

**Graphics  
Research and  
Analysis  
Facility**

**Technical  
Report**

**Validation of a Human Body Dynamics  
Computation using SD/FAST**

Andrew Cheang, Kristian Mueller, Abhilash Pandya

Graphics Research and Analysis Facility  
Lockheed Martin  
2400 NASA Road 1  
Houston, Texas 77058  
Phone (281) 483-3728

**August 1998**



National Aeronautics and  
Space Administration

Scientific and Technical  
Information Branch

SPACE HUMAN FACTORS

# Table of Contents

<b>TABLE OF CONTENTS</b> .....	<b>I</b>
<b>ABBREVIATIONS &amp; ACRONYMS</b> .....	<b>II</b>
<b>ABSTRACT</b> .....	<b>III</b>
<b>1.0 INTRODUCTION</b> .....	<b>1</b>
1.1 OVERALL GOAL .....	1
1.2 PREVIOUS WORK .....	1
1.3 SCOPE OF RESEARCH .....	1
1.4 SD/FAST .....	2
1.4.1 <i>Summary of SD/FAST routines used</i> .....	2
1.4.2 <i>Program Flow</i> .....	3
<b>2.0 METHODOLOGY</b> .....	<b>3</b>
2.1 BUILDING THE SD-FILE .....	3
2.2 EQUATIONS OF MOTION GENERATION WITH SD/FAST .....	4
2.3 SIMULATION OF HUMAN BODY DYNAMICS .....	4
2.4 CREATING THE INTERFACE .....	5
2.5 VALIDATION DATA COLLECTION .....	5
2.5.1 <i>Apparatus</i> .....	5
2.5.2 <i>Data collection methodology</i> .....	6
2.6 COMPARISON OF SIMULATION DATA WITH COLLECTED DATA .....	6
<b>3.0 RESULTS</b> .....	<b>7</b>
3.1 FIRST CASE STUDY .....	7
3.2 SECOND CASE STUDY .....	7
<b>4.0 CONCLUSIONS</b> .....	<b>8</b>
<b>5.0 FUTURE DIRECTIONS</b> .....	<b>8</b>
<b>6.0 REFERENCES</b> .....	<b>9</b>
<b>7.0 ACKNOWLEDGEMENT</b> .....	<b>9</b>
<b>8.0 APPENDIX</b> .....	<b>10</b>

## **Abbreviations & Acronyms**

**ABF** - Anthropometry and Biomechanics Facility

**DOF** - Degree Of Freedom

**EVA** - Extra Vehicular Activity

**GRAF** - Graphics Research and Analysis Facility

**JSC** - (Lyndon Baines) Johnson Space Center

**NASA** - National Aeronautics and Space Administration

**SAR** - Simplified Analysis Routines

**STS** - Space Transportation System

## **Abstract**

The goal of this project was to create a task-oriented human model which can be used to emulate the physical characteristics of the human body. In the development process, equations of motion were generated using the software program called SD/FAST. Simulation routines, which are also provided by this program, were used in conjunction with user supplied analysis routines to study the load propagation and dynamic behavior of the human model. Two validation studies were conducted, and the data generated by the program was compared against actual test data. The results of this study indicate that dynamic modeling of the human body with SD/FAST produces reasonable results when compared with actual data. From two separate test case studies between actual data and simulated data, the overall average percentage error was 20%.

## **1.0 Introduction**

Due to the shortcomings and cost of current micro-gravity simulators, planners, engineers, astronauts and researchers who are concerned with EVA activities will have to rely on computer models of humans and their environments as one of the tools to assist them. Currently, physical simulations are used to evaluate EVA methods and procedures. These earth-based simulations are time consuming and expensive to build, and have obvious limitations. For instance, in the Neutral Buoyancy Laboratory, gravity and the drag coefficients of the water are both critical deficiencies. The failure of the initial Intelsat capture attempt (STS 49) is a glaring example of the limitations of physical simulators. As the number and duration of EVAs increase, the amount of training time spent for each hour of EVA will have to be optimized. There is a need for better methods of evaluation. The validation performed in this paper is a first step towards building a computer-human model of EVAs. Scenarios ranging from simple activities to complex tasks could be visualized, analyzed, reviewed, and modified in conjunction with present-day micro-gravity simulators.

### **1.1 Overall goal**

The overall goal of this study is to predict human body dynamics such as mass handling properties and joint fatigue occurrences. By modeling case studies using SD/FAST, such analyses as joint fatiguing and maximum and minimum torque requirements can be made available before the studies are actually carried out. Ergo, this project was conducted to validate the effectiveness of the software package SD/FAST. Furthermore, driver code was created that can be used for general analyses that encompass a wide range of test cases.

### **1.2 Previous Work**

In 1995, Schaffner and Newman at MIT presented a methodology for analyses of an EVA crewmember using multi-body dynamics [1] and an equation of motion generation system called SD/FAST [2]. They have simulated a 7 degrees-of-freedom (DOF) planar model of the Extra Mobility Unit (EMU) and successfully computed the inverse dynamics solution of an astronaut moving a satellite in a plane. Their inverse dynamics model computed torques at each joint axis given the motion of the end-effector. This data was compared to measured maximum torques [5] at each joint to create an EVA simulation. Ko Badler [3] at the University of Pennsylvania in 1996 presented a paper on the modeling of locomotion of humans with inverse dynamics using a similar approach. His dynamics model used balance control and a kinematics walk generator to produce gait motion. Based on this motion, torques were computed at each joint of the leg using inverse dynamics. The computed torques were compared to biomechanical data to feed back to the system joint strength violations. This produced dynamically sound gait motion with various loads applied to the human model. Phillips *et al.* [38] have done a similar evaluation of human reaching and motion which is strength guided and based on a feedback system using strength data. It is worth noting that all of the above authors (Phillips, Ko, and Schaffner) used strength data produced at the GRAF from previous code UL sponsored work.

### **1.3 Scope of Research**

The objective of this research is to develop a validated tool which will provide for human-body dynamics analysis capabilities at the GRAF. The model will be validated against actual subject data.

Biomechanics is the subject of forces and their effects on the human body. It relates the forces in muscles, bones, and joints with externally applied loads. Biomechanics deals with motion and forces which result from muscular contractions and flexions or externally applied forces. Such loads or forces may result from the pull of gravity on the body parts, or from resistance which is applied to the body segment.

## 1.4 SD/FAST

SD/FAST is a dynamics software package designed by Symbolic Dynamics that generates the equations of motions for a system of joint connected rigid bodies. Examples of such systems include mechanisms and machines, ground vehicles, gear train, articulated spacecraft, manipulators, cables, and high speed electromechanical devices, suspension components, multiple-arm robots, anti-lock brakes, helicopters, walking/hopping machines, farm machinery, construction equipment, precision pointing mechanisms, momentum wheels, and biomechanics. Furthermore, SD/FAST aids in the modeling of forces such as dampers, gravity, springs, motors and engines, feedback control (space craft stabilization system), feedforward control (remote manipulator system), gas pressure, tires, hydraulic forces, aerodynamics forces, bushings, vibration inputs, empirical data, actuators, gravity gradient, stochastic loads, and friction. SD/FAST can have prescribed positions for the system joints: e.g. locked joints, constant-speed motors, road profiles, orbital motion, sinusoidal motion, and cam follower. The forces required to implement this prescribed motion are calculated and returned by SD/FAST. SD/FAST allows constraints to restrict motion such as: gears, tracks, belt and pulleys, screw joints, complex cams, roll-without-slip, pin-in-slot constraint, rack and pinion. SD/FAST will calculate the reaction loads required to implement these constraints. Types of analyses and design studies include: assembly analysis, velocity (initial conditions) analysis, forward and inverse dynamics, statics and steady motion, limit cycle identification, mechanical advantage, power flow, design sensitivity, design optimization (with external optimizer), real-time hardware-in-the-loop. SD/FAST uses a symbolic dynamics formulation. SD/FAST generates code that contains the simplified equations of motion specific to that system much as would be generated by hand. These equations of motion are simplified to their most efficient form thus allowing a faster run time. These simplified analyses routines (SAR) used the more general dynamics routines. Because of time constraints, the SAR files were used for the validation tests.

System parameters can be hardwired into the input file, thus allowing SDFAST to optimize the equations of motions by means of symbolic dynamics. If a user wishes to change the parameters of a system he simply leaves a question mark (?) in the parameter slot of the input file of SDFAST, thereby allowing variable parameters.

### 1.4.1 Summary of SD/FAST routines used

The driving program (excluding the interface) is comprised of four main files. One file contains the main subroutine (usually named the *proj.c* file). The *initial.c* file contains the calls to *sdassembly* and *sdintvel*. In the *forward.c* file is the call *sdmotion*. And finally in *sdu.c* are the user defined functions, *sduforce* and *sdumotion*, which must be written whenever SARs are used.

*proj.c* - System parameters are set using the following subroutines.

- *sdmass* - defines in three directions the gravity
- *sdiner* - defines the 3x3 body inertia matrix
- *sdbtj* - defines the body segments relative position to joint to the inboard body
- *sbitj* - defines the inboard body position to the joint to the current body segment
- *sdinit* - sets system parameters
- *sdpres* - turns on/off prescribed motion for individual body segments

*sdu.c* - Within *sdumotion*, each axis of each body segment may have its own prescribed motion. The prescribed motion consists of functions of time, position, and acceleration. Once an axis has been allowed to be prescribed (*sdpres*), the actual functions to prescribe the motion are *sdprespos*, *sdpresvel*, and *sdpresacc*. Forces may be applied to a body segment inside the *sduforce* subroutine. Forces are defined as three vectors defined either in the body or local frame. They are applied to a point by the *sdpointf* function. Furthermore, hinge torques may be applied to any joint that has any rotational axis (*sdhinet*). Lastly body torques may be applied to a segment in all three directions (*sdbodyt*).

*initial.c*- The SD configuration file is usually written with all body segment axes written in the same orientation as the ground reference frame. However, during analysis, this may not be the ideal initial configuration. Therefore, after initial positions have been set, a call to *sdassemble* is made to find a

configuration that conforms to restraints while maintaining the initial positions. If a segment is given an initial position and the user wishes to maintain that initial position, it may be locked. This lock will result in that body maintaining its initial position throughout the assembly process.

When using user define prescribed motion, a body usually must have an initial velocity. Once these initial velocities have been set and locked, an initial velocity analysis must be made in order to confirm that the body does not defy any constraints.

forward.c - Herein lies the bulk of the computational analysis of the program. The function *sdmotion* uses a variable-timestep integrator to calculate the motion in the timestep *dt* from time to time+*dt*. The number of steps can be specified in *numtime*. The first call to *sdmotion* initializes all internal data structures so that the routines to be called do not depend not the current values of the derived state.

The function call to *sdreac* returns all reaction forces and torques acting on the segments of the body.

## 1.4.2 Program flow

- 1) set variable system parameters: *sdmass()*, *sdpin()*, etc.
- 2) initialize system: *sdinit()*
- 3) close loops by adjusting q's: *sdassemble()*
- 4) compute compatible velocities (u's): *sdinitvel()*
- 5) perform static or steady analyses if needed: *sdstatic()*, *sdsteady()*
- 6) perform motion analysis if needed for each communication interval:
- 7) integrate equations across interval: *sdmotion()*, *sdfmotion()*
- 8) compute system momentum, energy, mass: *sdmom()*, *sdsys()*
- 9) compute reaction forces: *sdreac()*
- 10) output quantities of interest
- 11) check for usage errors: *sdprinterr()*

A diagram of the program flow is found in the Appendix. See Figure 1.

## 2.0 Methodology

The process of developing an SD/FAST graphics utilization interface, GUI, that lets the user specify a set of analyses that are to be performed with the model, started with the creation of an SD/FAST System Description File referred to as the SD-file.

### 2.1 Building the SD-file

This file describes the geometrical dimensions as well as the relative positions of each body segment. It also specifies the inertia matrix as well as the mass of each segment and the global gravitational force. Additionally, a name was given to each body segment, and the type of joint that it is connected to was specified. All properties except for the mass are specified in vector format. See the appendix for a copy of the SD/FAST file.

The masses were taken from table 1 of the NASA document entitled "Weight, Volume, and Center of Mass of Segments of the Human Body" [1, p2]. The geometrical dimensions were taken from a technical report from the U.S. Department of Defense [2, pp 32-52,58-60,66-69,74-77]. The geometry of the model is specified in the SD-file with vectors that locate the centers of mass of a segment and its adjoining body to the joint that connects the two segments. Furthermore, the principal moments of inertia, obtained from the same pages of source #2 as the geometrical dimensions, were used to construct the inertia matrix with the products of inertia being all zero. The principal moments were used in order to reduce the complexity of the equations of motion and, hence, reduce computation time. This was possible because the global coordinate axes coincided with the principal moments of inertia of each segment at the time of assembly. Finally, the gravitation was prescribed to act globally on the entire model in the positive x-direction, i.e. the direction from head to feet.

The final SD-file represents the system model as a linked chain of rigid body parts. Each body segment is connected to another by either a one-degree of freedom (DOF) pin-joint, a three DOF gimbal joint or a six DOF bushing joint. The type of linkage chosen between two segments depended upon the physical characteristics of the corresponding human joint. For the elbow and shoulder, for instance, one and six DOF joints were chosen, respectively.

The full SD file can be found in the Appendix.

## **2.2 Equations of Motion Generation with SD/FAST**

The completed SD-file was then run through the SD/FAST compiler, which in return produced three C files as mentioned in the above section on SD/FAST. In the interest of producing a refined jet plain model in a short time frame, only simplified analysis routines were used for this human model analysis.

## **2.3 Simulation of Human Body Dynamics**

The dynamic model created by SD/FAST, together with the simplified analysis routines are called upon by the driver code that comprises the quintessence of the simulation environment. The objective was to create flexible C code that could eventually be commanded through an easy to use user interface to perform general dynamic analyses. In order to develop a generic algorithm, however, specific simulations were created first such that similarities could be gathered and reapplied to a comprehensive code. Specifically, three distinct simulations were created:

- 1) A model that performs adduction and abduction motions with the right upper arm
- 2) A model that raises and lowers the left and right arms sequentially

The fundamental structure for all three simulations is identical. It consists of the program structure defined in Section 1.4.

The main C code organizes the driver code and calls the necessary subroutines in the proper order. The following list summarizes the program structure:

- Change SD-file variables if necessary
- Initialize the state array and integration time to zero
- Call the model initialization routine *sdinitconds*
- Check for position and velocity errors
- Turn prescribed motion on/off
- Call the dynamic equation integration routine *runfwd*
- Zero out velocities if performing a second run
- Go back to bullet three

After the main program was written, the subroutines were composed. First, the model had to be assembled. Thus, an initialization routine was written that assembles the model to its initial configuration, but also initializes the starting hinge velocities of each segment of the model. Its algorithm is as follows:

- Initialize positions for assembly
- Lock/unlock joint rotation and translations for assembly
- Set the tolerance and maximum number of calls
- Call the SD/FAST supplied assemble routine *sdassemble*
- Initialize velocities for velocity analysis



- Lock/unlock joint velocities for velocity analysis
- Call the SD/FAST supplied initial velocity analysis routine *sdinitvel*

Next, the models forward and inverse kinematics was implemented. This was done with a subroutine that applies external forces to the model and/or prescribes the motion of one or more body segments. Forces are modeled in the body local frame as vectors in three-space. Each vector component is represented with a sixth degree polynomial as a function of time that can be applied as a point force on any body segment at any location. Forces are applied to the model through an SD/FAST supplied point force routine called *sdpointf*. Additionally, an inverse kinematics routine can be implemented by prescribing the joint angles as a function of time.

## **2.4 Creating the Interface**

The graphical interface was designed to facilitate the use of the SD/FAST program. The code for the actual interface is found in *sdfast.c* and *sdfast.h*. Interface control is also found in *case.c*, *case2.c*, and *trial.c*. The design of the interface was aided by using the *fdesign* .

The user can change system parameters such as the properties of the individual bodies as well as prescribe any forces, torques, or motion to the individual body segments. The forces, torques, and motion equations are able to accept up to a sixth order polynomial equation. For now, the equations must be functions of time. If the user wishes to have these equations as functions of other variable other than time, the equations must be put in the actual c program. The interface also allows the user to choose which segments information should be printed for.

The top form shows the individual body segments that can be manipulated. Also, there is an input blank for the gravity. Including the *misc* button, all buttons lead to a new form.

The *misc* form contains tolerance values for both initial analyses (*sdassemble* and *sdinitvel*) as well as tolerance values for *sdmotion*. Also the number of times that *sdmotion* will run and what step size can be adjusted in this form.

In order to do multi-run analysis and to avoid starting over again from scratch, the interface had to have a method that saves the system and user-set parameters from the previous run. The *RUN3.out* file is printed at the end of each run (*print\_info*). This contains all the data for the form. Through a number of subroutines, the data is read from the file (*read\_info*) and transferred to another set of variables (*put\_in*). These variables are then used to input the data to the form. The form sets the values of the interface. As the interface is changed, the data is updated in the form variables. After the form has finished, the data is passed back to the program variables for use (*get\_out*).

The top level interface is shown in Figure 2 of the Appendix.

## **2.5 Validation Data Collection.**

To validate the SD/FAST software, two studies were conducted in the Anthropometry and Biomechanics Facility of the Flight Crew Support Division at the Johnson Space Center, Houston Texas. The apparatus and data collection methodology are presented below.

### **2.5.1 Apparatus**

The equipment used for the data collection of the two validation studies consisted of a dynamometer and a force plate. Loredan Biomedical, Inc. built the dynamometer called LIDO, that was used in the first experiment. The force plate, build by Advanced Mechanical Technology, Inc. (AMTI), measures forces and torques in three mutually perpendicular axes; it was used in both experiments.

The force plate data was modulated by an AMTI amplifier and transmitted via an RG174 coaxial cable to a lap-top computer. The LIDO data was amplified through the LIDO controller and relayed to the same computer. The data processing for both experiments was performed with a commercially available software package called Lab View. This software package expressed all forces and moments in pounds and foot-pounds, respectively.

## 2.5.2 Data collection methodology.

The first study consisted of a load-transfer analysis of a person performing an adduction and abduction motion with the right arm on a Loredan Biomedical, Inc. LIDO Active machine (see Figure 3). The test subject was asked to stand on top of the AMTI force plate. While the LIDO arm moved at a constant rate of 60 degrees per second, the subject applied an action force to it, opposite to its direction of motion. During the experiment, the force plate registered the reaction forces and moments imparted to it by the subject's feet as a function of time. Furthermore, the force supplied by the subject was measured as a torque about the shaft of the LIDO arm; again as a function of time.

The second test case was similar to the first. Instead of applying a force to the LIDO, however, free-weights supplied the action forces while the subject performed isokinetic movements. The test subject, holding a 10 pound test mass in each hand, stepped onto the force plate as before. Next, the subject performed successive left and right arm-raises by rotating the shoulder joints exclusively: Starting with both arms hanging straight down, the subject raised the left-hand weight with a constant velocity to an angle of 90 degrees in front of his body while keeping his elbow position locked. The weight was then lowered to its original position at the same pace of 45 degrees per second. The motion of the right arm was similar to that of the left. Instead of raising and lowering to the front of the body, however, the weight was raised out to the right.

Next, the torque and force data was plotted against time and compared to the data obtained from the SD/FAST software.

## 2.6 *Comparison of Simulation Data with Collected Data.*

Having attained the data from the ABF, the same motions of the two studies were simulated using the SD/FAST routines.

For the first test, a two-step method was incorporated to simulate the movement of the left arm. First, a rocker arm was modeled in the SD/FAST model. This rocker arm was connected to the ground by a pin joint. The right hand was attached to this rocker arm by a weld. The rocker was then given a constant velocity. The right arm was not prescribed and moved according to the rocker. Furthermore, a point force was applied to the right hand in the opposite direction that the hand was moving. This force was found from the LIDO data. The angles for each segment of the right arm were output. The output was then taken and a curve fit was applied to the data. Next, the equation for the curve fit was applied and prescribed as the motion for the right arm segments. For the second step, another SD/FAST file was generated, this time without the rocker in the model. The arm was prescribed to the equations found previously, and again the force was applied to the right hand. The resulting loads were plotted against time for the SD/FAST generated data as well as the collected data. Furthermore, error analyses were conducted by calculating the percent error between each data point generated by SD/FAST and the averaged collected data.

The second study was modeled in a similar way. The SD/FAST model rotated its shoulder to the desired final angle and then lowered it again. Then the right shoulder was rotated up and down. After further analysis, it was discovered that while raising the weights, the test subjects swayed slightly at the hip. The ideal SD/FAST model did not sway while rotating the shoulders. This led to an inconsistency in the reaction torques. Therefore, the SD/FAST model was prescribed to rotate at the hip as well. Again, plots of all the loads were generated, and error analyses were conducted for the predominant torques and forces.

## **3.0 Results**

The two validation studies show that in general, the SD/FAST generated data predicts the actual data to within an uncertainty of 20%. However, the errors are dependent upon the specific analysis under consideration. Furthermore, the error analysis for both studies were conducted for the dominant forces and moments only, assuming that the remaining loads are small enough such that even large errors do not produce a significant change in the dynamics of the system.

### **3.1 First Case Study**

Figures 4 and 5 show the force-plate forces and moments that were collected in the ABF. It is seen that the dominant force acts in the positive x-direction with a magnitude of about 700 Newton; this is due to the gravitational pull on the model's mass. The slight increase and decrease of that force at the beginning and end of the adduction and abduction exercises, respectively, is due to the change in direction of the force applied to the LIDO. In addition, Figure 5 shows the moments about the system's x- and y-axis to be approximately one order of magnitude smaller than the z-moment. For the adduction exercise, the z-moment is at a constant 160 Newton meters, and for the abduction exercise, the moment acts in the opposite direction at about the same magnitude.

Looking at Figures 6 and 7, it can be observed that the data generated with SD/FAST and the collected data have a high correlation: Similar trends are seen in both sets of forces and moments. The error analyses of the dominant force and moment suggest that there is an average error of 3% for the forces, and an error of less than 7% for the moments (see Figures 8 and 9).

### **3.2 Second Case Study**

Figures 10 and 11 show the forces and moments that act on the forceplate during the second validation study. The only force that is apparent is the in the x-axis. This is due to the weight of the subject. And again in Figure 11, it is clear that the z-axis moment is dominant when the arm is extended out the side of the body. When the arm is raised out the front, the dominant torque occurs in the y-axis. Comparison with the SD/FAST generated simulation graphs of forces and torques shows an agreement in the general trend. Figures 12 and 13 show that the x-axis is indeed the dominant force throughout the study, as expected. For the moments, the z-axis is dominant during arm raises out to the side and the y-axis torque being dominant during arm raises out the front of the body.

A percent error analysis was performed between the actual data collected and the data generated by SD/FAST. Looking at Figure 14, the dominant force, in the x-axis, shows an average error less than 3.5%. The explanation for the discrepancy is that the overall mass of the model created for the SD file was less than that of the test subjects. Figure 15 shows a less convincing figure: the average percent error in the z-torque is approximately 26%. The most notable difference lies in the two ends. This discrepancy is explained as follows: during the data collection at the ABF, the test was conducted as two movements with one movement consisting of the left arm being raised and lowered, and the second movement being the movement of the right arm. Within SD/FAST however, the analysis had to be done in four parts. The first part would consist of initializing the system for one arm raise. The second part would require the system to be reassembled for the lowering of the arm. Two more parts were involved for the resetting of the other arm. The ability to change direction in SD/FAST was not able to be done at the time of this report for this project. The change of direction would result in an acceleration change and cause a torque at the feet as showed by the actual data collection. However, since the SD/FAST model was not able to model that acceleration change, the error become apparent.

## 4.0 Conclusions

A working user's interface has been created that can be used to perform simple dynamic analyses on a variable-size human SD/FAST model. It is possible to prescribe the motion of any joint, and apply a point-force to any desired body. Likewise, prescribed motion can be turned on or off, and the positions and velocities of any joint can be held fixed during assembly and initial velocity analysis, respectively. Also, the reaction forces and moments that act throughout the body can be determined as functions of time. Although the results are satisfying to the appetite, we'd prefer more salt. Moreover, the loads generated with the SD/FAST analysis predict the actual loads to within an uncertainty of 20 percent. Nonetheless, more validation studies need to be performed, since only two cases have been analyzed thus far.

## 5.0 Future Directions

So far, all analyses have been relatively "simple." The motion of the various joints have been simplified from their actual complexity in order to generate an efficient code. One problem of intricate motion is found in the prescription. The motion allowed currently allows the equation to be only of the 6<sup>th</sup> order. Real time capture of the motion would perhaps be a better way of prescription. The complex Multijoint motion can be captured using the virtual reality system. And in order to capture acceleration data, the use of accelerometers might also be an option. In addition, joint limits and passive joint characteristics need to be incorporated into the model.

Furthermore, although the SAR are relatively easy to use, the complexity and potential of SD/FAST is limited with the use of these routines. User developed algorithms, by contrast, allow SD/FAST to be limited only by the imagination of the user.

Pro\_Mechanica is SD/FAST based. More and more design work is being created via Pro/Engineer. Dynamic modeling may be eased by the use of a Pro/Mechanica interface. A quick glimpse at the User's Manual for Pro/Mechanica revealed a plethora of possibilities that could be explored.

## 6.0 References

1. Schaffner, G., "Dynamic Analysis of Astronaut Motions During Extra Vehicular Activity", Masters of Science Dissertation, MIT, June 1995.
2. Ko., H., Badler, N., "Animating Human Locomotion with Inverse Dynamics", IEEE Computer Graphics and Application, March 1996.
3. SD/FAST User's Manual, Symbolic Dynamics, Inc., 1991.

4. A. Pandya, S. Hasson, A. Aldridge, J. Maida , B. Woolford, "Correlation and Prediction of Dynamic Human Joint Strength from Lean Body Mass", *NASA Technical Report 3207*, June 1992.

5. A. Pandya, S. Hasson, A. Aldridge, J. Maida , B. Woolford, The Validation of a Human Force Model to Predict Dynamic Forces Resulting from Multi-joint Motions, *NASA Technical Report 3206*, June 1992.

## **7.0 Acknowledgement**

James C. Maida  
GRAF  
ABF  
Symbolic Dynamics Inc.  
Carlos Roithmayr

## 8.0 Appendix

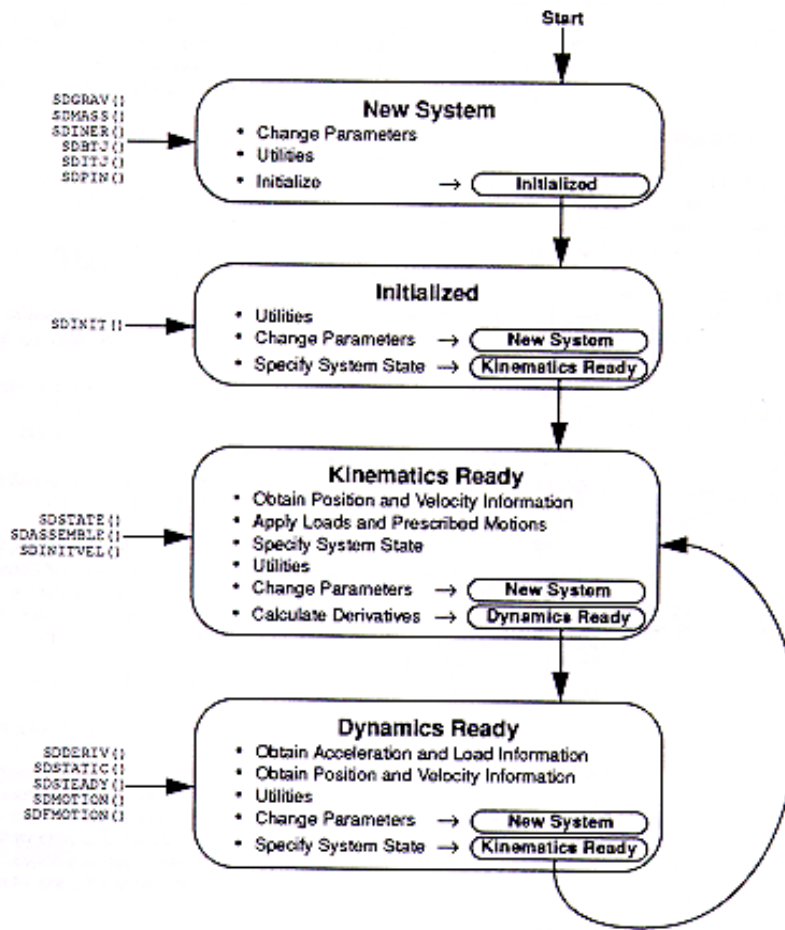


Figure 1. Diagram of Program Flow for General SD/FAST Analysis.

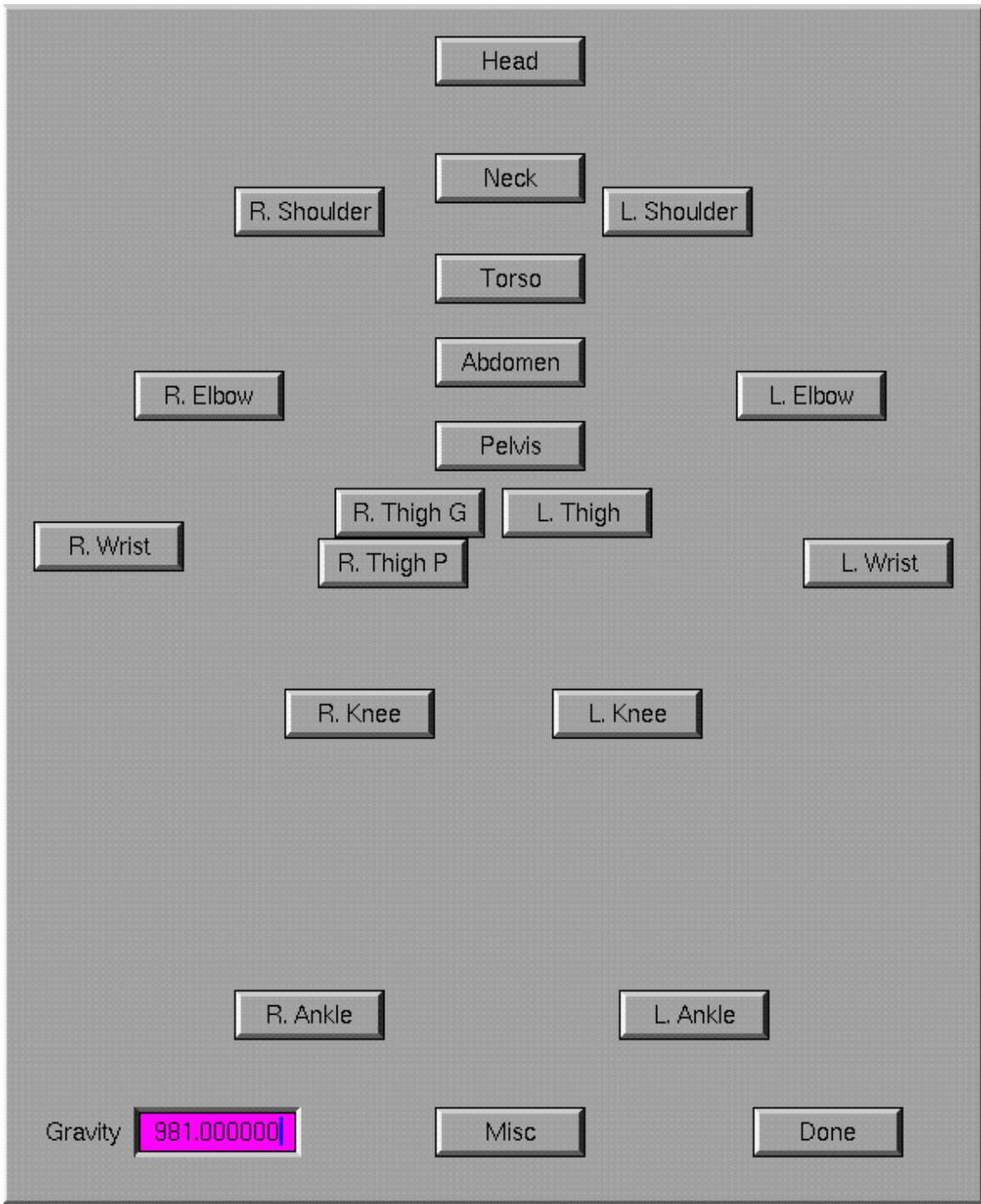
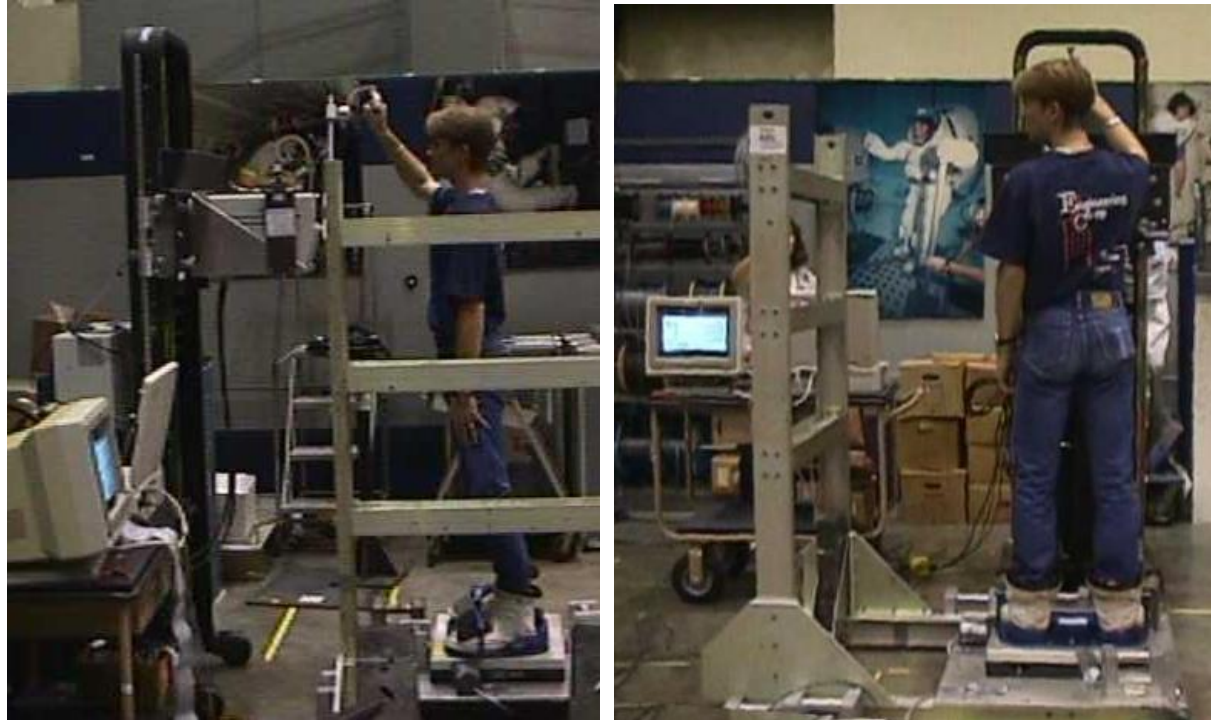


Figure 2. Top level of Graphical User Interface



**Figure 3. Side and Front view of 2nd case Study: Adduction/Abduction Isonkinetic Arm Movement.**