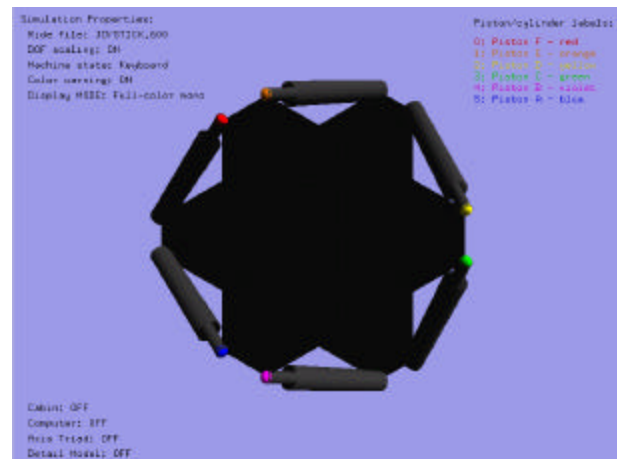


Figure 4: NVMP main window - default view

As can be seen in Figure 4, there are numerous corner screen display windows that convey simulation-relevant information to the user. The information in the lower left corner of the screen involves numerous “detail display” components that the user can toggle on/off at any time. These include the passenger “cabin” that sits atop the flying frame of the motion base, the model computer that sits on the lower frame, a reference axis triad, and the highly detailed CAD model of the motion platform hardware, described in subsection a). (None of these items are shown above – this allows for slightly faster graphics performance). The upper right corner of the screen contains technical information pertaining to the simulation itself. Shown above is the numbering and color scheme for the 6 pistons and cylinders – the color of the text corresponds to the colors of the reference spheres that are located at the base of each cylinder and at the top of each piston. The user can toggle this information to one of 3 other collections of information: the degrees of freedom INPUT (roll, pitch, yaw, in radians, and heave, surge, sway in meters) values for the system at any time frame, piston lengths (OUTPUT), and simulation time/graphics frame rate.

The upper left corner of the screen summarizes numerous properties of the simulation at hand, all of which will be discussed in forthcoming subsections of this chapter. These include: current ride file title, DOF scaling state, current machine state (keyboard, joystick, or ride file), color warning state, and display mode state. At any time, the user can access a “Help Menu” that outlines all of the mouse menu options and keystrokes required to fully operate the NVMP.

Piston lengths and DOF scaling

As mentioned previously, the NVMP tracks the outputs of the NVMP (the piston lengths) once presented with candidate inputs (values of degrees of freedom) to the motion platform. Each of the pistons on the physical motion base has inherent limits on its extents of motion. In the case of the Moog 2000E platform, each piston has a minimum length of 0.739 meters, and a maximum length of 1.019 meters. By default, the NVMP is programmed to not allow motion outside of this region. If the simulator is presented a set of inputs and the resulting outputs are such that one or more of the piston lengths falls outside of this range, the NVMP will, by default, “downscale” the input degrees of freedom minimally so as to produce acceptable outputs. This operation has to be performed quickly so as to maintain a 60 Hz minimum motion/display rate. This is presently accomplished using a very simple and heuristic “error minimization” model, that functions as follows. A dynamic move limit (dml) is defined:

$$dml = mlimit * cycno \quad (1)$$

where “mlimit” is the initial move limit, typically defined to be a small number (perhaps 1%), and “cycno” is the error minimization cycle number, initialized to 1. If, at a given time step, any of the piston lengths is deemed above or below its allowable limits, the following adjustment is made to 5 of the 6 DOF’s: roll, pitch, yaw, surge, sway:

$$dofnew = dofold * (1.0 - dml) \quad (2)$$

where “dofnew” is the newly computed DOF, and “dofold” is the original DOF value prior to augmentation. If one or more piston lengths is deemed below its allowable limits, an adjustment in the heave DOF also takes place. So, supposing an initial DOF value of 0.25 (a surge of 0.25 meters, perhaps) and an initial move limit of 1%, one can see the “pseudo linear” numerical trend in Figure 5. Small changes initially (i.e. we are trying to attain an acceptable motion with minimal change to our “desired” DOF values), are followed by a period of accelerated change (i.e. we have to work fast, as we are battling the clock

to maintain 60 Hz), which are then followed by a final period of slow change (i.e. nonlinear, infinite numerical degradation to zero). Initial experiments have found that given a “reasonable” range for allowable D.O.F. values (i.e. the Moog-published “software limits”²⁴ for the 2000E platform), rarely are more than 3-5 scaling cycles required to downscale the D.O.F.’s such that the piston lengths all fall within the allowable range.

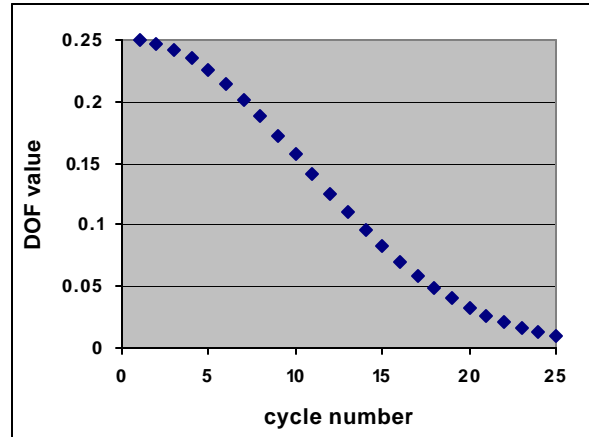


Figure 5: Numerical D.O.F. scaling by cycle

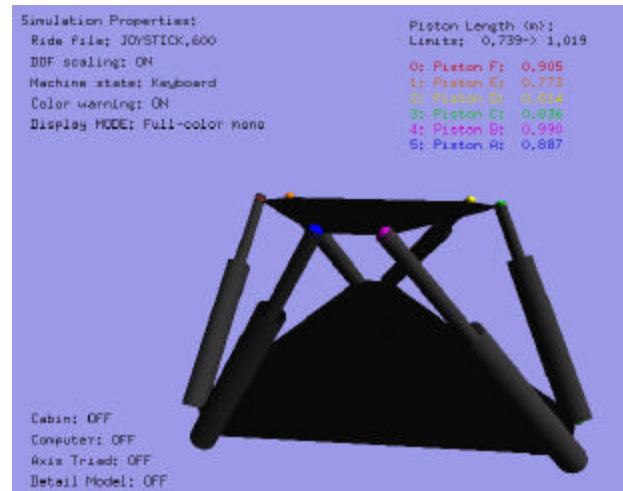


Figure 6: allowable piston lengths (blue)

The user can also instantiate another visual cue that will at least notify the user when one or more of the pistons travels outside of its allowable mechanical range, and thus when DOF scaling has been utilized to produce a “legal” motion frame. When turned on, the “color coding” option will still produce a motion with a piston or pistons traveling outside of its allowable range, but will supplement this illegal motion by changing the background of the NVMP from a default blue color to a “danger” red color. Figures 6 and 7 help to illustrate this feature. In Figure 6, all 6 pistons lie safely within the allowable safety zone; in Figure 7, one can see that

“Red” color-coded 0-Piston F has a length of 1.016 m, which has been internally downscaled, as described earlier, to be just below the 1.019 m piston limit.

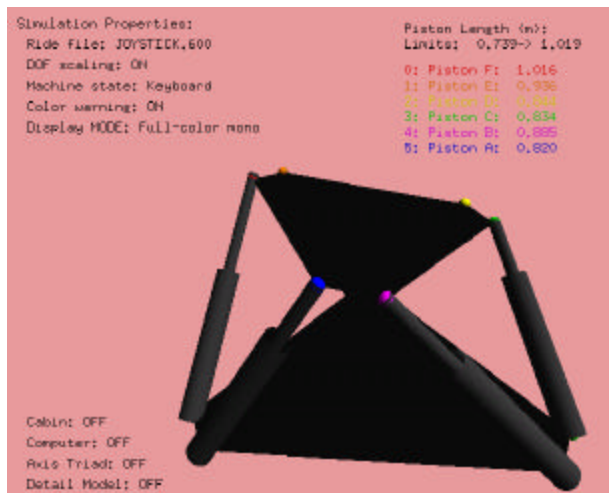


Figure 7: violated piston lengths (red)

Animation

Clearly, what will make a digital motion platform potentially useful is its ability to safely emulate dynamics - realistic, user-prescribed and user-interactive motion patterns. There are various mechanisms by which this will be accomplished, and these allude back to the “machine state” category first seen in the upper left corner of Figure 4. The default input mechanism will be the keyboard, through which the user is able to manually input individual D.O.F.’s through simple keyboard keystrokes – namely – the first letter of the desired D.O.F. For example, to surge the motion base forward by some user-prescribed incremental amount (0.05 meters, for example), the user will simply strike the “s” key, or the “S” key for reverse motion. Similarly, r/R are to be used for roll, p/P for pitch, y/Y for yaw, h/H for heave, and l/L for lateral (sway) motion. In “keyboard” machine state, the user is able to issue single click mouse commands to perform commonly needed “multiple piston motion” commands such as *proceed to neutral*, and *motion base park*. The former command moves the motion platform to a position where all 6 D.O.F.’s are in their intermediate position, and the latter command moves the platform to a position where all D.O.F.’s are in their resting position.

One can envision how the user may wish to interact with the NVMP in a more sophisticated fashion than through simple application of individual degree-of-freedom keyboard keystrokes. There are plans 2 additional user-selectable machine state modes to operate the simulation. The first is through the use of

the PC joystick port. By using previously developed, DirectX (DirectInput) based Windows function calls (www.microsoft.com/directx, “SDK On-line Documentation - Direct X for C/C++ - DirectInput module”) to the joystick port, the user is able to drive the virtual motion platform with devices ranging from a simple 2-axis joystick, to something more application tailored, such as a flight yoke for flight simulation, or a steering wheel and pedals for a driving simulator.

Finally, the NVMP allows for a third machine state of operation: “ride file input”. Moog motion platform users often require their hardware to perform pre-fabricated, non user-interactive ride motions repeatedly. These motions are stored in a binary ride file of prescribed format²³, where for each 1/60 second time increment, each of the 6 D.O.F.’s has a “double byte-based” scaled value ranging between 0 2^0 - 1) and 65535 (2^{16} - 1). Such ride files are certainly useful for verifying the correspondence between the physical hardware and the simulated motion of the NVMP, and are perhaps as common (if not more common) than user-interactive motion simulation applications in the field. Moog researchers have related to the authors that the generation of ride file motions is very cumbersome by conventional mechanisms, using the physical hardware itself. Certainly, an eventual use of the NVMP could be for the very generation of these “canned” ride excursion profiles.

Stereo viewing

To convey a sense of depth and immersion, the user will be to view the NVMP simulation in stereo. The visual simulation will be viewable in high-fidelity “active” stereo for expensive visualization display systems with auxiliary hardware (namely, a stereo emitter and high frequency shutter glasses). As a lower-cost alternative, the user will be able to view the simulation in stereo, irrespective of hardware sophistication, using a low-fidelity, 2-color anaglyphic stereo⁴. The generation of stereo images on the screen clearly requires 2 distinct images in the left and right OpenGL buffers. The apparent distance between these images on the screen is dictated by what is commonly known as Interocular distance, which is the distance between the left and right “eyes” of the viewer. This distance has a great deal to do with the success of the stereo effect, and the user will be able to interactively adjust this distance, up or down, using the “I” and “i” keys, respectively.

The next section discusses pertinent details regarding the science and engineering behind the proposed NVMP graphical simulation.

The Science behind the NVMP

This section presents a sampling of the calculations and programming sophistication that have been designed within the proposed simulator. Hence, the purpose of this section is to stress the concept of scientific visualization as it relates to this research effort. In other words, there are relevant, previously published and trusted mathematics and algorithms that underlie the computer graphics that are depicted in the NVMP. This reliable technical backbone will allow researchers to perform rapid synthesis of candidate motion profiles and perform “what if?” scenarios much more quickly, easily, and safely than could ever be performed using the physical hardware itself.

Inverse Kinematics

With forward kinematics¹⁹, one attempts to attain the outputs of a linked mechanical system (e.g. the position of the endpoint of a crane arm) based on a given set of inputs (e.g. the positions/angles of all joints that comprise the arm). With the NVMP, we are trying to solve the *inverse* kinematical problem - given a set of user-defined “outputs” (roll, pitch, yaw, heave, surge, sway) to the virtual motion platform at each time step – we wish to compute the desired “inputs” – the lengths of the individual 6 pistons/cylinders – these define the position/orientation of the entire system. The inverse kinematic problem will be solved here using a purely *geometrical* approach,^{7,9,11,16} and is outlined in the following narrative.

Consider a fixed coordinate frame (R) affixed to the base of the motion platform, and a moving coordinate frame (R') affixed to the top (“flying frame”) of the motion platform; refer to Figure 8. If the position of O' with respect to the origin of the fixed coordinate frame (O) is denoted by vector $[\mathbf{r}]_{\mathbf{R}} = [x_r, y_r, z_r]^T = [surge, sway, heave]$ in the Moog-defined reference frame (refer to Figures 1-2), we can define:

$$[\mathbf{b}_i]_{\mathbf{R}} = [\mathbf{r}]_{\mathbf{R}} + \mathbf{Q}[\mathbf{b}_i]_{\mathbf{R}'}, \quad i = 1...6 \quad (3)$$

Here, \mathbf{Q} is the rotation matrix describing the orientation of the platform (R') with respect to the fixed base (R), and is clearly a function of the user-specified *roll, pitch, and yaw*. If we subtract vector \mathbf{a}_i from both sides of the above equation, we get:

$$[\mathbf{b}_i - \mathbf{a}_i]_{\mathbf{R}} = [\mathbf{r}]_{\mathbf{R}} + \mathbf{Q}[\mathbf{b}_i]_{\mathbf{R}'} - [\mathbf{a}_i]_{\mathbf{R}}, \quad i = 1...6 \quad (4)$$

Clearly, the left hand side of the above equation represents the vector (along the i^{th} leg) connecting point A_i (fixed plane) to B_i (flying frame). By taking the *Euclidian norm* (of the above equation), i.e.:

$$\|\mathbf{x}\|_2 \triangleq \left(\sum_{i=1}^n |\mathbf{x}_i|^2 \right)^{1/2} \quad (5)$$

we get:

$$\rho_i = \left\| [\mathbf{b}_i - \mathbf{a}_i]_{\mathbf{R}} \right\| = \left\| [\mathbf{r}]_{\mathbf{R}} + \mathbf{Q}[\mathbf{b}_i]_{\mathbf{R}'} - [\mathbf{a}_i]_{\mathbf{R}} \right\| \quad i = 1...6 \quad (6)$$

In *scalar* form, this Euclidian norm can be re-written as follows:

$$\rho_i^2 = (x_r - u_i)^2 + (y_r - v_i)^2 + (z_r - w_i)^2, \quad i = 1...6 \quad (7)$$

where:

$$\begin{aligned} u_i &= x_{ai} - q_{11}x_{bi} - q_{12}y_{bi} - q_{13}z_{bi}, \quad i = 1...6 \\ v_i &= y_{ai} - q_{21}x_{bi} - q_{22}y_{bi} - q_{23}z_{bi}, \quad i = 1...6 \\ w_i &= z_{ai} - q_{31}x_{bi} - q_{32}y_{bi} - q_{33}z_{bi}, \quad i = 1...6 \end{aligned} \quad (8)$$

Here, q_{ij} stands for the (i,j) entry in the rotation matrix \mathbf{Q} , and the x,y,z components in the above equations relate back to equations (3) through (7), and Figure 11, as follows:

$$\begin{aligned} [\mathbf{a}_i]_{\mathbf{R}} &= [x_{ai}, y_{ai}, z_{ai}]^T, \quad i = 1...6 \\ [\mathbf{b}_i]_{\mathbf{R}'} &= [x_{bi}, y_{bi}, z_{bi}]^T, \quad i = 1...6 \end{aligned} \quad (9)$$

Clearly, all elements in the above equations are known and/or specified by the user: $[x_r, y_r, z_r]^T$ are the user specified surge, sway, and heave; the 9 q_{ij} matrix elements encapsulate the user specified roll, pitch, and yaw D.O.F.'s, and the (x,y,z) elements of equation (9) are the known Cartesian coordinates of the fixed and flying reference frames. With these quantities at hand, the user can thus, using equation (7), calculate the 6 piston lengths at each and every time step, hence describing the unambiguous position and orientation of the NVMP.

The next section will discuss details of a NVMP-physical motion platform comparison, which will attempt to validate and quantify the utility of the NVMP simulator.

Results and Discussion

With a simulation of this nature, it is difficult for the authors to present “results” in the conventional sense of the word. In essence, and for the intention of this paper – the simulation itself, and its features and functionalities, are themselves the “Results” of this research effort thus far. Nonetheless, the authors have attempted to supply some quantitative and pictorial measure of the preliminary performance of the NVMP. This section is decomposed into 2 parts: a) a

performance discussion of “pure” D.O.F.’s vs. “scaled” D.O.F.’s, and b) an image-based comparison of the NVMP vs. the actual hardware, for various configurations of a motion excursion.

“Pure” D.O.F.’s

When an individual D.O.F input is sent to the physical motion platform (full, 100% positive “pitch”, for example), the hardware performs a true, pure pitch for a small percentage of the excursion. When a certain range is achieved, this pitch is then augmented with a partial heave, thus making for a “non-pure” pitch, but a sustained motion nonetheless. Clearly, due to the constrained nature of the “6 legged” system, this augmentation allows the motion platform to extend its range of overall “pitch” beyond that which would be possible if only “pure” pitch were permitted (i.e. no heave whatsoever), and allows for a longer sustained motion. This process of augmentation is internal to the Moog hardware, the details of which are proprietary in nature. Hence, when creating the NVMP, rather than hypothesize as to the mechanics of this hardware-specific, proprietary augmentation, the authors decided to generalize the operability of the virtual simulator. As such, only “pure” non-augmented D.O.F’s were implemented, and error-minimization D.O.F. scaling (previously described) was used to numerically minimize any discrepancies between the resulting outputs and allowable piston length limitations.

Ride	Frames	Violated	Excess	Max	Avg
<i>Flight</i>	16767	5117	1	.40	.099
<i>Smooth</i>	13793	1636	0	.54	.189
<i>Daytona</i>	19067	664	0	.14	.065

Table 1: Ride file performance comparison

Table 1 presents a comparison of NVMP vs. physical simulator performance for 3 ride excursion files (each, approximately 45 minutes in duration) supplied to us by Moog. These files were generated for specific customers of their motion platforms, and thus the dynamic details of each are thus proprietary in nature. However, Moog was able to tell us that the first is an aggressive driving simulator, the second is a “slow and smooth” vehicle simulator, and the third is a Daytona race car driving simulator. The “frames” column is the total number of animation frames, each representing 1/60 second; the “violated” column lists the number of these frames that were violated by the NVMP (which again, unlike the hardware ONLY implements “pure” D.O.F. motion), requiring the use of D.O.F. scaling. The final two columns list the maximum and average dynamic move limit values required to scale the D.O.F.’s to within range (refer to equation (1)). The “excess” column lists the number of animation frames

that were violated, and could not be downscaled quickly enough within the allowable 1/60 second time step limit – indicating a “severe” violation.

Clearly, the “Flight” ride file required substantial augmentation for simulation using the NVMP. Roughly 1/3 of the total frames were found to have “violated” motion using pure D.O.F.’s, with 1 frame having an excess violation. The maximum move limit was found to be 0.4, with an average move limit of just lower than 0.1, including only the violated frames. The “Slow/smooth” ride file had substantially fewer violated frames, but a larger maximum move limit (0.54), and an average move limit that almost doubled that of the “flight” ride file. The “Daytona” ride file had by far the fewest number of “violated” frames, as well as the smallest maximum and average move limits.

NVMP vs. the Physical Simulator

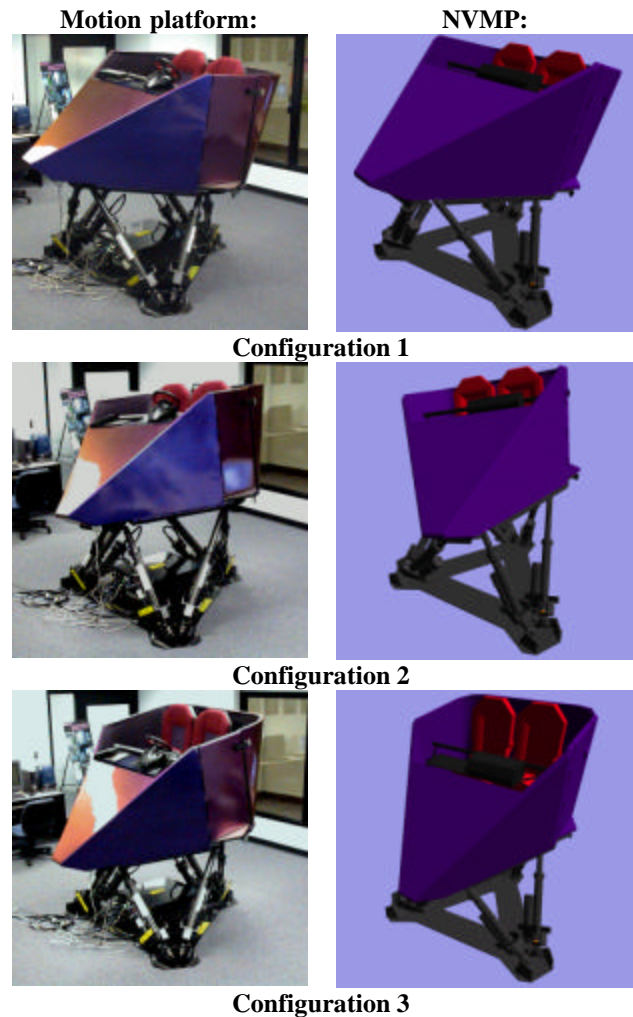


Figure 10: Frame comparison

From an isometric vantage point, the Figure 10 images attempt to qualitatively validate the realism of the NVMP simulation, both in terms of the actual appearance of the motion platform model, and in the output configuration of numerous common maneuvers. In the sequence of image pairs (NVMP-physical motion platform) are comparisons of 3 miscellaneous platform configurations. In each of these test frames, all six D.O.F.'s are non-zero, utilizing the full capability of the digital and physical simulators. Note that the primary discrepancies result from the difficulty of precisely matching the viewpoint and perspective between corresponding images. Further, recall from earlier discussion that these motion pairs will not be exact, due to the virtual motion platform exhibiting “pure” D.O.F.'s, for each motion, while the physical hardware exhibits some degree of minor augmentation. Table 2 lists the D.O.F. values for each of the three configurations shown. Note that the 2000E motion platform is shown in tandem with the passenger cabin in both the “NVMP” simulator view and the physical hardware view.

D.O.F.	Config #1	Config #2	Config #3
Roll (rad)	-.175	.131	-.087
Pitch (rad)	.035	-.105	-.340
Yaw (rad)	.148	-.218	.061
Heave (m)	.251	.260	.254
Surge (m)	.035	-.092	-.114
Sway (m)	-.013	.006	.035
Table 2: D.O.F. values for comparison frames			

Conclusions and Future Work

This paper has presented the design, development, and potential utility of a virtual prototyping utility called the NYSCEDII Virtual Motion Platform (NVMP). The NVMP simulation can be used to test candidate motions that are to be sent to the physical motion platform. This utility serves as a rapid and easy verification of the accuracy, validity, and/or “safety” of a candidate motion profile. Clearly, the expense of the physical motion platform hardware, and the safety of the platform passengers deem such a computer simulator a necessity. Once the candidate motion is validated by the NVMP, the motion profile can then be transferred to the actual motion platform for physical testing and experimentation.

For sake of realism, the NVMP has been detail modeled after the physical motion platform that resides at NYSCEDII, the 6 D.O.F. 2000E motion platform, donated by Moog, Inc. The NVMP boasts a variety of user-interactive features, including basic navigation (i.e. translation/rotation) using the mouse and keyboard, a

heuristic “error minimization” feature to scale excessive input D.O.F.'s, color-coding to notify the user as to when the max/min piston limits have been violated, animation functionality by way of ride-file input and/or joystick input, and an anaglyphic stereoscopic viewing functionality. The NVMP is a truly scientific visualization in that it incorporates literature trusted, previously published algorithms to define the simulator motions, by way of inverse kinematics. Further computational intelligence has been incorporated into the NVMP by way of the message passing paradigm that has been incorporated to accommodate simultaneous user-input, NVMP simulation, and if desired, corresponding scene graph modification and physical motion platform function.

The possibilities of useful, and as of yet unincorporated “future” features into the NVMP are limitless. The authors have numerous ideas for features that would be quite easily incorporated into the NVMP, and have received pertinent suggestions from various designers, engineers, researchers, and software developers who regularly work with the physical hardware. These planned functionalities are listed below.

Expanded D.O.F. washout

One of the primary difficulties with practical motion simulation is achieving the effect of continuous linear acceleration. The basic trick relies upon providing an initial acceleration along the proper vector, and then decelerating at a level below the threshold of perception, so that you can return the platform to its initial state as quickly as possible. This is necessary, since at the limits of excursion of the platform no additional motion cues can be generated. Washout algorithms are implemented for moving the platform back to its starting position, while still providing a good illusion of acceleration. Furthermore, these algorithms must be tuned to the given application – be it a flight simulator²⁸, or a driving simulator²⁹, or other. The “D.O.F. Scaling” technique proposed earlier serves as a preliminary mechanism for guaranteeing NVMP simulator motion to lie within the published hardware limits, and can be considered to be a very primitive form of motion “washout”. Downstream, the authors foresee the incorporation of a variety of more sophisticated washout techniques into this simulation toolkit to improve upon the existing (and admittedly simplified) heuristic error minimization scheme that was previously discussed, and which will be implemented as the lone preliminary “washout” mechanism. Our team envisions a menu-driven, modular functionality whereby NVMP users can make use of any of a number of previously published washout algorithms, or they can incorporate their own (self-

designed) algorithms if they so choose. This expanded mechanism for D.O.F. washout will be incorporated during the later stages of the proposed period of research.

“Picture in picture” simulation viewing

Our research team further envisions a “picture in picture” feature for the NVMP toolkit, which will not only show the user-controlled movement of the computer motion platform itself, but will show other dynamic visuals relevant to a motion platform programmer. Namely, the simulation will display the corresponding motion of an accompanying scene graph, shown on a small portion of the computer screen. The “picture in picture” feature would be very useful when attempting to tailor the motion of the virtual platform to some pre-defined scene graph motion. A related functionality will be the ability for the user to generate and simultaneously capture a motion profile for future replay – i.e., the user dictated motion is written to a binary data file in real-time.

Simulation data charts

Finally, the research team foresees numerous on-screen plotting functionalities that will augment the visuals with engineering data of practical utility. The team will incorporate a menu-driven capability for the user to plot input (D.O.F. values) and output (piston lengths) in real-time. The functionality will be “corner screen” by default, or the user can toggle the plotting to full screen for greater detail, interactively. A longer-range project goal will be to incorporate charting of the dynamics of the virtual platform, i.e. the velocity and acceleration of the moving pistons, and/or the flying frame itself. With masses of the motion platform hardware known, the NVMP could eventually plot forces and torques present during a motion excursion, which would certainly be of interest when attempting to assess rider safety for a candidate motion profile.

Acknowledgements

The authors would like to acknowledge the partial funding support of Sheldon Silver and the New York State Assembly for their support of NYSCEDII, the research organization for which the present work was performed. The author’s would also like to thank the East Aurora branch of Moog, Inc., who donated the 6 D.O.F. motion platform to NYSCEDII. It has been our research with the Moog platform that inspired the need for the research presented in this paper.

References

- [1] Advani, S.K, Nahon, M.A., Haeck, N., and Albronda, J., “Optimization of Six-Degrees-of-Freedom Motion Systems for Flight Simulators.” *Journal of Aircraft*, Vol. 36, No.5, September-October, 1999.
- [2] Besseling, J.F., and Gong, D.G., “Numerical simulation of spatial mechanisms and manipulators with flexible links.” *Finite Elements in Analysis and Design*, Volume 18, Issues 1-3, December 1994, Pages 121-128.
- [3] Bonev, I.A., and Ryu, J., “A geometrical method for computing the constant-orientation workspace of 6-PRRS parallel manipulators.” *Mechanism and Machine Theory*, Volume 36, Issue 1, 1 January 2001, Pages 1-13.
- [4] Bourke, P., “Creating Anaglyphs Using OpenGL.” On-line web reference: <http://astronomy.swin.edu.au/~pbourke/opengl/redblue/>, August, 2000.
- [5] Brown, J., Sorkin, S., Latombe, J-C., Montgomery, K., and Stephanides, M., “Algorithmic tools for real-time microsurgery simulation.” *Medical Image Analysis*, Volume 6, Issue 3, September 2002, Pages 289-300.
- [6] Fichter, E.F., “A Stewart platform-based manipulator.” *International Journal of Robotic Research*, 5(2): 157-181, 1968.
- [7] Gosselin, C., “Determination of the Workspace of 6-DOF Parallel Manipulators.” *Journal of Mechanical Design*, Vol. 112/331, September, 1990.
- [8] Gosselin, C., Laverdière, S. and Côté, J., “SIMPA: A graphical simulator for the CAD of parallel manipulators.” *Proceedings of the ASME International Computers in Engineering Conference*, San Francisco, Vol. 1, 1992, pp. 465-471.
- [9] Gosselin, C., “Parallel computational algorithms for the kinematics and dynamics of parallel manipulators.” *Proceedings of International Conference on Robotics and Automation*, Volume 1, pages 883-888, 1994.
- [10] Grant, P.R. and Reid, L.D., “Motion Washout Filter Tuning: Rules and Requirements.” *Journal of Aircraft*, Vol. 34, No. 2, March-April, 1997.

- [11] Hunt, K.H., "Kinematic geometry of mechanisms." Oxford University Press, Oxford, 1978.
- [12] Idan, M. and Nahon, M., "Off-Line Comparison of Classical and Robust Flight Simulator Motion Control." *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 5, September-October, 1999.
- [13] Kernighan, B., and Ritchie, D., "The C Programming Language, 2nd Edition." Prentice Hall, Englewood Cliffs, NJ, 1988.
- [14] Lee, J., Albus, J., Dagalakis, N., and Tsai, T., "Computer simulations of a parallel link manipulator." *Robotics and Computer Integrated Manufacturing*, August 1988.
- [15] Lazard, D., and Merlet, J-P., "The (true) stewart platform has 12 configurations." *IEEE Int. Conf. on Robotics and Automation*, pages 2160-2165, San Diego, May 8-13 1994.
- [16] Liu, K., Fitzgerald, J.M., and Lewis, F.L., "Kinematic analysis of the Stewart platform." *IEEE Transactions on Industrial Electronics*, 40(2): 282--293, 1993.
- [17] Lu, Y., "Using CAD functionalities for the kinematics analysis of spatial parallel manipulators with 3-, 4-, 5-, 6-linearly driven limbs." *Mechanism and Machine Theory*, Volume 39, Issue 1, January 2004, Pages 41-60.
- [18] Merlet, J.P., "Parallel Manipulators, Part I: Theory, design, kinematics, dynamics and control." Technical Report #646 INRIA, France, 1987.
- [19] Merlet, J.P., "An algorithm for the forward kinematics of parallel manipulators." 91 ICAR, Fifth International Conference on Advanced Robotics, Volume 2, pp. 1136-40, 1991.
- [20] Merlet, J-P., "Orientation workspace of a parallel manipulator with a fixed point." ICAR, pages 141-146, Tokyo, November 1-2, 1993.
- [21] Merlet J-P., "Trajectory verification in the workspace for parallel manipulators." *The International Journal of Robotics Research*, 13(4): 326-333, August 1994.
- [22] Merlet, J-P., "Designing a parallel robot for a specific workspace." Research Report 2527, INRIA, April 1995.
- [23] (Moog1), "Moog 6 DOF 2000E Motion Base Joystick File Format." Addendum to Motion System User's Manual, July, 1999.
- [24] (Moog2), "Moog 6 DOF 2000E Motion System Interface Definition Manual." Part No.: c37968, Doc No.: LSF-0446, November, 1999.
- [25] (Moog3), "Moog 6 DOF 2000E Motion Mechanics Installation Manual." Part No.: c12643, February, 2000.
- [26] (Moog4), "Moog 6 DOF 2000E Motion System User's Manual." Part No.: c37970, Doc No.: LSF-0462, April, 2000.
- [27] Parrish, R.V., Dieudonne, J.E., and Bowles, R.L., "Coordinated Adaptive Washout for Motion Simulators." *Journal of Aircraft*, Vol. 12, No. 1, pp. 44-50, 1975.
- [28] Reid, L.D. and Nahon, M.A., "Response of airline pilots to variations in flight simulator motion algorithms." *AIAA Journal of Aircraft*, Vol. 25, No. 7, pp. 639-646, 1988.
- [29] Reid, L.D. and Grant, P.R., "Motion Algorithm for Large-Displacement Driving Simulator." *Transportation Research Record*, No. 1403, pp 98-106, Mar. 1993.
- [30] Stewart, D., "A platform with six degrees of freedom." *The Institution of Mechanical Engineers*, 180(15): 371--384, 1965.
- [31] Toogood, R. and Zecher, J., "Pro/Engineer Tutorial Release 2000i2." SDC Publications, Mission, KS, May, 2000.
- [32] Woo, M., Neider, J., Davis, T., and Shreiner, D., "OpenGL Programming Guide, Third Edition." Addison-Wesley Publishing, Reading, MA, 2000.
- [33] Yia, L. and Leinonen, T., "A computer simulation approach to machining complicated shapes, on an orthogonal 6-rod machine tool." *International Journal of Machine Tools and Manufacture*, Volume 42, Issue 4, March 2002, Pages 441-447.
- [34] Zhao, Q., and Sun, Z., "Image-based robot motion simulation." *Optics Communications*, Volume 205, Issues 4-6, 1 May 2002, Pages 257-263.