The Link Foundation Fellowship In Advanced Simulation and Training

Hybrid Force-Feedback Control for Virtual Environments

by Carol Chesney

Department of Engineering
University of Florida

September 1998

The Link Foundation Fellowship In Advanced Simulation and Training

Hybrid Force-Feedback Control for Virtual Environments

by Carol Chesney

Department of Engineering
University of Florida

September 1998

The Link Foundation Fellowship In Advanced Simulation and Training

Hybrid Force-Feedback Control for Virtual Environments

by Carol Chesney

Department of Engineering
University of Florida

September 1998

The Link Foundation Fellowship in Advanced Simulation and Training

A grant is awarded to foster advanced level study in simulation and training research: to enhance and expand the theoretical and practical knowledge of how to train the operators and users of complex systems and how to simulate the real-world environments in which they function; and to disseminate the results of that research through lectures, seminars and publications. This publication is the final report of the fellow's research.

LINK AWARDEES

1991-92	Joe Dumas / University of Central Florida
1992-93	Julia Carrington / University of Central Florida
1992-93	Ginger Watson-Papelis / University of Iowa
1993-94	Michael Bajura / North Carolina University
1993-94	Eric Foxlin / Massachusetts Institute of Technology
1993-94	Gary R. Geroge / Binghamton University
1993-94	Matthew P. Reed / University of Michigan
1993-94	Benjamin A. Watson / Georgia Institute of Technology
1993-94	Ronald J. Guckenberger / University of Central Florida
1993-94	Henry A. Nelson / University of Central Florida
1994-95	Kurt M. Joseph / Kansas State University
1994-95	Mark R. Mine / University of North Carolina
1994-95	James R. Otto / University of Kentucky
1994-95	Jim Chen / University of Central Florida
1995-96	Ellen Bass / Georgia Institute of Technology
1995-96	Jonathan Cohen / University of North Carolina
1995-96	Doreen Comerford / Kansas State University
1995-96	Curtis Lisle / University of Central Florida
1995-96	Hans Weber / University of North Carolina
1995-96	Donna Wilt / Florida Institute of Technology
1996-97	Robert A. Glaser / University of Massachusetts
1996-97	Leslie L. Heimenz / Ohio State University
1996-97	Mark C. Kilby / University of Central Florida
1996-97	William R. Mark / University of North Carolina
1996-97	Ellen Scher Zagier / University of North Carolina
1996-97	Douglas A. Peterson / University of South Dakota
1997-98	Robert Franceschini / University of Central Florida
1997-98	Barry Kelly / Binghamton University
1997-98	Darren Law / University of Central Florida

1997-98	Mark Livingston / University of North Carolina
1997-98	Glenn Martin / University of Central Florida
1997-98	Carol Chensey / University of Florida

The Institute for Simulation and Training and the University of Central Florida
Administer this program on behalf of the Link Foundation.
Dr. A. Louis Medin, Executive Director
Institute for Simulation and Training
3280 Progress Drive
Orlando, Florida 32826-0544

Hybrid Force-Feedback Control for Virtual Environments

Carol Chesney
September 1998

Abstract:

The scope of this project is the design and implementation of a force-feedback device that utilizes brakes as well as motors for hybrid force control.. The device is a pen-sized device for planar point force-feedback. The design and software are implemented with consideration towards extending the force-feedback technology to three dimensions.

The benefits of such a hybrid device include increased safety and realism. Larger forces can be reflected safely in the form of impenetrable constraints and instabilities can reduced or eliminated. The passive force component is the only way to accommodate the model of static friction. Delay in force measurement means that movement would occur before actuators could respond. Frictional forces inherent in the device dictate the perceived static friction, and it is desirable to be able to control the static friction.

Introduction to force feedback User Interfaces

Hannaford characterized the aim of force-feedback research as consisting of 3 Pillars. The first and most important is the goal of safety. Force display inherently contains more potential for safety hazards than other user interfaces. The second pillar is realism. He describes this as making free space feel free and solid objects feel solid. Other definitions include Colgate's concept of impedance matching. The third pillar is ease of use. Included in this goal is portability. There is not so much standardization of force display as there is for other types of interfaces. They usually require their own controller and power supply. They can also be quite large and may need adjustments or calibrations to accommodate different users. The use of hybrid force-control allows for improvement in meeting the first two goals.

<u>Simulation of virtual environments</u>

The three pillars that Hannaford describes relate both to tele-operation where the reflected force is measured and virtual environments where the interaction is simulated and the reflected force is calculated. Colgates impedance matching only applies to virtual environments where the impedance of the environment can be modeled. The effectiveness of force-feedback in virtual environments is dependant on the performance of simulation software and the hardware. Virtual Reality is an application where physical modeling is needed to determine the graphics and the user interaction. The physical interaction is bidirectional: the user input alters the modeled environment and determines the force reflected to the user.

The task of predicting contact forces in a virtual environment can be quite complicated. Realistic physical simulation requires mass and damping properties as well as stiffness. Physical constraints must also be taken into account. Inertial and frictional forces are relatively simple to model and calculate, but contact force calculation becomes increasingly difficult when the model is allowed to deform. In modeling solid objects, the global displacements must be calculated considering any given constraints. Various types of constraints can consist of absolute/global constraints, constraints relative to other objects, and constraints based on user input position. Some constraints necessitate the use of efficient collision detection.

Hardware

The most basic task force simulation is to be able to simulate perfectly rigid surfaces such as a wall. These surfaces should be passive but generally are not because of the nature of discretely sampled control systems. The model may cause unintended motion and instabilities. The walls will always behave in an active manner, reflecting more force than they dissipate unless damping is employed. Colgate proved that damping is necessary in both the manipulator and modeling to ensure the passivity of virtual walls. He developed simple criteria needed to guarantee this passivity [col 93].

Previously, most researchers attempted to minimize the damping of friction in the hardware, and most still do. This is because minimal friction is desired for the non-contact state, and there will always be a minimal amount of damping in any device. Colgate and others have attempted to control the amount of physical damping with various types of brakes. They have limited use of the passive component to velocity related damping.

Goodall experimented with the design of a device which used only brakes for actuation. This had the advantage of increasing safety, but it meant that only passive environments could be reflected. There were several other limitations that will be discussed later.

Modeling

Metaxas developed a systematic approach for adding dynamics and constraints to parametric models for animation purposes. This approach allowed for local geometric deformations and separate free body motion. Calculations can quickly be performed in the parametric domain and efficient constraint techniques allow for simultaneous calculation of contact forces [met 93].

Qin used this approach to develop Dynamic NURBS (DNURBS). NURBS is the defacto standard for CAD models because they can represent analytic shapes exactly and are easy to calculate. The current state of development for DNURBS includes free form deformation, data fitting, integration of geometric constraints, and surfaces in the forms of tensor-product, swung and triangular parametric domains.

The general procedure for calculating forces and deformations is to assemble mass, damping and stiffness matrices using integration. The dynamic equation is solved for current values of control points and weights. Because the goal of Qin's work was an intuitive interface for free-form design, several modifications are being implemented to allow for the computation of realistic force modeling. The current state only allows for constant dynamic properties in the parametric domain. A systematic approach is being developed to convert properties of the virtual environment into the parametric domain. The damping model is Raleigh dissipation within the object. Surface friction can easily be added to the model with the selection of two frictional coefficients.

To represent point contact with user input, geometric constraints need to be manipulated dynamically. Adachi developed a fast method by calculating the closest point on the object relative to the input position. The surface normal at this point is compared to the direction of the distance from the object to the input point. If the two directions have a positive dot product, then no contact is made and no constraint is enforced. If the directions have a negative dot product, then the input position is taken as a new geometric constraint. Adachi's work involved non-deformable models. He found performance of surface following was improved with the addition of viscous damping.

Hannaford and Buttulo looked at friction and reaction force modeling during object collision. They examined the three surface friction force models of dynamic, static, and coulombic or kinetic friction. They did not consider rolling friction, which is a resistive moment, not a force. In their joint model, the objects were not allowed to rotate. This would correspond to having infinite rotational inertia. The radius of gyration would be infinitely larger than its physical radius at any point! The static friction's upper value is given by the coefficient formula, but it is also dependant on the object's inertial characteristics, both linear and rotational.

Buttulo discussed the effects of static and Coulombic friction on collision in his book, but declared that they were too difficult to be used in his simulation. Instead he used the easily calculated dynamic friction. This is the same model that Adachi used to increase users surface following abilities; howvere, it is unrealistic for most surfaces. Buttulo did mention this simplification would result in a lack of realism and suggested further research on the subject.

Design Development

Development of general specifications

Human Factors Research

According to Shimoga, human fingertips can actuate at approximately 5-10 Hz., and can sense chattering in the range of 300-400 Hz. The fingertip force resolution is about 0.5 N. Maximum fingertip exertion is 30-50 N for short periods and 4-7 N for sustained periods without compromising force sensation. According to Shimoga, the minimum bandwidth a force-feedback device should be able to reflect is 20 - 30Hz [shi 93].

More human factors information is available from Woodson. This source states that the hand can sense from 0.016 lb. To 4.5 lb. The maximum force a user can exert on a joystick type device is approximately twenty pounds in any direction [woo 92].

Tesar defines a force resolution of .5 to 5 grams as a typical requirement for microsurgery tasks. This value corresponds to devices smaller than the joysticks listed above. It represents the very high end of resolution [tes 96].

What is perhaps more useful than measured performance data is user reaction to available technology. For example the resolution of pointer devices range from 300 dpi to 1200 dpi. User reaction shows that anything 400 dpi and below is unacceptable while device about 600 dpi are quite good. A small work spake is tolerable if there is high resolution and some means of relocating the virtual workspace..

Available force-feedback devices include the PhANTOM, Impulse Engine and Per-force. Each of these devices is in the range of desktop manipulation. The PhANTOM is a finger based manipulator, the Impulse engine is a pen sized joystick. Information was obtained through manufacturers's literature and tech support.

Available Device Specifications

Device	PhANTOM	Impulse Engine	Per-force
workspace	5" x 7" x 10"	5" x 9" x 9"	4" x 4" x 4"
position resolution	800 dpi (0.03 mm)	1100 dpi (0.023 mm)	0.0077 mm linear
max force	1.9 lb.	2 lb.	motors 30 oz-in
bandwidth	unavailable	650 Hz	unavailable
degrees of freedom	3 translation	5 tracked, 3 force-feedback	6 force-feedback

<u>general specifications</u>

From the above sources of information, the design criteria were determined. A set of intermediate design criteria will be used for a scaled up intermediate proto-type version. The magnetic hysteresis brakes currently used have an intrinsically low bandwidth and will be replaced with alternate technology after the first prototype has been tested.

Desired Device Specifications

	final device specs.	prototype goals
workspace	6" diameter	12" diameter
position resolution	900 dpi	400 dpi
max force	2 lb. active 10 lb. passive	2 lb. active 10 lb. passive
force resolution	10 grams	10 grams
inertia	< 50 g "perceived"	minimal directionality of inertia throughout workspace before software correction is implemented
bandwidth	100 Hz actuation, 600 Hz sampled	30-50 Hz actuation, 600 Hz sampled

Development of additional performance criteria: Inertial and control requirements

Ideally, the interface should have no apparent inertia, or inertia equivalent to the simulated device. For example, it would be acceptable to have an inertia equivalent to a scalpel in a simulated surgery. Unfortunately the inherent inertia of input devices tends to be very directional, unlike objects in free space. The PhANTOM's design and controller theoretically reduce the perceived inertia to less than 75 grams. High quality feedback and a good controller can be used to eliminate perceived inertia; however, a good design will greatly improve performance. The inverse dynamic analysis will be implemented by the controller to remove inertial effects of the device, but the design stage and controller require a thorough definition of inertial properties.

Yashikawa developed a dynamic manipulability index that reduces actuator and inertial characteristics to a single value. Several other researchers use similar dexterity measures based on the condition number of the Jacobian. The dexterity indices are local values and need to be studied over the workspace. Tesar used an integrated average over the workspace as a global performance measure [tes 96]. Lee and Duffy have shown that for some manipulators, the prescribed calculations produce invalid results of mixed units [lee 98]. These dexterity measures do not provide a valid comparison between vastly different types of manipulators, and care must be used to insure they are valid for the manipulator being studied

Featherstone provides a rigorous derivation of equivalent inertia and inverse inertia using a lie algebra equivalent to screw theory. It can be quite cumbersome and only the inverse inertia can be guaranteed to exist. Kane and Levison use a Lagrangian approach and screw algebra for their dynamics derivations. These works provide a foundation for the inertial performance anaylsis.

Formal Inertial Characterization

A combination of Featherstone's work and a lagrangian approach was used to derive quality indices. The type of device considered is a five-bar mechanism because parallel devices are generally considered to be stiff and have more symmetrical inertial effects. The same analysis applied to a serial robot shows this to be true.

First the equivalent mass as seen from the actuators is derived in matrix form. The two actuated joints will be both connected to ground as this helps to reduce the inertia of the device. The individual elements of the mass matrix Mij are calculated from the Kinetic energy K as follows:

$$M_{i,j} = \partial/\partial\ddot{\theta}_j(\,d/dt\,(\partial k/\partial\theta_i))$$

Here the two variables of interest are the actuator joint angles. The inertial quantity of interest is the equivalent mass as seen from the user's input. This is given by

$$\Phi = J \cdot M \cdot J^T$$

where J is the jacobian. The desired qualities are a non-singular jacobian, and an inverse inertia with large diagonality and small directionality. Normalized indices for the symmetry and diagonality can be calculated from the eigenvalues as follows.

$$SYM = \left|\frac{\lambda_2-\lambda_1}{\lambda_2+\lambda_1}\right| \quad ; \qquad DIAG = \frac{\varphi_{1,2}\times\,\varphi_{2,1}}{\varphi_{1,1}\times\,\varphi_{2,2}}$$

It is desirable for these indices to be close to zero.

Optimization of Design Parameters

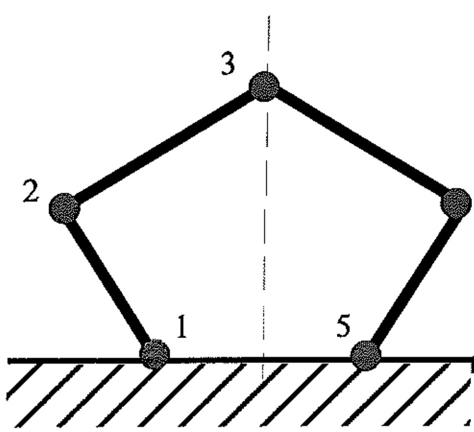
The symmetry and diagonality indices are functions not only of the manipulator parameters, but also of the location of the end effector. The value of a given design must consider the range of values throughout the workspace. It is even more important for the indices to be consistent throughout the workspace than it is for them to be close to zero. For this reason the variance of the indices was considered in the optimization process instead of using just a weighted average. The design optimization is based upon these desired value functions as well as the constraints of the desired workspace and force-reflection capability. For the following hybrid mechanism the optimization process gave a parameter values of

l0 = 4"
l1 = 14 "
l2 = 15"

Hybrid Design

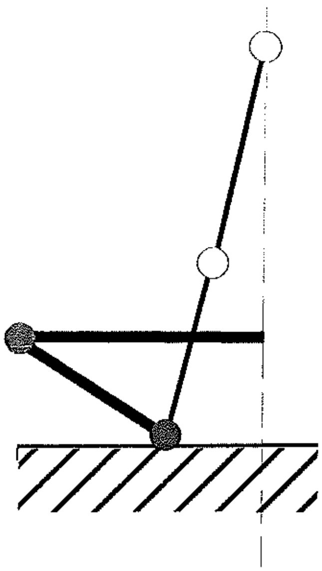
The hybrid design differs from the other products in that it is a planar device, and utilizes passive actuation. The passive brake actuation necessitates the decoupling of directionality into normal and tangent directions according to the model. This gives 2 + D.O.F. The brakes operating in the normal direction can be used as dampers or as passive actuators if the required force becomes too large to reflect safely with motors. The brake corresponding to the direction corresponds to surface friction (static, coulombic, etc.).

A generic symmetric 5-bar was chosen as for the design because parallel mechanisms are stiffer than serial mechanisms. It allows good mechanical advantage for force control with smaller inertial effects.

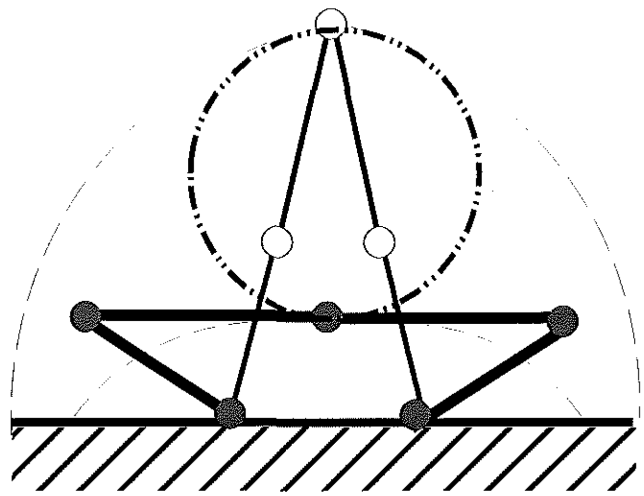


Five bar mechanism

There are a couple singularities that affect the workspace. The first singularity is caused when joints 2,3, and 4 become co-linear. The second type of singularity occurs when joints 1,2 and 3 or 3,4, and 5 become co-linear. This second singularity is the limit of the mechanism's reach. The best area of workspace is above the first singularity and below the second singularities. The user workspace should be symmetric. A circular workspace uses the available area and satisfies the criteria well.



Singular positions



Available and user workspaces

The mechanism inertia will be directional. It is desirable for the perceived inertia to be small and symmetric. To obtain the equivalent inertia, the position of all joint axes needs to be calculated. The following equations give the reverse analysis.

$$\sin(\alpha+\varphi) = \sin\alpha\cos\varphi+\cos\alpha\sin\varphi = \frac{\delta y}{2 l_2} + \frac{h\delta x}{l_2 R}$$

$$\cos(\alpha+\varphi) = \cos\alpha\cos\varphi-\sin\alpha\sin\varphi = \frac{\delta x}{2 l_2} - \frac{h\delta y}{l_2 R}$$

$$\sin(\varphi) = \frac{\delta y}{R} \qquad \cos(\varphi) = \frac{\delta x}{R}$$

$$\sin(\alpha) = \frac{h}{l_2} \qquad \cos(\alpha) = \frac{R}{2 l_2}$$

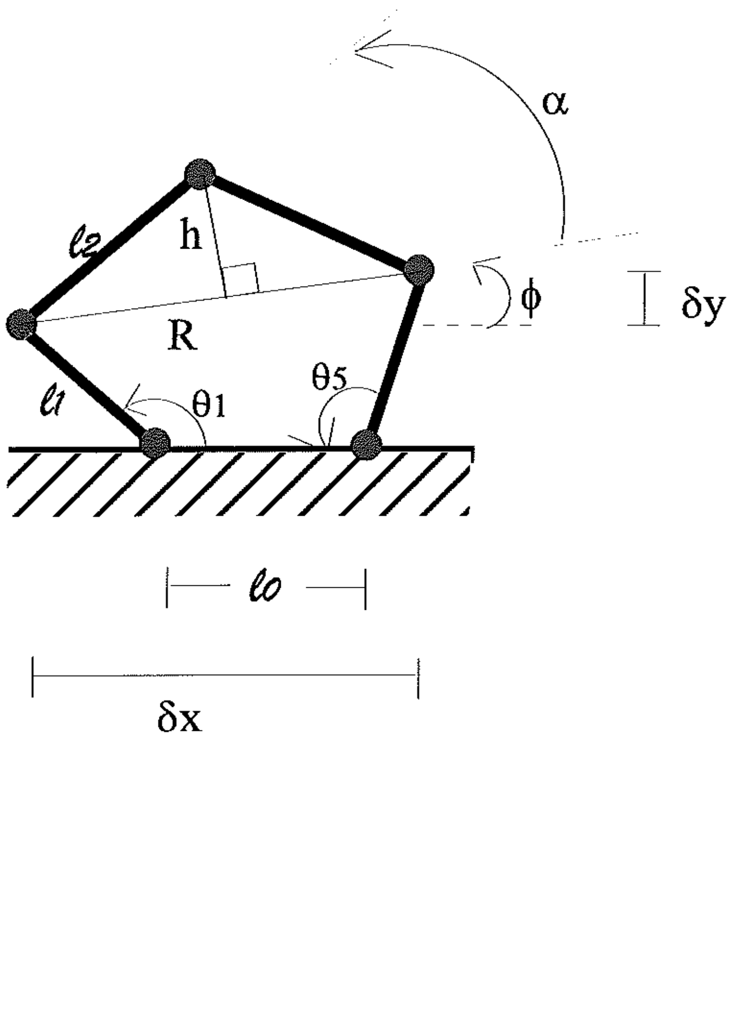
$$\delta x = l_0 - l_1(\cos\theta_5 + \cos\theta_1)$$

$$\delta y = l_1(\sin\theta_5 - \sin\theta_1)$$

$$R = \sqrt{(\delta x)^2 + (\delta y)^2}$$

$$h = \sqrt{l_2^2 - \frac{1}{4} R^2}$$

$$\frac{h}{l_2 \; R} = \frac{\sqrt{l_2 + \frac{1}{4} R^2}}{l_2 \; R} = \sqrt{\frac{1}{4} R^{-2} - l_2^{-2}}$$



REFERENCES

[bur 96] Burdea, Grigore. Force and Touch Feedback for Virtual Reality. John Wiley & Sons, Inc. New York, New York. 1996.

[but 95] Buttolo, P., Kung, D., and Hannaford, B., *Manipulation in Real, Virtual and Remote Environments.* Proceedings of IEEE conference on System, Man and Cybernetics, vol 5, Vancouver, BC, October 1995.

[col 93]Colgate, J.E., Grafing,P.,Stanely,M., and Schenkel G. *Implementation of Stiff Virtual Walls in Force-reflecting Interfaces.* IEEE Annual Virtual Reality International Symposium, 1993.

[fab 95] Fabiani, Lionel, Burdea, Grigore, Lagrana, Noshir, and Gomez, Daniel. *Human Interface Using the Rutgers Master II Force Feedback Interface.* IEEE Annual Virtual Reality International Symposium, 1995. pp. 54-59.

[fea 84] Featherstone, Roy. Robot Dynamics Algorithms. Kluwer Academic Publishers. Norwell, MA. 1987.

[fox 93] Fox, Stuart Ira. Human Physiology. Fourth edition. Wm. C. Brown Publishers Dubuque, Iowa. 1993.

[lee 98]Lee, J., Duffy, J., and Hunt, K. A Practical Qualiy index Based on the Octahedral Manipulator. Intgernaltional Journal of Robtics Research. Vol. 17. October 1998.

[leh 83] Lehmkuhl, L. Don, and Smith, Laura. Brunnstrom's Clinical Kinesiology. F.A. Davis Company. Philadelphia, PA. 1983.

[met 92] Metaxas, D., and Terzopoulos D. *Dynamic Deformation of Solid Primitives with Constraints.* Computer Graphics volume 26, number 2, July 1992.

[met 93] Metaxas, D., and Terzopoulos D. *Shape and nonrigid motion estimation from synthesis.* IEEE Transactions on Pattern Analysis and Machine Intelligence. pp.580-591. July 1993.

[ogi 95] Ogi, Tetsuro, and Hirose Michitaka. *Multisensory Data Sensualization Based on Human Perception.* IEEE Annual Virtual Reality International Symposium, 1995. pp. 66-70.

[qin 95] *Dynamic NURBS Swung Surfaces for Physics-based Shape Design.* Computer-Aided Design, Vol. 27, No 2, pp. 111-127, 1995

[qin 96] Qin, H., and Terzopouls, *D. DNURBS: A Physics-Based Framework for Geometric Design.* IEEE Transactions on Visualization and Computer Graphics, Volume 2, Number 1, March 1996

[shi 93] Shimoga, K. *A Survey of Perceptual Feedback Issues in Dexterous Telemanipulation: Part I. Finger Force Feedback.* IEEE Annual Virtual Reality International Symposium, 1993.

[tes 96] Tesar, D. and Peterson, J. *Criteria-Based Performance Evaluation of Manual Controllers.* a report to the Department of Energy. Department of Mechanical Engineering, University of Texas at Austin. 1996.

[ter 91] Terzopoulos D., and Metaxas, D., *Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics* . Computer Graphics volume 26, number 2, July 1991.

[woo 92]Woodson, Wesley E., Tillman, Barry, and Tillman, Peggy, Human Factors Design Handbook, McGraw-Hill, Inc., New York, 1992.

[yag 96] Yagel, Roni, Stredney, Don, Wiet, Greogory. Schmalbrock, Petra, Rosenberg, Louis, Sessanna, Dennis, Kurzion, Yair, and King, Scott. *Multisensory Platform for Surgical Simulation.* IEEE Proceedings of VRAIS 1996.

[yok 95] Yokokohji,Yasuyosh, Hollis, Ralph, and Kanade, Takeo. *What You See is What You Can Feel- Development of a Visual/Haptic Interface to Virtual Environment.* IEEE Annual Virtual Reality International Symposium, 1995. pp. 46-53.

[yos 90]Yoshikawa, Tsuneo. Foundations of Robotics Analysis and Control. Massachusetts Institue of Technology, United States of America, 1990.

Appendix A: Optimization source files

```
%graph_val::: returns a value and 6 constraint values for a given leg length
%       (value, diag, and sym  are integrated over workspace, variance is
%       also taken for diag & symm)
% This will eventually be the value and constraint function for the
%standard optimization
% the input parameter l is leg lengths (l0,l1=l4,l2=l3)
% I would like to define additional parameters to the function, but I will
%have to hard code them instead.

%The parameter uservars would be the variables
%R,MESH, MvarD, MvarS, Im, Tm, gam,  MaxD, and Maxs

%The value & constraint functions in order are.....
% value function= average equivalent mass
%
%CONSTRAINTS
%1st geometric compatability requirement: If this is not met,
%       then all the other constraints are to be returned as a constant >0
%2nd Work space < available workspace: If this is not met,
%   then a workspace of 90% available will be meshed and used to calculate
%   the other constraints (for smoothness)
%3rd = Variance of Diagonality - MvarD
%4th = Variance of Symmetry - MvarS
%5th = Max(Diag) - MaxD
%6th = Max(Symm) - MaxS

% I have been having trouble mixing dfn.s of l0, here lo = 1/2 dist. between
% joints

function [value,res]=val(l)
% set parameters
MinR = 5;   %inches
MESH = 8;  %mesh size of workspace grid
MvarD = 1;
MvarS = 1;
MaxD = 1;
MaxS = 1;
dens = 1;% oz/inch
Im = 0; %for now, do not consider motor inertia!
%Tm = uservars();
```

```
%1st calculate geometric constr: = compatability & regularity!
compat =l(1)- l(2) -l(3);
if(compat>= 0)
  res=[ compat 1 1 1 1 1];
  value = compat;
  return
end
regu =(l(3) -l(1))^2 -l(2)^2;
if(regu>= 0)
  res=[ regu 1 1 1 1 1];
  value = regu;
  return
end

%Calculate Inertial expressions
m2=3*l(3);
Inert=dens*[l(2)^3 l(3)^3] +[Im 0];
%Calculate workspace requirement
% Calcualte reusable analytical expressions
ymin = miny2(l);
ymax = maxy2(l);
dely = ymax-ymin;
if isreal(dely) == 0
  fprintf(1,'dely has gone complex for the following leg configuration:');
  l
  fprintf(1,'Warning: something REALLY wrong with code to get here');
  res=-1;
  return
end
if(dely < 2*MinR)
% for optimiztion code, just plug on with smaller r for more continous
%constraints
%   r = 0.45* dely;
  res=[dely 1 1 1 1 1];
  value = dely;
  return;
else
  r = MinR;
end
yinc = r/MESH;

% REDO FOR NEW DFN. OF l(1)
%center of workspace
```

```
 center = [l(1), ymin + 0.5*dely];

 %set up x & y  mesh points
 x = center(1)-r: yinc: l(1);
 y = center(2)-r: yinc: center(2)+r;
 dim=size(y);
 max_row = dim(2);
 dim=size(x);
 max_col = dim(2);

 data_pt=1;
 for row=1:max_row
   for col=1:max_col
     if((center(1)-x(col))^2 +(center(2)-y(row))^2 ) <= r^2;
       X(data_pt) = x(col);
       Y(data_pt) = y(row);
       data_pt = data_pt +1;

       s = valNew(x(col),y(row),l,m2,Inert);
       %DJ(row,col) = s(1);
       %DMM(row,col) = s(2);
       PH(data_pt) = s(3);
       SYMM(data_pt) = real(s(4));
       DIAG(data_pt) = real(s(5));

     end
   end
end
value  = real(max(PH));
if imag(mean(PH)) ~= 0
       fprintf(1,'Warning: Phi has gone complex');
       r
       l
       fprintf(1,'***************\n');
%return
end
res(1)= compat + regu;
res(2) =-dely + 2*MinR;

res(3)=cov(DIAG);
res(4)=cov(SYMM);
res(5) = max(abs(DIAG));
res(6) = max(abs(SYMM));
```

```
%File = valNew.m is now being used for the full optimization.
%it is the values  of inverse inertial characteristics
%Have had plenty of problems with mixed dfn.s of l0
%here, l0 = 1/2 the dist.  The jac function assumes it is
%full length


%value1= For now the first value function will be the determinate of
%the Jacobian
%3rd is |Phi|
%4th normalized differance the eigen values of phi
%5th is meaure of phi's diagonalness
function val = val5(x,y,l,m2,Inert);

%syms t1, t5;
%check range
if( (l(2) + l(3))^2< ( (l(1) - x)^2 + y^2 ))
  val = [0 0 0 0 0 0];
  return;
else
  l(1) = 2*l(1);
  p_=[x,y];
  thetas = rev(p_,l);
  J=jac(thetas,l);
  if isreal(J) ~= 1
    val = [1 0 0 0 0 0];
    return;
end
%removed the analytical substitution
% now just 1 function call needed

%Do NOT want to try to redo for new dfn. of l0, just double it
%(ooops, apparantly, it didn't need to be doubled)
Mm2 = find_m(l(1),l(2),l(3),thetas(1), thetas(2));

  Mgen(1,1) = m2*l(2)^2 + Inert(1) + Mm2(1);
  Mgen(2,2) = Mgen(1,1);
  Mgen(1,2) = Mm2(2);
  Mgen(2,1) = Mgen(1,2);


  phi = J*inv(Mgen)*J';
```

```
   val(1) = real(det(J));
   val(3) = det(phi);
   val(2) = val(1)/ det(Mgen);
   temp   = eig(phi);
   val(4) = abs(temp(2) - temp(1))/(temp(2) + temp(1));
%   val(5) = abs(phi(1,2)*phi(2,1)/val(3));
%redo this value to try to "ensure" it is between 0 & 1"
   val(5) = abs(phi(1,2)*phi(2,1)/(phi(1,1)*phi(2,2)));

end
```

```
%File = pos
%Forward analysis
%pos(thetas,lengths)= (x,y)'
%performs the 5-bar forward analysis for theta1 & theta5
% Note, here l(1) is the full base length (not half)
% t(1) is theta 1, t(2) is theta5 !
%Tried to update to make more robust

function p=pos(t,l)

ddX = l(1) -l(2)*(cos(t(1)) +cos(t(2)));
ddY = l(2)* (sin(t(2)) - sin(t(1)));
R = sqrt(ddX^2 +ddY^2);
h = sqrt (l(3)^2 - 1/4*R^2);
if R == 0
  p(1)= l(2)*cos(t(1));
  p(2) = l(2)*sin(t(1));
  return
end
p(1)= l(2)*cos(t(1)) + 0.5*ddX -ddY*h/R;
p(2) = l(2)*sin(t(1)) +0.5*ddY +ddX*h/R;

%file = rev
%t(2)= rev(p_,lengths) = (theta1, theta5)
% Inverse analysis for the 5 bar-mechism.
% uses routines from the trig library
%First solves for theta1, and then same method for theta5
%p_ is vector of x & y location, lengths is vector of l0, l1, & l2
% assume l0 = full distance

function t=rev(p_,l)
%first solve for theta1

c2 = (l(2)^2 +l(3)^2 - p_(1)^2 -p_(2)^2) / (2*l(2)*l(3));

t2 = acos(c2);
s2 = sin(t2);
a1=l(2)-l(3)*c2;
b1=+l(3)*s2;
a2=-l(3)*s2;
b2=l(2)-l(3)*c2;

t(1) = Trig2Solve(a1,b1,-p_(1),a2,b2,-p_(2));
```

```
% Now use the same type of code, but now x' = l(1) -p_(1)

c2 = (l(2)^2 +l(3)^2 - (l(1) - p_(1))^2 -p_(2)^2) / (2*l(2)*l(3));

t2 = acos(c2);
s2 = sin(t2);
a1=l(2)-l(3)*c2;
b1=+l(3)*s2;
a2=-l(3)*s2;
b2=l(2)-l(3)*c2;
t(2) = Trig2Solve(a1,b1,-(l(1) - p_(1)),a2,b2,-p_(2));

%Trig2Solve.m
%theta = Trig2Solve(a1,b1,d1,a2,b2,d2)
%solves a system of equations in the form
%a1*C + b1*S +d1=0
%a2*C + b2*S + d2 = 0
% Does not check if a common root exists
% This is may not the best way to do it
% may want to call TrigSolve and look for commons soln.s

function theta = Trig2Solve(a1,b1,d1,a2,b2,d2)

d1=d1+a1;
d2=d2+a2;
a1= d1 -(2*a1);
a2= d2 -(2*a2);
b1= 2*b1;
b2= 2*b2;
d = (a1*b2)-(b1*a2);

residual = (a1*d2)-(d1*a2);
residual = abs( residual*residual - d*((b1*d2)-(d1*b2)) );

if abs(d)>0.001
  x = (d1*a2-a1*d2)/d;
  theta = 2*atan(x);
else
  theta = 0;
end
```