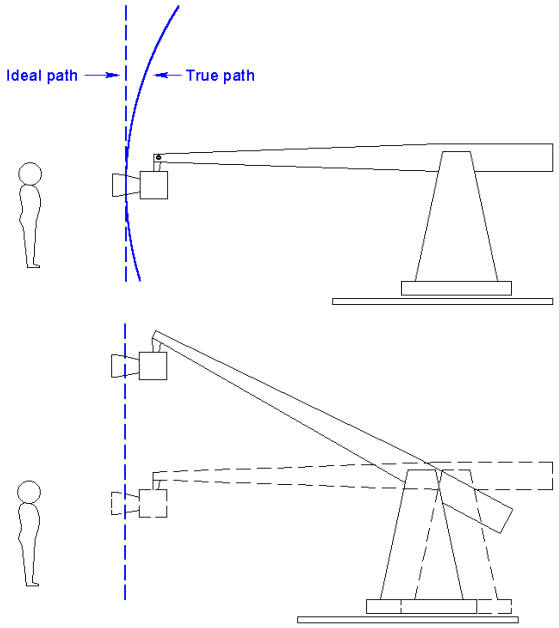
Non-Linear Boom Rigs in Kuper Virtual Axes

Swinging boom designs have several advantages for motion control rigs. For machines of medium to large size, they're much lighter and more compact than Cartesian rigs. Rotary bearings are much cheaper than linear rails, and the machine sweeps less floor space, reducing interference with light stands and sets.

On the other hand, swinging booms have one big disadvantage. It's really difficult to draw straight lines with a swinging arm.

We can easily illustrate this problem by imagining a writing straight vertical move using the arm. As the arm rises or falls, the arc it makes tends to fall back away from the subject.



Naturally, we could (and do) correct this by moving the machine back in toward the subject, in effect "tweaking out" the arc.

This is, however, somewhat tedious on complicated moves, so ideally, we'd like this process to happen automatically and transparently.

That's the basis of the Kuper "virtual axis" strategy. You describe the machine mathematically to the Kuper software, and the software then continuously calculates the "tweaks" necessary to keep the camera moving in an idealized line.

Virtualizing a relatively "pure" motion control rig, where the swing and boom axes are driven through gears or belts and therefore have a constant ratio to the drive motors, is fairly straightforward.

Mathematically, these machines are refereed to as "linear" machines, since the motion of the axes have a constant ratio to motion of the motors. Since the angle of the real axes can be calculated from the motor positions alone, the software can back-calculate the real position internally using only trigonometry. The Kuper virtual modes "Boom swing and lift" and "Cartesian" are implemented in this fashion.

However, many boom-type machines are driven by lead-screws, and this makes them naturally "non-linear". In these machines, the change in boom angle is not constant with motor movement, and varies depending on where you are in the boom range.

To virtualize a machine like this, in place of trigonometry, the Kuper uses a table of correction coefficients. The Kuper interpolates the boom position based on a curve it derives from a handful of experimentally established points.

These points are determined by the operator through measurement, and stored in a file named “nsgeom.tab”. This file is loaded into the Kuper on start-up and used as the basis of a table file that drives the real-time correction process.

Preliminary setup

To set up a good virtual machine, you should start with a good setup for the real machine.

To minimize confusion when paging in and out of virtual axes with the jogbox, you should set up the major axes in the same order as the virtual axes. That way, jogbox buttons have mostly the same effect in real and virtual modes.

Inother words, you’d put track on axis 1, so that jogbox button 1 & 2 would change between real track and virtual track. Putting swing on channel 2 means that jogbox buttons 3 & 4 would change between real swing and virtual E/W, etc.

For a real machine the mapping usually ends up looking something like…

	Real Axis	Virtual Axis
Axis 1 / 49	Track	Vtrack
Axis 2 / 50	Swing	VE/W
Axis 3 / 51	Boom	VN/S
Axis 4 / 52	Pan	Vpan
Axis 5 / 53	Tilt	Vtilt
Axis 6 / 54	Roll	Vroll

Some machines, especially older ones, offer both a swing and orthogonal transverse axis. On these types of machines, you should decide which axis you’re going to use the most for east/west work. Typically this is the swing axis, since it has more range, but not always, since the transverse axis is usually captured rail and therefore smoother.

Logical though it may seem to have the swing and transverse axis next to each other in the axis list, but you should resist this temptation and banish the less used axis farther down the list to channel 15 or so. In actual programming, you will usually “set-and-forget” the second transverse axis anyway, so you want it somewhere where you don't have to think about it. This will place it “out of the way” and prevent needless confusion as you page back and forth between jogbox modes.

Pay attention to axis direction. For proper operation the real axes should move in the same direction as the virtual axes. Since you can’t really change the virtual axes, you’ll have to use that convention for the real axes too.

For reference, the traditional direction conventions are…

Track – Positive moving away from subject
Swing – Positive swinging camera right
Boom – Positive booming up
Pan – Positive panning right
Tilt – Positive tilting up
Roll – Positive with the image rolling CCW

If you're used to animating in a computer environment, you may find swing and pan to be "backwards", that's because Kuper follows the left-hand "east/west" conventions of optical printing, rather than the right hand conventions of CG 3-space.

Decide which units you're going to use for translational axes, and calibrate your rigs' real axes in using conventional methods. Use degrees for all rotational axes like swing and boom. Of course, you'll never be able to get an exact number for a degree of boom lift (that's the whole point of this exercise, the number is non-linear) but you can find an approximate value.

Swing zero is the point that brings an imaginary line through the swing axis and the camera pan rotator parallel to the track axis. Since most modern rigs have the pan rotator centered on the boom, this means that for most rigs the boom is running straight down the center of the track.

By convention, boom zero has usually been with the boom level, since it makes the math easy, but there's nothing sacred about this if for some reason you're forced to use some other position.

Whatever you pick, you're wedded to it as long as you want to use this particular table, so if you don't use boom level, choose carefully and make sure that wherever you put your zero, you can find it again after the home mark falls off.

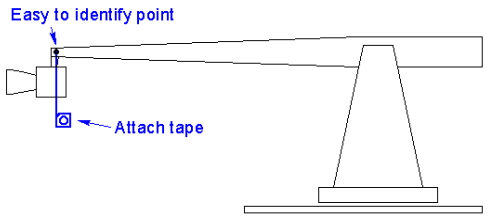
To find an approximate value for a degree of boom lift, boom down, -10° degrees from zero using an inclinometer, set your ppu's to 20 and zero your position. Now boom up to +10° degrees, again using an inclinometer Whatever you've got for a channel position is your new ppu count that approximates one degree of boom lift.

Write this number down, you'll use it later in making the table. Then re-zero your boom wherever you want it.

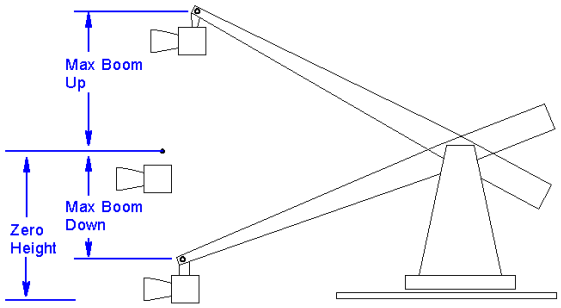
Measuring data for the geometry table

Most of the data from here on in derives from physical measurements of the rig. It's worth spending some time working out an efficient way to take the measurements of your particular rig, since the table will only be as good as the data you enter.

First, you'll need a consistent way to measure the height of the boom (and remember, it'll get high out of reach). One of the best methods is to clamp the end of a tape measure to the boom extension, and let it dangle like a yo-yo. Note that the tape doesn't have to be on the actual camera itself, a good, solid part that moves *with* the camera but isn't on the head, like a flat spot on the boom extension or pan rotator, is probably better.



Measure the boom height at zero. Then exercise the boom to find it's upper and lower range limits. On some machines, you may have to move down the track and work past the track end or even remove the camera head to find the practical lower limit. You're going want to include any boom excursion that you're likely to use in real-life in the table.



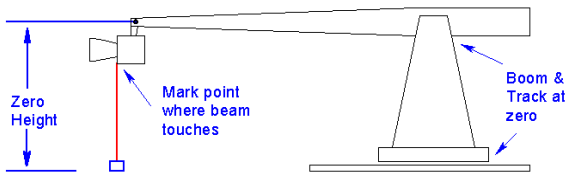
Calculate the range above and below zero and break it into 3 or 4 intervals each side of zero. On the rig used in this example, the upper excursion was 133cm, and the lower excursion was 138cm.

Break this into 4 intervals of roughly 30cm each (use round numbers, it's OK if the ones at the ends are slightly larger). They do not have to be even, and more than 4 is usually unnecessary, and actually counterproductive, since, like any table you tend to start introducing high-frequency noise when you get too many keyframes. The only restriction on the intervals is that they should be even integers – nothing but zeros after the decimal point.

Write down the intervals. For the the example rig, the intervals looked like this

```
Interval
+ 133 cm
 + 90 cm
 + 60 cm
 + 30 cm
    0 cm
 - 30 cm
 - 60 cm
 - 90 cm
- 138 cm
```

Now, take a laser plumb bob, place it in a spot on the floor where it will not be disturbed but where you can still get full vertical excursion (past the end of the track, maybe, or against the last cross-tie) and drive the rig up to it, marking the spot where it touches machine.

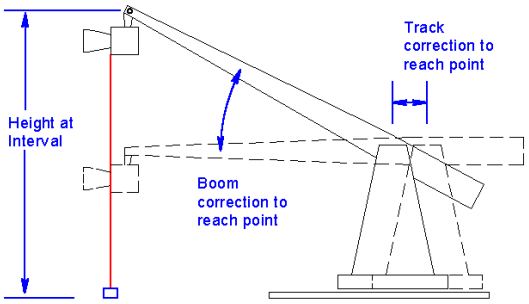


Carefully zero your track and boom and take one last measurement of the height of your reference point. For our CNC rig example, the zero height was an even 146cm. Add this zero height to each interval to calculate the how high your reference point would be above the floor at each of those intervals.

(this is all the math you'll ever have to have to do, really)

Height	Interval
279 cm	+ 133 cm
236 cm	+ 90 cm
206 cm	+ 60 cm
176 cm	+ 30 cm
146 cm	0 cm
116 cm	- 30 cm
86 cm	- 60 cm
56 cm	- 90 cm
8 cm	- 138 cm

Now, for each point in the height table, boom the rig up (or down) to the desired height, and drive the rig forward so that the laser dot hits the correct point again. Note the Track and Boom axis positions from the Kuper screen.



In our case, for the first point, 30cm, we had to boom up to 176cm above the ground. Once we did this, and corrected the track forward to return to the laser dot, we found we had a boom angle of +6.840 degrees and a truck forward of -1.961cm. These are the correction coefficients the Kuper would have to use if you told it to raise 30cm straight up.

Enter those correction coefficients in the interval table.

Height	Interval	Boom	Track
279 cm	+ 133 cm		
236 cm	+ 90 cm		
206 cm	+ 60 cm		
176 cm	+ 30 cm	6.840	-1.961
146 cm	0 cm		
116 cm	- 30 cm		
86 cm	- 60 cm		
56 cm	- 90 cm		
8 cm	- 138 cm		

Continue with the other 7 positions until you have filled the table (the zero row is 0,0,0, of course, since no correction is required)

Height	Interval	Boom	Track
279 cm	+ 133 cm	27.804	-40.449
236 cm	+ 90 cm	19.412	-17.757
206 cm	+ 60 cm	13.281	-7.906
176 cm	+ 30 cm	6.840	-1.961
146 cm	0 cm	0	0
116 cm	- 30 cm	-7.309	-1.860
86 cm	- 60 cm	-15.056	-7.396
56 cm	- 90 cm	-23.560	-17.002
8 cm	- 138 cm	-39.053	-41.930

You have just gathered the basic data for the nsgeom table, and it barely even hurt.

Creating the actual table file

To make the actual table file itself, you'll have to leave Kuper (but make sure you stay in the Kuper directory). The simplest tool to create table file is the DOS EDIT command.

For those rusty in the ways of DOS, make sure you're in the Kuper directory, and rename any nsgeom.tab file that's already there. Then type EDIT NSGEOM.TAB <enter>.

You'll open a basic but effective text-editing program. You can then type in your data. Unlike our paper example so far, a real nsgeom table has only three columns, interval, boom and track, and has no need for labels, so loose the "height" column and the labels. Formatting is not important to Kuper, but use spaces to keep it legible. Kuper does not use a "+"sign in front of positive numbers. Enter your data in columnar form.

```
  133      27.804     -40.449
  90       19.412     -17.757
  60       13.281      -7.906
   30       6.840      -1.961
    0           0           0
 - 30      -7.309      -1.860
- 60      -15.056      -7.396
 - 90     -23.560     -17.002
- 138     -39.053     -41.930
```

That's it for data. The only other entries needed are some comment lines. Any line starting with a ";" character is read as a comment, and will be displayed when the table is loaded as Kuper starts up. It's good form to add a comment line or two so the operator can make sure that the nsgeom table matches the rig he's using, since that data is normally not available inside the program.

I like to add one comment line that describes the machine and the units used,

```
; Rig # 1 calibrated in cm
```

And a second line that specifies the boom ppu used when you created the table (Geometry tables and the boom ppu must work together. In case something gets changed in the rig setup file and the boom ppu number gets lost, you'll need to find it before you can use the nsgeom table again)

```
; Use boom PPU = 4445.4
```

Our final, complete nsgeom table looks like this.

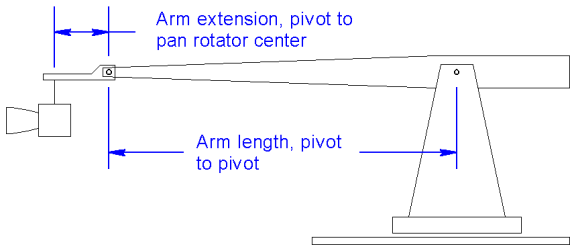
```
; Rig # 1 calibrated in cm
; Use boom PPU = 4445.4

   133      27.804     -40.449
   90       19.412     -17.757
   60       13.281      -7.906
    30       6.840      -1.961
     0           0           0
  - 30      -7.309      -1.860
 - 60      -15.056      -7.396
  - 90     -23.560     -17.002
 - 138     -39.053     -41.930
```

Save the file (alt-F) and exit the editor (alt-X).

Rig measurements (Nodal Parameters)

That's the first half of the process, creating the nsgeom table. For the second half, we go back into Kuper and set up the rig type and define some mechanical constants.

Measure the working length of the arm, from pivot center to pivot center.



Boom lift rigs typically have some sort of extension or mechanism at the end of the arm that keeps the camera level. Measure the horizontal distance between the arm end pivot center and the pan rotator axis.

For our example rig, the working length of the arm is 243.8cm, and the extension is 30.5cm.

Inside Kuper, in the function button area in the lower right of the main screen, click the hardware setup (HardSet) button.

In the hardware setup menu, click the "Setup Nodal Parameters" button. You will get a data entry form named "Nodal Parameters"

Select the rig type as Table N/S.

De-select all the other buttons, including the "Enable Model Compensation" button.

Enter the arm length in the "Boom Parallelogram Length" box.

Enter the arm extension in the "Arm Extension" Box, if there is no extension, enter 0.

Enter any lens nodal point offsets you might know at this time. Offsets are measured from the imaginary point where the pan, tilt and roll rotator axes cross in space.

Accept the changes, and close the box.

Nodal comp axes

The Kuper needs to know which real axes correlate to the virtual axes, that is, which major axis is really moving in the track direction, the swing direction, etc.

Inside Kuper, in the function button area in the lower right of the main screen, click the hardware setup (HardSet) button.

In the hardware setup menu, click the "Setup Nodal Comp Axes" button. You will get a data entry form named "Nodal Axis Compensators"

Read the directions, and assign the axes as necessary. If your rig is set up like most machines, the axis assignments will be straightforward. The Z axis, Virtual track, is derived mostly via real track, the X axis, Virtual EW, is derived mostly via real swing, and the Y axis, Virtual NS, is derived mostly via real boom.

The axes specified for "Model Compensation" don't matter, so long as compensation is turned off in the "Nodal Parameters" screen we looked at in the last paragraph. (If compensation isn't turned off, the Kuper assumes that you have a model mover that wants to maintain it's orientation relative to the camera, so those slave axes will move).

Arrange your axes as necessary, and close the box.

Save your data!

Don't forget to save the setups you've so carefully created. You've got nsgeom.tab on the disk already and hopefully after you did all the hardware setup in Kuper you remembered to use the "UtilFiles" button to save the current setup as the default, but those files are awfully easy to corrupt accidentally, especially the default setup.

For this reason, it's important that you make some duplicates of your configuration file and nsgeom table.

The configuration file is easy, after you've got all you setup menus entered in Kuper, you can use the "Save named setup" (under the "UtilFiles" menu) to save a backup copy of your setup with the rigs name, append an extension of ".set" so you can find it again (in our case, we'd use something like "Rig1.set").

Exit the Kuper software, and copy the nsgeom.tab file to a backup file with a more specific name.

The simplest way to do this is with the DOS COPY command, in this case something like COPY NSGEOM.TAB RIG1GEOM.TAB would make a backup copy with the name "rig1geom.tab"".