Scaling Analysis for Rigid Body Simulations

Andrew P. (Drew) Sabelhaus

apsabelhaus@berkeley.edu

Started 14 July 2014

Current as of 27 Dec 2014

This short white paper illustrates the scaling analysis that needs to be done on an rigid body simulation system, such as the NASA Tensegrity Robotics Toolkit (NTRT or its C++ library, NTRTsim) and the Bullet Physics system. This work is designed to convince NTRT/Bullet users that care must be taken for scaling techniques, moreso than occurs on the Bullet Physics forums or wiki. In particular, simple scalings work in some circumstances with certain types of simulated parameters, 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 anaysis is put in the context of NTRT simulations, and we show why simple simulations have (so far) worked in NTRTsim and where they may become improperly scaled in future work. 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,

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

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, θ_r is the rest angle of the spring. So, then, our rigid body dynamics are

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

For a solid rod,

$$I = \frac{ml^2}{3}$$

And the full dynamics equation is

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

Note that this system has a nonlinear forcing function: the gravitational force has a $cos(\theta)$ term in it. Although the plant can be formulated as a linear system, the whole system response cannot be. Thus, unlike the example linear systems in the last sections, arguments about transfer function magnitudes do not apply here in the same way.

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

2   Π Terms and Nondimensional Numbers

For an in-depth discussion of the necessity of Pi terms, and their use, history, and applications, see references such as chapter 7 of the undergraduate fluid mechanics textbook by Munsen [2] and Buckingham's original work on the subject [1]. 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:

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

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

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

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

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

$$\text{angular damping coefficient:} \quad c_\theta \sim M^1 T^{-1} L^2$$

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

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 Π-Terms

In order to make use of Buckingham's Pi Theorem to scale our system correctly, we need to make $n - k = 7 - 3 = 4$ 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, k_θ. See the Munsen textbook (or other resources) to see how to procedurally derive all the pi terms. Here are the first 3 of the 4 terms that would be needed. Again, I'm doing this erroneously on purpose for example purposes. You can check the units in all of these: they divide out evenly.

$$\Pi_1 = \Pi_{\theta_r} = 0$$

$$\Pi_2 = \Pi_g = \frac{gml}{k_\theta}$$

$$\Pi_3 = \Pi_{c_\theta} = \frac{c_\theta^2}{ml^2 k_\theta}$$

2.3 Π-Term Analysis for Scaling

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

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

We would also need to scale length of rod (because of Π_2) and angular damping constant (because of Π_3) by the following:

$$l_2 = \frac{1}{s}l_1, \qquad c_{\theta_2} = \frac{1}{s}c_{\theta_1}$$

However, there are other combinations we could choose! Pi term scalings are not unique. One could choose other sets of scalings but I'll leave that as an exercise for the reader! By the theorem, these scalings should all give us the same result.

3 Simulation of scaled systems

In order to verify that following these scalings is actually required, here are a handful of simulations.

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. As expected, the latter figure shows an inherently different result. Here, the scaling factor was $s = 2$.

Figure 3 shows a simulation with length scaled with gravity. This one is done with proportional scaling: gravity multiplied by s, length multiplied by s. Note that this is similar to the assumptions passed around on the Bullet Physics forums: when scaling gravity, the only other scaling that's needed is length. Both can be seen as incorrect scalings.

Then, figure 4 shows the same system but this time following the three Pi-terms above. Specifically, all of g, l, and c_θ are scaled.

What happened? The plots of Figures 2 and 3 are wrong, as expected, but so is Figure 4! It's not the same as Figure 1!

As designed by this tutorial, plot 4 shows what happens when not all Pi-terms are calculated or followed. We're not following the theorem exactly! We conveniently left out the Pi-term corresponding to the time scale. It's possible that this requires scaling also. Let's see if including this compels us to change our scalings or to scale additional parameters. After some algebra, this Pi-term is

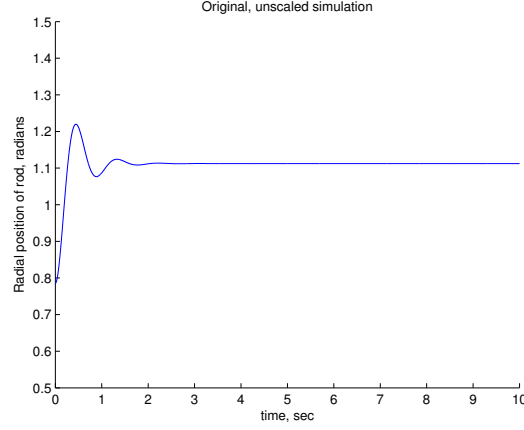$$\Pi_4 = \Pi_t = \frac{t^2 k_\theta}{ml^2}$$

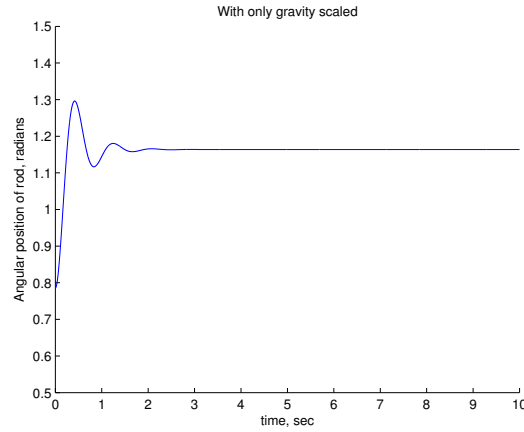Figure 1: Original system, no scaling.

Figure 2: System with gravity scaled by 2.

Ack! The time scaling! This gets complicated now: ideally, we'd want to find a scaling that doesn't require a change to the time axis - this makes real-time simulations particularly difficult to interpret. However, a brief bit of investigation leads us to believe that any possible scaling for this system that involves a gravity term will also necessarily involve a time term. A future task is to show that this is true by either exhaustive search of Pi-term options or by finding other work that would show it as such.

Here's our final scaling with time:

$$l_2 = \frac{1}{s}l_1, \qquad c_{\theta_2} = \frac{1}{s}c_{\theta_1}, \qquad t_2 = \frac{1}{\sqrt{s}}t_1$$

This system with all four terms scaled is simulated in figure 5. See how this is the same result as figure 1, finally!
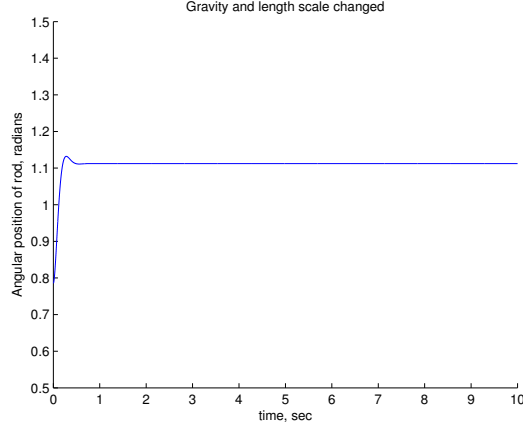
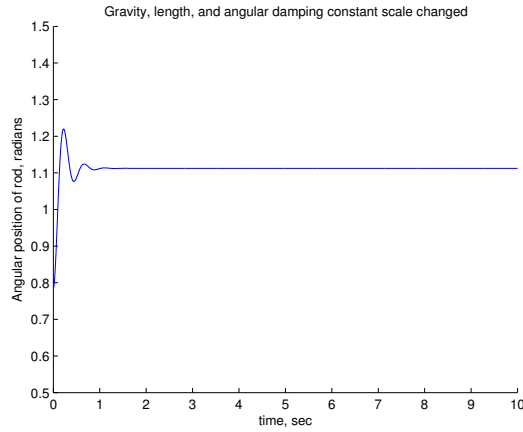Figure 3: System with length scaled as well as gravity, both proportional with s.

Figure 4: System with gravity, angular damping constant, and length scaled.

3.1 Discussion

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, e.g., only scaling length will result in similitude. $n*(length/time^2) == (n*length)/time^2$.

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. This conjecture relies on a notion of commutativity of multiplication for a single equation - that involving only length and gravity - and neglects equations that might involve gravity and other terms, or length and other terms. This is the motivation behind the Buckingham Pi Theorem: such a procedural method ensures that all such interactions are accounted for, and the many potential multiplicative commutativity relationships are always satisfied.
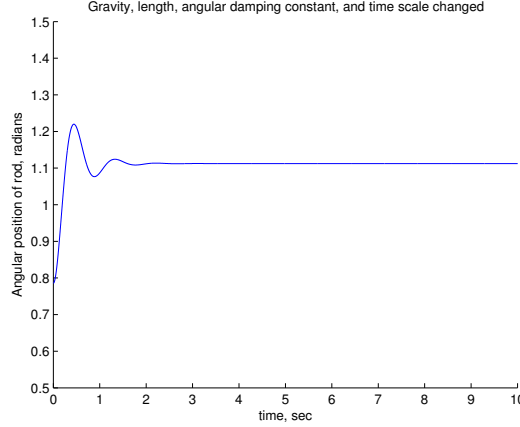
Figure 5: System with gravity, angular damping constants, and time scaled.

2. Instead of simulating a robot of characteristic size l, mass m and gravity g, we simulate a robot of size 10*l, mass m and gravity 10*g. The result will be dynamically and kinematically similar (in the technical sense of similitude.)

Similar to the above reasoning, this neglects other possible variables in the system. While it is true that such a scaling is needed, this conjecture doesn't consider other possibly-necessary scalings. In particular, even something like mass may need to be scaled here depending on the situation and if the Pi-terms call for it. Care must be taken! However, note again that this does work for certain linear systems, such as those described with the simpler Pi-terms in section

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:

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

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

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

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 Π_1 and to F by Π_4. The takeaway here is that without extra Pi terms for more parameters like k_θ, 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 informal argument as to why a truly linear system would not need any of this scaling analysis. In brief, if input forces can be represented as time-invariant, and the Laplace transform of the system shows a scale of the input to correspond proportionally to an output magnitude change, then scaling is simple and intuitive. 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. This is, however, informal: more research must be done to show this result, and to review the literature for similar work or proofs.

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.

References

[1] Edgar Buckingham. On physically similar systems; illustrations of the use of dimensional equations. *Physical Review*, 4(4):345–376, 1914.

[2] Bruce Munson. *Fundamentals of fluid mechanics.* John Wiley and Sons, Inc, Hoboken, NJ, 2013.