

# NTRT Scaling Analysis work

Andrew P. (Drew) Sabelhaus  
apsabelhaus@berkeley.edu

14 July 2014

This short white paper illustrates the scaling analysis that needs to be done on an NTRT-like system. This work is designed to convince NTRT/Bullet users that current scaling techniques work in some circumstances, but may have to be modified elsewhere (see: scaling of forces, scaling of angular vs. longitudinal spring constants.) First, we introduce a simple example system for analysis, then do the nondimensionalization of parameters, and evaluate which scalings need to occur. Then, simulations of the example system are shown under these scaling conditions. Once this has been established, the analysis is put in the context of NTRT simulations, and we show why current simulations have (luckily) worked. Finally, suggestions are made for generalizing these concepts to new parameters in the simulator.

## 1 Example system, angular spring/damper

In this example system, there is a rod on a hinge (hinge at (x,y) = (0,0) ) with length  $l$ . There is an angular (rotational) spring, and an angular damper, acting at the origin also. Gravity is present, and we assume it acts fully at the center of mass for the rod.

NOTE that this is NOT the same as the longitudinal/linear spring models that are used in NTRT. This specific system model is analysed first to emphasize that current assumptions don't work. See later sections for applications to our specific model, once you've convinced yourself that this dimensionless term analysis is necessary.

With the state variables for this system as the angle between the rod and the horizontal, and its derivative,

$$x = [\theta, \dot{\theta}]^T$$

We can derive the following system dynamics. (Note, these are NOT needed for scaling analysis, but are just provided for reference in case you'd like to simulate this yourself to check this math. I did!)

$$\sum T = I\ddot{\theta}, \quad \sum T = T_s - T_d + T_g$$

...where these torques are for the angular spring and damper, and the gravitational force. Their magnitudes are

$$T_s = k_\theta(\theta - \theta_r), \quad T_d = c_\theta\dot{\theta}, \quad T_g = (mg)(r) = \frac{mgl}{2}\cos\theta$$

Here,  $\theta_r$  is the rest angle of the spring. So, then, our rigid body dynamics are

$$\ddot{\theta} = \frac{1}{I} \left( \frac{mgl}{2}\cos\theta + k_\theta\theta - k_\theta\theta_r - c_\theta\dot{\theta} \right)$$

For a solid rod,

$$I = \frac{ml^3}{3}$$

And the full dynamics equation is

$$\ddot{\theta} = \frac{3}{ml^3} \left( \frac{mgl}{2} \cos\theta + k_\theta\theta - k_\theta\theta_r - c_\theta\dot{\theta} \right)$$

We will not return to this for the rest of the analysis, but it's useful for reference.

## 2 Π Terms and Nondimensional Numbers

I won't include a discussion/justification of NDMs here, see other work for that discussion - e.g. the undergrad fluids textbook by Munsen. It's sufficient to recount the following theorem, the Buckingham Pi Theorem, which states, loosely:

Given a system with  $n$  variables and parameters, of which there are  $k$  underlying basic units (phenomena), construct exactly  $n - k$  nondimensional numbers. Then, if the magnitudes of each of these dimensionless numbers is kept the same under scaling of some of the parameters, via scaling other parameters equivalently, then the kinematic and dynamic responses of the original system and the scaled system will be *similar* (e.g. equal.)

Usually,  $n$  includes both constants in the system as well as anything else that one might care about, such as initial conditions or output results (forces, trajectories.) Also, for physical rigid body dynamics,  $k$  is almost always the set of  $\{Mass, Length, Time\}$  or  $M, L, T$ , so  $k = 3$  moving forward.

### 2.1 System variables and units

The above example system has the following parameters and variables (both implicit and explicit) and the following units:

$$\text{mass: } m \sim M^1$$

$$\text{spring rest angle: } \theta_r \sim 0, \text{ because radians!}$$

$$\text{length of rod: } l \sim L^1$$

$$\text{gravitational acceleration: } g \sim L^1 T^{-2}$$

$$\text{angular spring constant: } k_\theta \sim M^1 T^{-2} L^1$$

$$\text{angular damping coefficient: } c_\theta \sim M^1 T^{-1} L^1$$

$$\text{force: } F \sim M^1 L^1 T^{-2}$$

$$\text{time: } t \sim T^1$$

These were derived with force in Newtons,  $kg * m/s^2$ .

Note that, unlike with longitudinal springs and dampers, their angular counterparts DO have a length dependence! This becomes important in a minute...

## 2.2 Generation of $\Pi$ -Terms

In order to make use of Buckingham's Pi Theorem to scale our system correctly, we need to make  $n - k = 8 - 3 = 5$  pi terms. I won't do that many until needed for illustrating a concept - I'll show the errors this generates as they occur.

The first step in this procedure is to select  $k = 3$  "repeating variables," which will make up our pi terms. Any subset of 3 is fine here, under the following conditions:

1. The set of repeating variables must not be able to make a nondimensional set by themselves (i.e. they must be an independent basis)
2. Avoid selecting difficult-to-scale variables as repeaters (e.g. don't choose time here because that's hard)

Let's choose, in this example, our repeaters as  $m, l, c$ . See the Munsen textbook (or other resources) to see how to procedurally derive all the pi terms. Skipping ahead, here are the ones I derived. You can check the units in all of these: they divide out evenly.

$$\begin{aligned}\Pi_1 &= \Pi_{\theta_r} = 0 \\ \Pi_2 &= \Pi_g = \frac{gm^2l}{c_\theta^2} \\ \Pi_3 &= \Pi_{k_\theta} = \frac{mlk_\theta}{c_\theta^2}\end{aligned}$$

## 2.3 $\Pi$ -Term Analysis for Scaling

So, if for this system, we want to scale gravitational acceleration by a factor  $s$ , such that

$$g_2 = (s)(g_1)$$

We would also need to scale length of rod (because of  $\Pi_2$ ) and angular spring constant (because of  $\Pi_3$ ) by the following:

$$k_{\theta_2} = (s)k_{\theta_1}, \quad l_2 = \frac{1}{s}l_1$$

However, there are other combinations we could choose! Pi term scalings are not unique. Here is another set of scaling factors we could choose, again assuming we want to scale gravity by  $s$ .

$$k_{\theta_2} = (s)k_{\theta_1}, \quad c_{\theta_2} = (\sqrt{s})c_{\theta_1}$$

...and we could come up with others involving the damping term, or involving all of them, if we wanted to. By the theorem, these scalings should all give us the same result.

### 3 Simulation of scaled systems

Figures 1 and 2 shows simulations of the original system, under parameters defined in the accompanying MATLAB file, as well as one with only gravity scaled by a constant (which is expected to yield a quantitatively different result.) Here, the scaling factor was  $s = 2$ . Figure 3 shows a simulation with length scaled with gravity, as per the naive assumption suggested by Bullet developers. This one is done with proportional scaling: gravity multiplied by  $s$ , length multiplied by  $s$ . However, to head off complaints, I've also included a plot where length is multiplied by  $1/s$  in figure 4. Both can be seen as incorrect scalings.

Figures 5 and ?? show two simulations of “properly scaled” systems under the two different parameter transformations listed above. In order to see differences in the following plots, pay attention to the magnitude of the first peak, as well as the settling time.

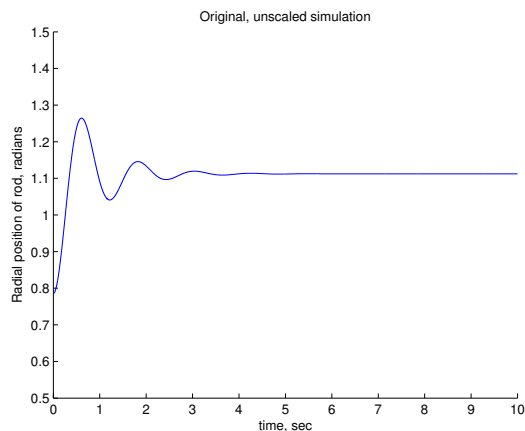


Figure 1: Original system, no scaling.

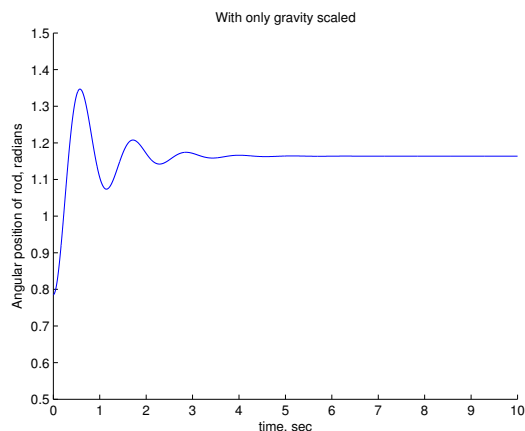


Figure 2: System with gravity scaled by 2.

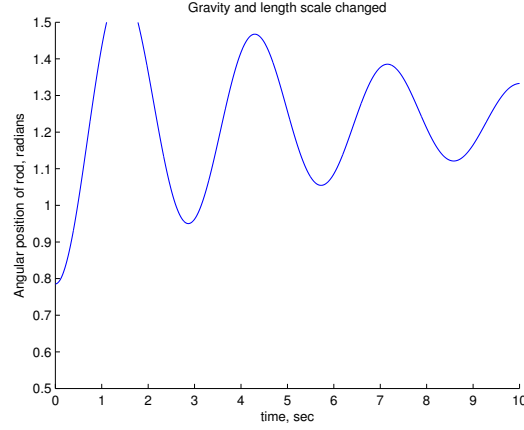


Figure 3: System with length scaled as well as gravity, both proportional with  $s$ .

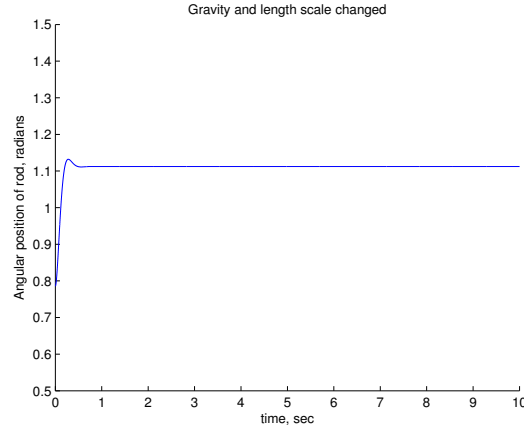


Figure 4: System with length scaled as well as gravity, inversely: gravity with  $s$ , length with  $1/s$ .

What happened? Plots 5 and 6 are NOT equivalent to the original in figure 1! For plot 5 there is likely something that I'm doing wrong with my simulation - I can't quite figure out why it's coming out like this. It's included for reference for someone else to figure out later if so desired.

But, plot 6 is expected, as per the analysis below. We messed up - we're not following the theorem exactly! We conveniently left out the two remaining Pi terms of force and time. Let's see if including these compels us to change our scalings. After some algebra, these two Pi terms are

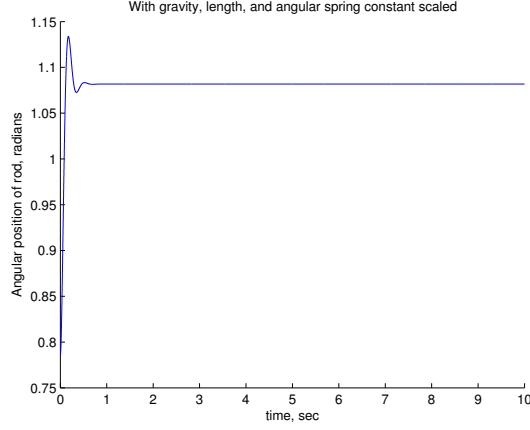


Figure 5: System with gravity, angular spring constant, and length scaled.

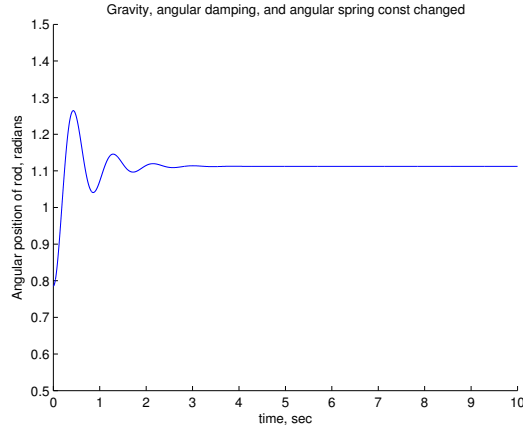


Figure 6: System with gravity and angular spring/damping constants scaled.

$$\Pi_4 = \Pi_F = \frac{fillinlater}{fillinlater}$$

$$\Pi_5 = \Pi_t = \frac{tc_\theta}{ml}$$

Ack! The time scaling! This gets complicated now: let's find a scaling that keeps time consistent also. Remember that time gets difficult for us to scale: we could do it post-hoc after the simulation, but that's a pain, so let's see if we can just change the constants.

...

...and it looks like this system of equations is unsolvable unless we scale time too. Darn. Here's a scaling with time, based on the original scaling:

$$k_{\theta_2} = (s)k_{\theta_1}, \quad c_{\theta_2} = (\sqrt{s})c_{\theta_1}, \quad t_2 = \frac{1}{\sqrt{s}}t_1$$

This system is simulated in figure 7. See how this is the same result as figure 1, finally!

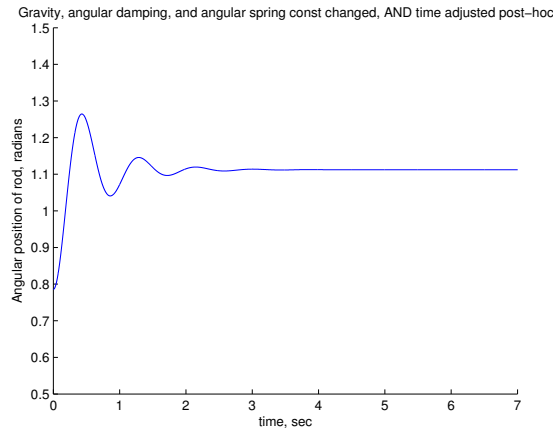


Figure 7: System with gravity, angular spring and damping constants, and time scaled.

In conclusion, this section should show by contradiction that the following two statements are untrue:

1. Commutativity of multiplication implies that we only need to scale length when scaling gravity:  $n * (length/time^2) == (n * length)/time^2$ .

If this were true, not only would figures 1 and 3 be identical, but there wouldn't be a need for dimensional analysis in the first place. The fact that the Buckingham Pi theorem exists in the first place really disproves this idea intuitively.

2. Instead of simulating a robot of characteristic size  $l$ , mass  $m$  in and gravity  $g$ , we simulate a robot of size  $10*l$ , mass  $m$  and gravity  $10*g$ . The resulting large robot has identical behavior to the original small robot, except for a multiplicative factor 10 for units which depend linearly on length, 100 for units which depend quadratically on length and so on.

This is not true, since we may need to pay attention to ALL variables at play. Some (seemingly unrelated) ones may cause significant changes in the required dimensionless terms, as explained in the next section on why our simulations currently "work".

## 4 Analysis of related longitudinal spring/damper systems, for NTRT use

The above analysis is based on a system where parameters are angular. However, our simulated systems use longitudinal/linear dampers and springs, and the analysis changes quite a bit. This is somewhat unexpected at first - why would the units of these extra parameters

influence the required scaling for unrelated terms? - but by the Pi theorem, it must work this way.

Ignoring a full dynamics derivation (no more simulations for me!), let's take the following constitutive equations for longitudinal springs and dampers:

$$F_s = k(x - x_r), \quad F_d = c\dot{x}$$

So, here is how the units and Pi terms would change for a longitudinal spring/damper system:

$$\text{spring rest length: } x_r \sim L^1$$

$$\text{spring constant: } k \sim M^1 T^{-2}$$

$$\text{damping coefficient: } c \sim M^1 T^{-1}$$

Note the missing length scale in  $k$  and  $c$ . This is because, for example,  $k$  needs to be units of  $N/m$  instead of  $N/rad$ , and the extra length term cancels out. The new Pi terms, using the same set of  $\{m, l, c\}$  are:

$$\Pi_1 = \Pi_{x_r} = \frac{x_r}{l}$$

$$\Pi_2 = \Pi_g = \frac{gm^2}{c^2 l}$$

$$\Pi_3 = \Pi_k = \frac{km}{c^2}$$

$$\Pi_4 = \Pi_F = \frac{Fm}{c^2 l}$$

$$\Pi_5 = \Pi_t = \frac{tc}{m}$$

Note that here, the scaling of gravity with length DOES actually work! When  $g$  is multiplied by  $s$ , we can do the same to  $l$ , then to  $x_r$  by  $\Pi_1$  and to  $F$  by  $\Pi_4$ . The takeaway here is that without extra Pi terms for more parameters like  $k_\theta$ , we can get away with this simple scaling, BUT we need to be careful and re-evaluate our Pi terms each time a new unit parameter is introduced.

## 5 Scaling Analysis for General Linear Systems

Finally, I'll finish with an argument as to why a truly linear system would not need any of this scaling analysis. I originally tried these simulations using a simple spring-mass-damper vertically oriented under gravity, e.g.

$$m\ddot{x} + c\dot{x} + kx = mg$$



With this system, the "gravity proportional to length" scaling also works, but for an entirely different reason. What's actually going on in this case is a change in the magnitude of the input-output transfer function, and so the scaling ITSELF is non-dimensional! One way to see this is to notice that the gravitational force is time-invariant, and could thus be replaced by a step input:

$$m\ddot{x} + c\dot{x} + kx = u(t)$$

By superposition of linear systems, the magnitude of the output of the system of

$$m\ddot{x} + c\dot{x} + kx = (n)u(t)$$

...will be exactly  $n$  times larger. Thus, dividing the output by  $n$  re-scales to the proper dimensions. I'm lazy and don't feel like looking up the theorems here, but it should be easy to justify if so desired.

## 6 Moving Forward

So, after all this work, where do we stand?

First, we need to check NTRT and Bullet for other parameters with other units, create Pi terms for them, and check to be absolutely sure we have the required number of dimensionless numbers. Coefficients that are dimensionless can be left out, so I'm not aware of other constants that would need to be scaled, but it's worth a hard look nonetheless. We should list all these, their corresponding Pi terms, and post the analysis somewhere so we can justify we're scaled correctly.

Second, we should look through old results and make sure we scaled (for example) output forces after the simulation. This would be relevant for the IPPW paper and Atil's learning results with the original SUPERball model.

Third, we should alert users to situations where their scaling might get crazy messy, and things to watch out for. Two situations come to mind. As one example, developing a model-based controller with parameters calculated analytically would have to be scaled. A controller using nonlinear damping techniques would require a complete re-derivation of these Pi terms and scaling. Another situation would be the creation of a new NTRT object, like a new model, which has constitutive equations that use new constants and terms. One example here would be the new string model that Brian Tietz Mirletz is working on.

Finally, all of this should be formalized and integrated into Bullet and/or NTRT in as scalable a way as possible. I'm not sure if there are automated techniques to developing Pi terms and scaling (?), but if those exist, it would be great to explore them. Nonetheless we need to write good code to perform these scalings behind-the-scene.

If anyone would like to run my MATLAB code (attached in this directory), it should be straightforward, but email me if you find issues.