Image Cover Sheet

CLASSIFICATION UNCLASSIFIED	**SYSTEM NUMBER** 507100

TITLE

DADS STERRING SIMULATION

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to:

DRES
DEFENCE RESEARCH ESTABLISHMENT SUFFIELD

CR 97-41

UNCLASSIFIED

DADS Steering Simulation

BY:

Dave Heitkamp, George Lance, Alan Lewis, Scott Happel
CADSI Consulting

SUBMITTED TO:
SCIENTIFIC AUTHORITY

David Mackay
Defence Research Establishment Suffield
Box 4000, Medicine Hat, Alberta, T1A 8K6

March 1997

WARNING

"The use of this information is permitted subject to recognition of proprietary and patent rights."

CRAD

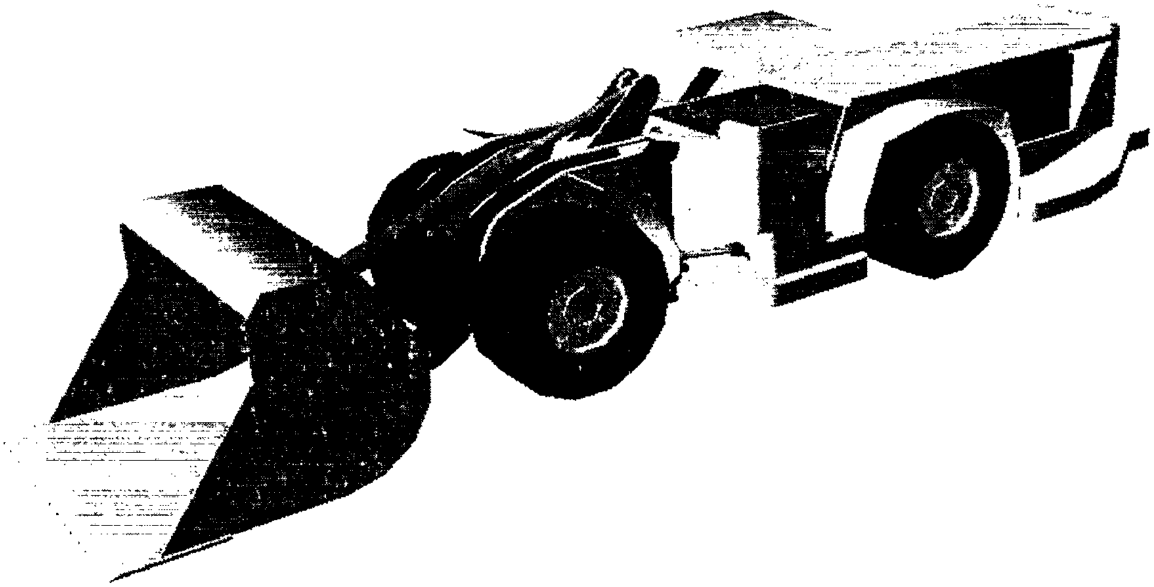
National Defence Défense nationale Canada

DADS Steering Simulation
for
Defense Research Establishment Suffield (DRES)
Mobility Systems Section
Defense Technologies Division

Dave Heitkamp
George Lance
Alan Lewis
Scott Happel

CADSI Consulting

31 March 1997



CADSI
2651 Cross Park Road
Coralville, IA 52241
(319) 626 - 6700

Table of Contents

Introduction

The Vehicle Concepts Group (VCG) of the Defense Research Establishment Suffield (DRES) conducts research aimed at quantifying and enhancing the mobility of vehicles, particularly in off-road environments. To facilitate ongoing research and to support DRES' commitment to the PRECARN TRAM project, AGCG developed a dynamic model of an articulated vehicle. The model will be used at DRES to further develop a sensing strategy and control algorithms for the vehicle allowing it to traverse a known path autonomously.

Model Description

Mechanical Model

Since the goal of this project was to implement an autonomous steering controller, the mechanical and hydraulic models have been defined as simply as possible. The model of the loader is comprised of 11 bodies which are summarized in Tables 1 and 2. Each body's Non-Centroidal Body Fixed (NCBF) reference frame is coincident with the world's coordinate system at (0.0, 0.0, 0.0) which is located along the centerline of the vehicle, below the front axle at ground level.

Table 1 shows the center of gravity (CG) location of each body and Table 2 shows the mass properties for each body. The CG locations, as well as the mass properties, were determined from a PRO/E model of the loader.

Table 1: Loader CG Locations

	X	Y	Z
BUCKET	0.000	78.049	12.023
BOOM	0.000	14.500	35.000
FRONT_FRAME	0.000	436.380	2.477
ART_FRAME	0.000	-48.729	33.840
REAR_FRAME	0.000	-176.000	40.000
FRONT_AXLE	0.000	0.000	27.800
REAR_AXLE	0.000	-124.999	27.800
L_FRONT_WHEEL	-29.900	0.000	27.800
R_FRONT_WHEEL	29.900	0.000	27.800
L_REAR_WHEEL	-29.900	-125.000	27.800
R_REAR_WHEEL	29.900	-125.000	27.800

Table 2: Loader Body Mass Properties

	Mass	I_{xx}	I_{yy}	I_{zz}
BUCKET	0.291	296.041	253.455	365.137
BOOM	0.178	96.502	50.769	106.733
FRONT_FRAME	0.255	19573.060	13654.690	33148.610
ART_FRAME	0.062	7.102	7.350	1.356
REAR_FRAME	1.170	331.500	175.500	331.500

FRONT_AXLE	0.078	0.627	23.838	23.838
REAR_AXLE	0.078	0.627	23.838	23.838
L_FRONT_WHEEL	1.149	443.960	253.694	253.694
R_FRONT_WHEEL	1.149	443.960	253.694	253.694
L_REAR_WHEEL	1.149	443.960	253.694	253.694
R_REAR_WHEEL	1.149	443.960	253.694	253.694

The model is oriented such that X is the lateral axis, Y is the fore-aft axis and Z is vertical. X is positive out the passenger side of the vehicle, Y is positive out the front of the vehicle and Z is positive up.

The loader's frame is divided into 3 major bodies; front, middle(articulate) and rear. The rear frame body is attached to the middle frame body at two points. The lower point is modeled as a spherical joint (ball-and-socket). The upper point is modeled as two perpendicular distance constraints. A distance constraint forces or constrains the points on two bodies to remain a fixed distance from each other. The bodies, however, are free to rotate relative to each other. This collection of constraints has a net result of one rotational degree of freedom (DOF) between the middle and rear frame bodies. The DOF is such that the bodies are free to rotate relative to each other about the vertical axis.

The front frame body is attached to the middle frame body through a single revolute joint (hinge). The revolute joint has one DOF and is oriented such that the axis of rotation is about the fore-aft axis of the vehicle. This creates a torsional DOF between the two bodies allowing one to roll relative to the other. There are also two translational spring-damper-actuators (TSDA) between the front and rear bodies. These springs are offset from the centerline of the vehicle an equal distance and serve as bump stops to limit roll.

There are two axle bodies which support the vehicle. There is one bracketed (welded) to the front frame body and one bracketed to the rear frame body. There are no other suspension components in this model (springs, shocks, etc.).

The axles are supported by two wheel bodies and two tire elements with the tire properties provided by DRES. Rotation of each wheel body is driven by a polynomial function, propelling the vehicle at a nearly a constant speed.

The boom is attached to the front frame body in the same manner as the middle frame body to the rear frame body. That is, a spherical joint and two perpendicular distance constraints. The net result is one rotational DOF, with the boom attached at two points to the front frame. There are also two TSDAs between the boom and front frame which serve to limit the range of motion of the boom (bump stops).

The bucket, in a similar fashion, is mounted to the boom at two points through a spherical joint and two perpendicular distance constraints.

Hydraulic Model

The hydraulic model has been described in a previous report so its discussion will not be repeated here. Appendix A contains a copy of the previous report for reference.

Steering Model

For each road profile, two models are included. One which uses DADS control elements to steer the loader through the course and one which uses DADS/Plant (MATLAB). These models use 2 *'steer'* rays, +/- 30° relative to the fore-aft axis. The wall profile is defined in the DADS curve element called *'wall'*. The distance at which the steer ray contacts the wall is fed back to the controller as the actual sensors are expected to do. The controller is then expected to update the steering input commands.

Figure 1 shows the control block diagram as it is in MATLAB. Only two outputs from DADS are used. They are the distance of the two rays from the nearest wall to the front of the vehicle. The difference between the two distances is passed back to DADS as the spool position for the steering valve.

Figure 2 shows the control block diagram as it is in DADS and how it works with MATLAB. The addition of the limiter/saturator elements keeps the spool position bound.

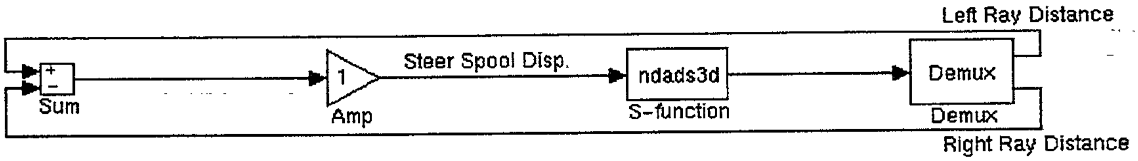


Figure 1: MATLAB Control Block Diagram

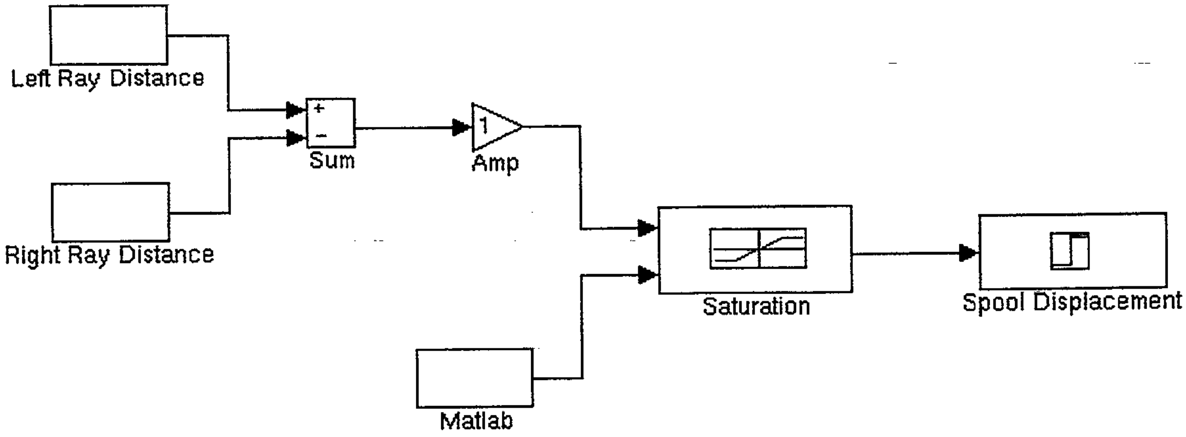


Figure 2: DADS Control Block Diagram

There are two differences between the MATLAB models and the DADS models. One is the method of integration under *analysis>dynamic*. *Method.Integration* should be set to 'MATLAB' for the MATLAB runs and 'PECE' for the DADS runs. The second difference is the input node for the limit control element called *'limit1'*. The input node should be *'dist'* for the MATLAB runs and *'amp'* for the DADS runs.

The functionality of this steering mechanism is based in three sections;

Plant Out/User Control Element
Plant In/General Control Element
User Defined Force Element

The Plant Out Control Elements pass data from DADS to MATLAB. Required for each Plant Out Element is an element name, node name and function parameters. These are all set up automatically if they are generated from the *'plout.x'* utility, which is described later. The function parameters are used as indices to arrays which contain the angles and distances of the rays.

Parameter **1** : (Acceptable Values: 1 or 2)
1 : Angle array
2 : Distance array

Parameter **2** : (Acceptable Values: 1 - n) where n is the # of rays

The Plant In Element passes data from MATLAB back to DADS. Required for this element is the element name and node name. The Plant In Element limits the way the data is applied to the DADS model. It expects to apply forces or torques to bodies or to modify the mass properties of a body. Since this is not the intent, a dummy body (ground) is used to receive the force. This body is fixed to ground and will in no way effect the results or behavior of the rest of the model. In addition to applying the value as a force to ground, this model takes the value and passes it to a DADS Limit control element which is then passed on to the steering hydraulic valve.

The User Defined Force Element is used to pass all relevant information to the FORTRAN subroutine which actually determines the distances and angles. Required first, is a body and a point (triad) on that body which is used to mount the *'rays'*. Next, is a starting, ending and angle increment to layout the rays. These angles are referenced from the above reference point (triad) using the X axis as ø = 0° which is pointing out the passenger side of the vehicle. This analysis used starting angle = 60°, angle increment = 30° and ending angle = 120° which created 3 'rays' at 60°, 90° and 120° degrees relative to the X axis. Equivalently, there is a ray pointing forward and one each at +\- 30° from the fore-aft axis. Only the rays at +\-30° are used for this analysis. Lastly, a curve needs to be specified which describes the *'walls'*. This curve is a collection of line-arc segments joined together to form the boundary walls. **Only line segments are used by this utility, No Arcs!** However, this is consistent with the roughness of the walls in typical passages. Also, the algorithm works well with this case, which is actually quite demanding due to the discontinuous nature of the input from the sensors.

DADS/PLANT Setup

To set everything up so that you can do your own analysis for the steering controller, start by going to the *dres/dadsaux* directory

The included *'paths.txt.tmp'* file is one that works at CADSI; it won't work for you, but might help you understand how to set up your own *'paths.txt'* file correctly. You will need to perform the following steps:

1) Copy your *'paths.txt'* file to this directory. For example, if DADS is installed in *'/usr/DADS/dads8'*, then you would copy the file *'/usr/DADS/dads8/dadsaux/paths.txt'* to this directory.
2) Change the first line so that it points to the *'def'* subdirectory in this directory. For example, if this directory is *'/users/bob/new_lhd/dadsaux'*, then the first line of the *'paths.txt'* file should read: *'/users/bob/new_lhd/dadsaux/def/'* (don't forget the '/' at the end).
3) Set your *'CADSI'* environment variable to this directory. For example: *'setenv CADSI /users/bob/new_lhd/dadsaux/'* (again, don't forget the '/' at the end).
4) To test that you are pointing to the correct dadsaux files, start up DADSModel in 3D mode (DADSModel -3 or dads -3), then go to *Force>UsrFrc*, and create a user-defined force element. If the resulting menu has a number of variables (as opposed to just a name), everything is set up correctly

To create your new DADS-3D analysis executable, first check to make sure that you have your CADSI environment variable set up correctly, go to the *dres/execute* directory and type *'make3 frc48.f usrplnt.f'*. After that, you should be able to type *'make'*, and it will compile the two routines, and link them into a new dads3d executable, called *'ndads3d'*. You can invoke this analysis executable explicitly from a Unix prompt, or you can access it automatically from DADSModel by copying or symbolically linking *'ndads3d'* to your dads *execute* directory. If you do the latter, you should get an informational message when you first start a *'Solve'* session, telling you that it is using the user-defined executable.

To create a new DADS-3D analysis executable for MATLAB, in the execute directory, type *'make mexfile'* and it will compile the two routines, and link them into a new dads3d executable, called *'ndads3d.mexsg'*.

The file *'dres/execute/plout.f'* is a simple utility which writes out a collection of DADS plant-out elements. You will need these to transfer the *'ray'* angle and distance data to SimuLink. Just type *'f77 plout.f -o plout.x'* to create the *'plout.x'* executable. Then type *'plout.x'*, and tell it how many rays to generate, and it will create a partial DADS *'.def'* file called *'plout.def'*.

Edit setup.m and change the *pathtodads* to the corresponding location on your system. Also change the name of the *dadsfiles* as appropriate. Start MATLAB and load the set up file and your control system. The number of plant-out and plant-in elements in DADS needs to correspond to the number of inputs/outputs in the Mux/DeMux elements. When ready, start the analysis in Matlab

Road Profiles

For this project, 4 different terrains were incorporated into the steering model. They are:

Single Bump
Double Bump (One for each track)
General Profile
Separate General Profile

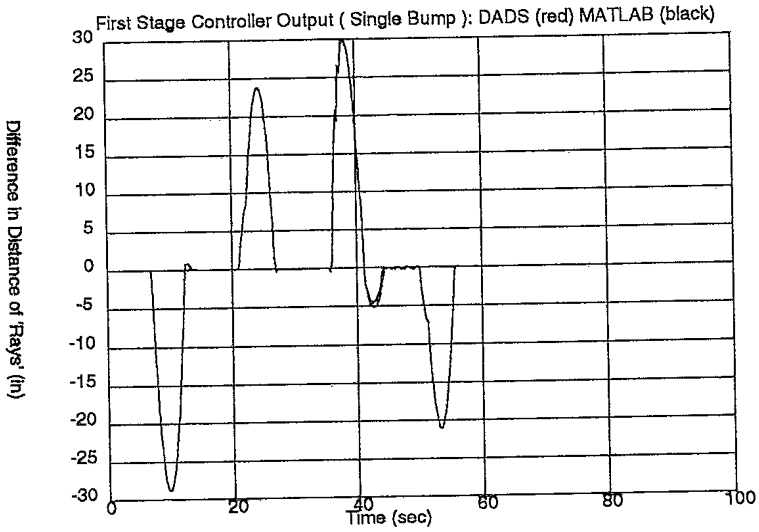
Each road profile is defined in DADS as two columns of data. The first column being the longitudinal coordinate of a point on the road and the second column being the vertical coordinate of that point in the road. This curve describes the road height as a function of longitudinal position which each tire can then reference. Since no suspension was modeled in this project, extreme road profiles were avoided.

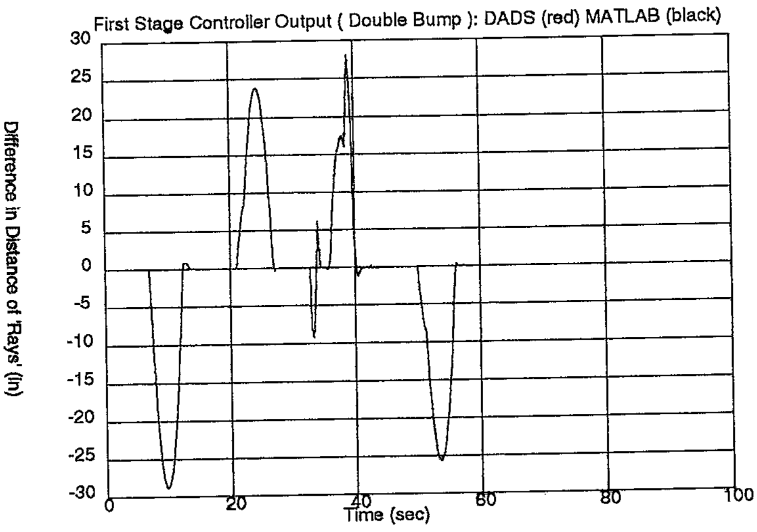
The *single bump* course uses a 6" X 6" bump that stretched from the left to right wall. The *double bump* course uses two 6" X 6" spaced 200" apart. The left and right wheels traverse separate bumps. The *general profile* is a 1.0" RMS curve which is referenced by all the wheels. The *separate general profile* uses the same 1.0" RMS curve, and in addition, has staggered bumps superimposed so the left and right wheels traverse different profiles.

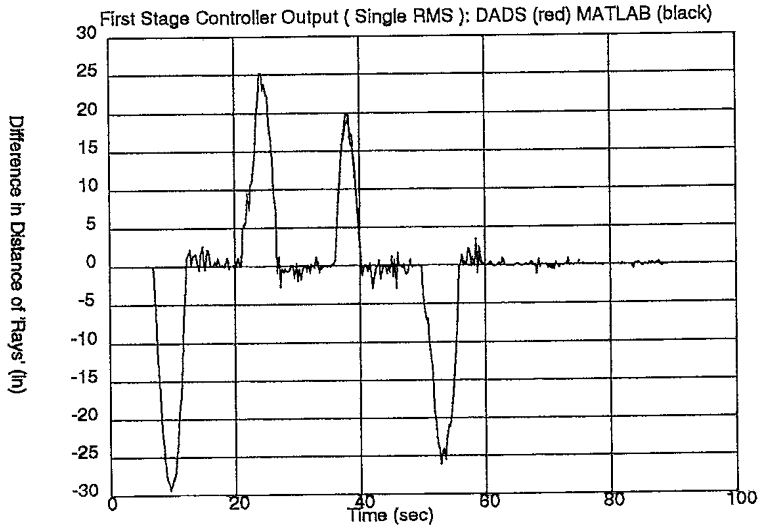
Results

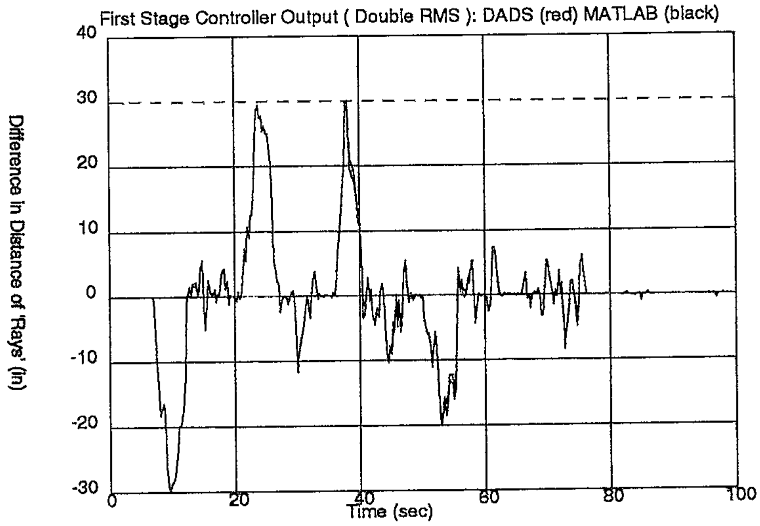
The DADS simulations were run for 100 seconds which takes the loader to the end of the course. The MATLAB runs took longer to perform since there are two sets of equations being solved sequentially. As a result, the MATLAB models were only analyzed for 75 seconds, which took the loader to the end of the bumps and the lateral offset in the course.

The following plots show the difference in distance from the 'walls' to the front frame body. Each of the four spikes in the data represents the vehicle encountering a new section of the course. The zero distance between the spikes shows that the controller was able to center the vehicle in each of the sections.


First Stage Controller Output (Single Bump): DADS (red) MATLAB (black)


First Stage Controller Output (Double Bump): DADS (red) MATLAB (black)


First Stage Controller Output (Single RMS): DADS (red) MATLAB (black)
Difference in Distance of 'Rays' (in)
30
25
20
15
10
5
0
-5
-10
-15
-20
-25
-30
0
20
40
60
80
100
Time (sec)


First Stage Controller Output (Double RMS): DADS (red) MATLAB (black)
Difference in Distance of 'Rays' (in)
40
30
20
10
0
-10
-20
-30
0
20
40
60
80
100
Time (sec)

Conclusions

This analysis has shown the ease with which DADS and DADS/Plant function. It has also shown how information can be passed back and forth between two useful programs to create a powerful utility. This analysis has shown that whether the analysis is performed in just DADS or if both are used, an autonomous steering system can be implemented.

This project has taken a very simplified approach (2 look ahead rays) and established a foundation for a more elegant and powerful controller. If more rays are utilized, so that the vehicle can *'look ahead'* and be more predictive, narrower paths and complexity can be added with confidence of success.

Appendix A
Hydraulic Circuit Model

Introduction

The overall assumption is that the purpose of the current simulation effort is to generate a model that is adequate for evaluation of steering control algorithms. This implies that the hydraulic model is of interest only to the extent that it adequately simulates the vehicle response. The details of the hydraulic parameters are not of immediate interest. The following assumptions and proposed approach are intended to result in a model that will provide entirely adequate results for the intended purpose with substantially less effort than a fully developed model would require.

A fully developed model of the hydraulic system would require substantially more detailed information for the individual components and their interconnection. This would entail considerably more time to develop. It is thought that the difference in the responses between the simplified and detailed models when applying a steering algorithm would be negligible.

The following assumptions are proposed:

1. The load sensing pump will deliver the desired flow rate and pressure upon demand, up to the rated values.

2. The valves provide a flow rate proportional to command input, independent of load.

3. All hydraulic system response times are negligible compared to the response times of the vehicle.

4. Unless test runs prove differently, maximum flow rates and pressures will remain below limiting values for reasonable inputs.

Discussion

Preliminary calculations indicate that rated flow through the respective valves will produce velocities of the associated cylinders as shown:

Steering	6.82 in/s
Hoist	
up	5.31 in/s
down	7.07 in/s
Dump	
up	5.97 in/s
down	6.94 in/s

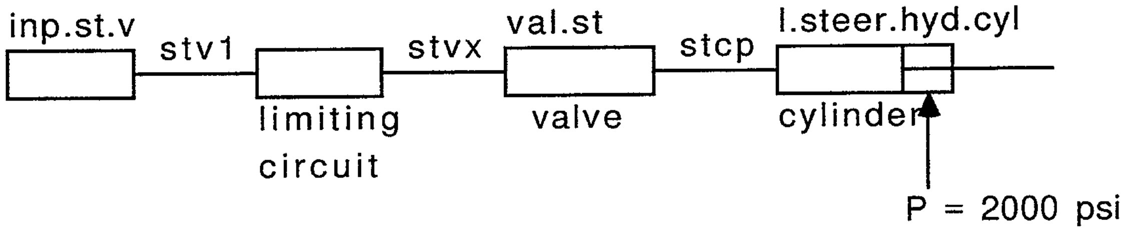
The use of rated flow as a reasonable estimate of the actual performance is based on a calculation of the maximum force outputs available at the cylinders and a comparison to the expected loads. The maximum forces, assuming a supply pressure of 3625 psi and a return pressure of 0 psi are:

Steering	79,750 lb
Hoist	204,994 lb
Dump	182,229 lb

Note that loads that are significant relative to the above values would reduce the maximum flow rates to the cylinders with a corresponding reduction in the associated velocities. However, very significant loads would be required to noticeably affect the velocities. As an example, if the steering load was equal to half the value given above, (approximately 40,000 lb), the maximum velocity would be reduced by only approximately 30%.

The simplified model would consist of replacing the hydraulic cylinders with driver elements, with the inputs being the steer, hoist and dump commands. Velocities and forces could be monitored to evaluate whether or not the model was performing adequately. In particular, it would be easy to determine if pressure or flow rate limits were being exceeded.

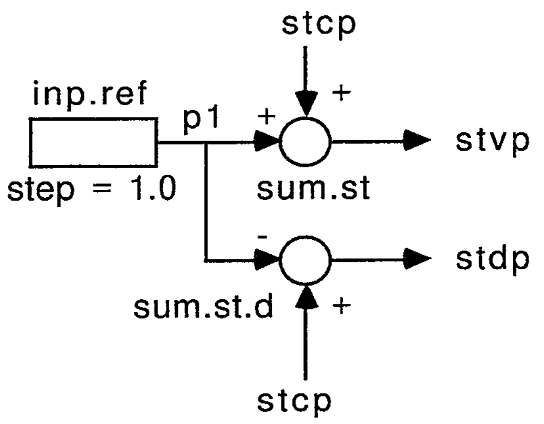
BASIC CIRCUIT - HYDRAULIC MODEL - TYP (STEERING)



Cylinder length = steerpos (hoistpos, dumppos)
Remaining designations throughout - replace st with h or d for hoist or dump.

Valve - 3-way: ch 1 - stvp, ch 2 - stcp, ch 3 - stdp

VALVE PRESSURE CIRCUIT



stvp = p1 + stcp
stvp - stcp = p1 = constant

stdp = - p1 + stcp

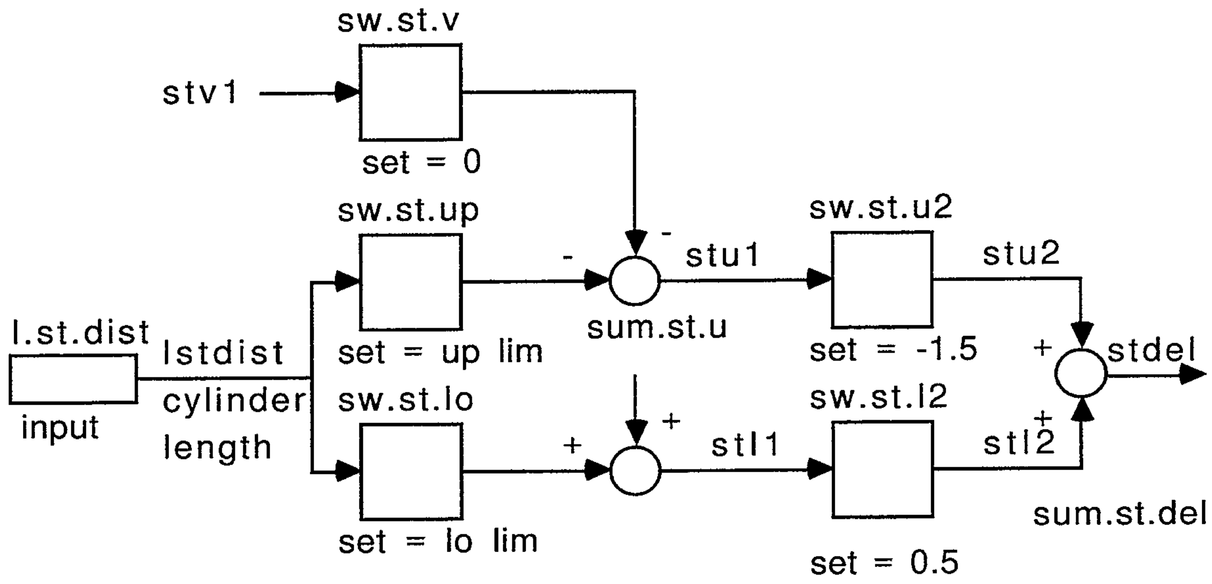
stcp - stdp = p1 = constant

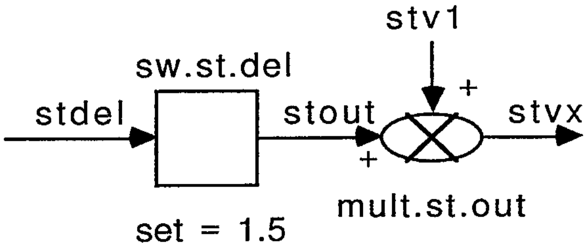
$$q_1 = cx\sqrt{stvp - stcp} = cx \qquad (p1 = 1.0)$$

$$q_1 = -cx\sqrt{stcp - stdp} = -cx$$

Therefore, flow rate is proportional to valve position.

LIMITING CIRCUIT - HYDRAULIC MODEL - TYP (STEERING)





LIMITING CIRCUIT FUNCTION

Desired: Velocity set to zero (i.e., valve position set to zero) if:

a) Position is greater than or equal to upper limit and command velocity is greater than zero.

or b) Position is less than or equal to lower limit and command velocity is less than zero.

Otherwise, velocity equals command velocity.

Circuit Function

lstdu = 1 if lstdist > upper limit

lstdl = 1 if lstdist > lower limit

stu1 = -2 if stv > 0 and lstdist > upper limit
= -1 if stv > 0 and lstdist < upper limit
= -1 if stv < 0 and lstdist > upper limit
= 0 if stv < 0 and lstdist < upper limit

stl1 = 2 if stv > 0 and lstdist > lower limit
= 1 if stv > 0 and lstdist < lower limit
= 1 if stv < 0 and lstdist > lower limit
= 0 if stv < 0 and lstdist < lower limit

stu2 = 0 if stv > 0 and lstdist > upper limit

stl2 = 0 if stv < 0 and lstdist < lower limit

stdel < 1.5 only if limiting is needed

stout = 1 if no limiting
= 0 if limiting

stvx = 0 if limiting
= stv1 otherwise

HYDRAULIC CIRCUIT

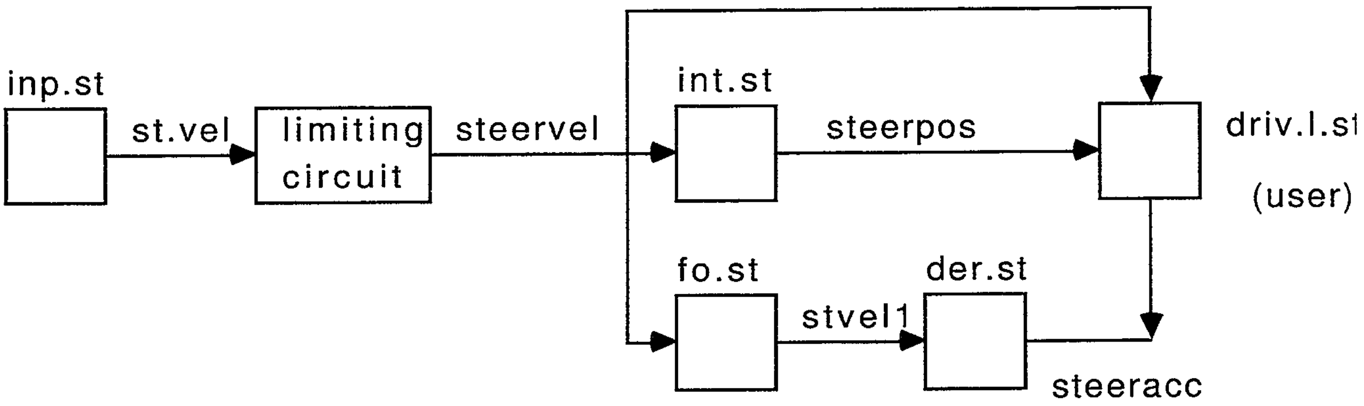
General Procedure

1. Use single acting cylinders with rod end pressure = 2000 psi (Arbitrary to give adequate return driving force)
2. Configure valves to give flow rate proportional to spool position. (Maintain constant pressure drop across valve)
3. Valves are 3-way. Supply and return pressures are generated as a function of cylinder pressure)
4. Position limits are maintained by setting spool position equal to zero when position limit is reached.
5. Only one steering cylinder is used. This was done for convenience since as one steering cylinder extends, the other retracts.
6. Viscous friction was added to cylinders to reduce oscillations and run time.

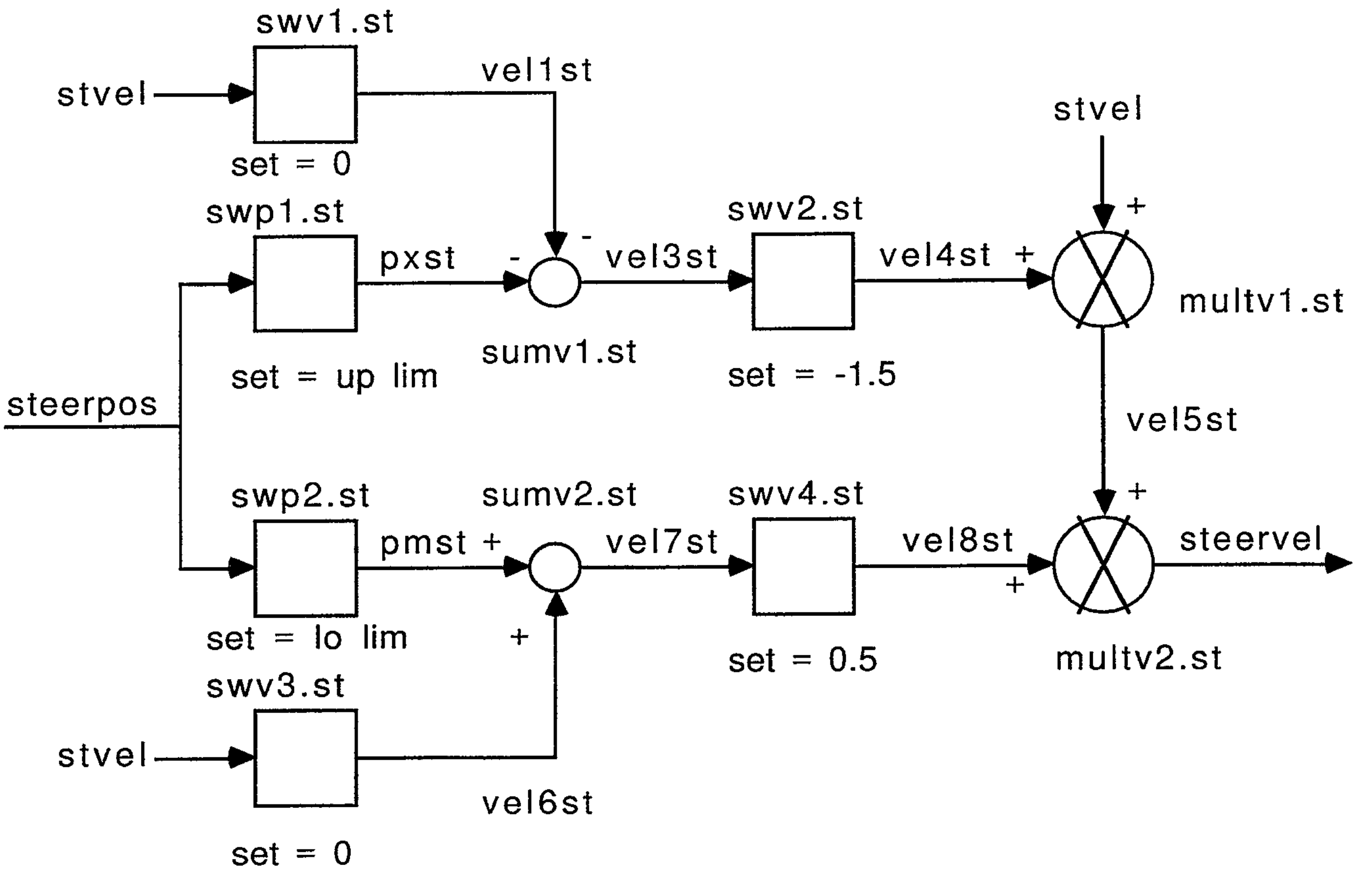
Driver Circuit

1. Driver elements replace cylinders
2. User driver is used - requires inputs of position, velocity and acceleration
3. Use of derivative element results in problems when switching occurs. Therefore, a first order element with a small time constant is added. This smooths out the signal without significantly affecting the dynamic response.

DRIVER CIRCUIT - TYP - (STEERING)



LIMITING CIRCUIT - DRIVER- TYP (STEERING)



Circuit behavior is the same as for the hydraulic limiting circuit up to the variables vel4st and vel8st.

If either vel4st or vel8st = 0, steervel = 0
If both vel4st and vel8st = 1, steervel = stvel

Driver and Cylinder Connections

Steering - Assume left side driven

L.STEER.HYD.CYL

Body 1	REAR.FRAME
Triad	REAR_FRAME_L_STEER_TRIAD
Body 2	ART_FRAME
Triad	ART_FRAME_L_STEER_TRIAD

Hoist - both sides driven

L.HOIST.HYD.CYL

Body 1	FRONT_FRAME
Triad	FFRAME_L_B_HOIST_TRIAD
Body 2	BOOM
Triad	BOOM_L_HOIST_CYL_TRIAD

R.HOIST.HYD.CYL

Body1	FRONT_FRAME
Triad	FFRAME_R_B_HOIST_TRIAD
Body 2	BOOM
Triad	BOOM_R_HOIST_CYL_TRIAD

Dump

HYD.DUMP.CYL

Body 1	BUCKET
Triad	BUCKET_HYD_CYL_TRIAD
Body 2	FRONT_FRAME
Triad	FFRAME_DUMP_CYL_TRIAD

Cylinder Data

Steering	End	Area-in^2	Volume-in^3	Stroke-in
Bore = 4 in dia	Head	12.57	145.6	11.58
Rod = 2	Rod	9.43	114.9	12.18
Nominal length = 43.732 in				
Hoist				
Bore = 6	Head	28.27	56.54	2.00
Rod = 3	Rod	21.21	378.92	17.87
Nominal length = 35.766				
Dump				
Bore = 8	Head	50.26	1004.0	19.98
Rod = 3	Rod	43.20	640.54	14.83
Nominal length = 106.388				

#507100