

Vortex Studio 2019b

Verification and Validation

The performance and accuracy of Vortex Studio have been carefully tested by carrying out analyses of problems with known analytical solutions. To provide confidence in the accuracy of simulations using Vortex Studio, this document presents a subset of the analyses performed.



TABLE OF CONTENTS

What’s New in This Document.....	4
1 INTRODUCTION.....	4
2 VERIFICATION OF VORTEX STUDIO FUNCTIONALITY.....	4
2.1 Verification Framework.....	5
2.2 Dynamics.....	5
2.2.1 Vortex Studio Integrator.....	5
2.2.2 Unconstrained Linear Motion: Zero Gravity.....	10
2.2.3 Unconstrained Linear Motion: Constant Gravity.....	10
2.2.4 Impulse Force Response.....	12
2.2.5 Mass-Inertia Matrix.....	15
2.2.6 Inertia Tensor Translation and Rotation.....	18
2.2.7 Friction.....	22
2.2.8 Contacts.....	26
2.3 Fluid Interaction.....	29
2.3.1 Added Mass Property.....	29
2.3.2 Restoring Torque due to Buoyancy.....	31
2.3.3 Drag and Lift.....	33
2.4 Dynamics Modules.....	36
2.4.1 Cable Systems: Catenary.....	36
2.4.2 Cable Systems: Pendulum.....	40
2.4.3 Cable Systems: Elongation and Bending.....	43
2.4.4 Cable Systems: Adaptive.....	48
2.4.5 Cable Systems: Fluid Interaction.....	50
2.4.6 Cable Systems: Ring.....	53
2.4.7 Vehicle Systems: Ackermann Steering.....	54
2.4.8 Vehicle Systems: Engine Torque Table.....	58
2.4.9 Vehicle Systems: Synchronizer.....	61
2.4.10 Vehicle Systems: Clutch.....	64
2.4.11 Vehicle Systems: Transmission.....	67
2.5 Constraints.....	72
2.5.1 VxBallAndSocket.....	73
2.5.2 VxDistanceJoint.....	73

2.5.3	VxHinge	75
2.5.4	VxHomokinetic.....	77
2.5.5	VxPrismatic.....	78
2.5.6	VxRPRO.....	80
2.5.7	VxSpring	82
2.5.8	VxUniversal	84
3	VALIDATION OF A VORTEX STUDIO-BASED SIMULATION.....	85
3.1	Aggregate Behaviors	86
3.2	Data-based and Statistical Analysis.....	86
3.3	Subject Matter Expert Review	86
3.4	Typical Tests during Validation	87
Contact	87

WHAT'S NEW IN THIS DOCUMENT

This section includes the summary of changes since the previous revision. The changes may include but are not limited to new experiments, amendments, and corrections applied to existing experiments.

Updates and addition of experiments:

- (new experiment) Contacts
- (update) Cable Systems: Catenary: updated in Vortex Dynamics 6.6.0
- (update) Cable Systems: Pendulum: Results
- (new experiment) Cable Systems: Ring
- Added Mass Property, Restoring Torque due to Buoyancy, and Drag and Lift — three experiments — are grouped to a new section: 2.3 Fluid Interaction.

1 INTRODUCTION

Simulations are numerical routines that mimic various features of real systems. Simulation applications must be carefully constructed with regards to the means and possibilities of verification and validation. Verification is a quality control process that ensures that the algorithms, routines and interfaces of Vortex Studio perform as intended. Validation is a quality assurance process on a Vortex Studio-based simulation that establishes fitness of purpose for some user-defined set of requirements.

Verification and validation procedures can overlap considerably. What is required is a structured comparison of the simulation's output. In verification this is usually against previously accepted mathematical equations, laws or formulas, or data available from carefully designed laboratory tests.

For validation of a Vortex Studio-based simulation, the data can be sampled in the field or laboratory. These tests are broader and are focused on aggregate behaviors of specific mechanisms. The values obtained will be different from the parameters used to define the vehicle as many actions, like braking distance, or maximal angle for road grip, emerge only when the vehicle or robot is actually operated.

Quality assurance is a top priority for CM Labs, and we make every effort to assist our clients in their validation efforts.

2 VERIFICATION OF VORTEX STUDIO FUNCTIONALITY

Verification of Vortex Studio is an internal process that tests Vortex algorithms and data structures to ensure conformance to expected results. The following areas are tested directly as rigid body mechanics:

- Energy stability of single and coupled rigid bodies using constraints
- Consistency of inertial properties over time

- Geometric properties, including overlap and contact creation
- Friction, impact, and other material properties
- Movement under gravity and potential forces

2.1 Verification Framework

In order to guarantee continuous verification of Vortex Studio algorithms, CM Labs has developed an internal verification test framework called *VxVerification*. Written in Python, it is used for verifying the Vortex Studio API and integrated modules by comparing them against underlying physics models and previous run baselines. A collection of Python test scripts in the automatic regression framework runs after code compilation. The test framework is used to develop a large set of standardized tests for Vortex Studio. Failed or modified modules are flagged for further review by developers to ensure accuracy.

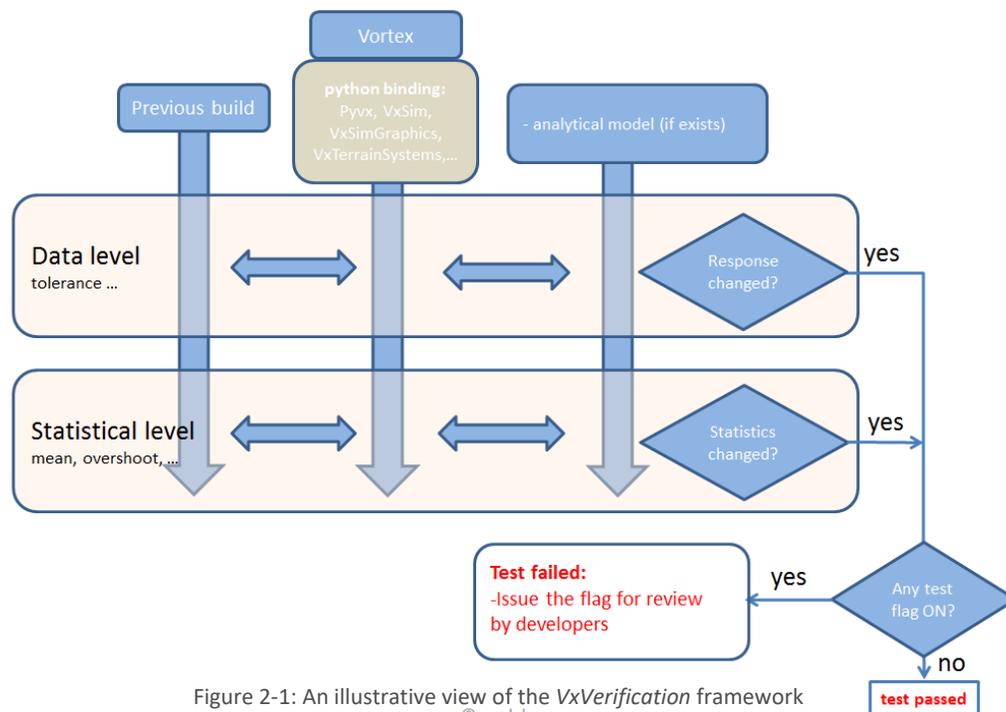


Figure 2-1: An illustrative view of the *VxVerification* framework
© cm-labs

2.2 Dynamics

This section includes the experiments deployed to verify the Vortex Studio module. Verification is done by observing the correctness of emerging behaviours and properties such as linear motion, force responses, inertia tensor, buoyancy, and friction.

2.2.1 Vortex Studio Integrator

Test Case

To derive the velocity for the next frame from the current acceleration, Vortex Studio uses an explicit Euler method, and then it computes the position for the next frame by using an estimated velocity for the next frame. Refer to (2.16) and (2.17).

The overall integration scheme is a semi-implicit symplectic integrator, which encapsulates aspects of both explicit and implicit methods. This type of integrator can yield better results as it avoids adding energy to the system¹.

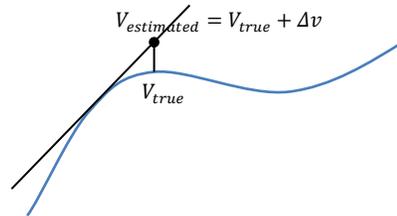


Figure 2-2: Error is introduced during velocity estimation

In a standard Euler method, the energy increases² steadily since each step may introduce a small error of Δv to the true velocity V_{true} . It results in a **second-order increase** in the energy if uncorrelated with estimated velocity $V_{estimated}$, which is usually the case for simple integrators.

$$\begin{aligned}
 E_{estimated} &= 0.5m \sum V_{estimated}^2 \\
 &= 0.5m \sum (V_{true} + \Delta v)^2 \\
 &= 0.5m \sum (V_{true}^2 + \Delta v^2 + 2V_{true} \cdot \Delta v) \quad (2.1)
 \end{aligned}$$

The term $2V_{true} \cdot \Delta v$ yields to zero if there is no correlation between true velocity and the error, and the estimated energy deviates from its true value.

$$E_{estimated} = 0.5m \sum V_{true}^2 + 0.5m \sum \Delta v^2 \quad (2.2)$$

In contrast, the symplectic integrator scheme features small but constant increases in the energy over a very long time³.

Experiment and Methodology

Liouville's theorem—a key theorem in Hamiltonian mechanics—asserts that *the phase-space distribution function is constant along any trajectory in the space*⁴⁵. In other words, as the trajectories of a cloud of points

¹ http://en.wikipedia.org/wiki/Semi-implicit_Euler_method

² In a discrete simulation, energy drift can be attributed also to other parameters such as imperfection in evaluating energy function and machine precision/rounding-off of the data.

³ http://en.wikipedia.org/wiki/Energy_drift#cite_note-Gans-1

stretch through phase space in one coordinate—say, position—it shrinks in the other direction—momentum in this case—so that the product area (position*momentum) remains unchanged. Since Hamiltonian mechanics comes from the symplectic structure of the Hamiltonian systems⁶, the theorem lays the foundation to establish an experiment to test if the integrator is symplectic or not.

As an initial experiment, an ensemble of eight horizontal oscillators is configured with different initial <position, momentum> states forming a rectangle in the position-momentum plane. As the simulation advances, the time-evolution of the rectangle shows how the energy in the system changes accordingly. The chosen mass and spring parameters are shown below:

Sphere mass: 5.0 kg
Spring length: 3.0 m
Spring damping: 0.0 Ns/m
Spring stiffness: 50.0 N/m

Results

⁴ http://en.wikipedia.org/wiki/Liouville%27s_theorem_%28Hamiltonian%29

⁵ <http://www.av8n.com/physics/liouville-intro.htm>

⁶ http://en.wikipedia.org/wiki/Hamiltonian_mechanics

test_phase_space_evolution_of_ensemble_of_eight_oscillators

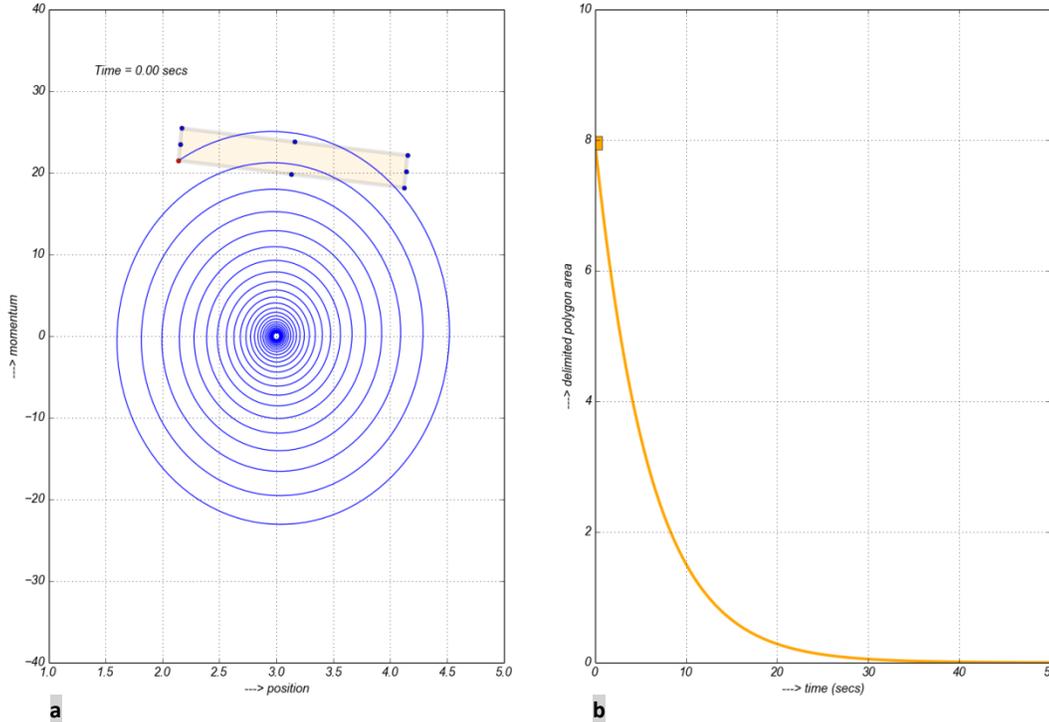


Figure 2-3: position-momentum plot and its time-evolution for a set of oscillators

The rectangle in Figure 2-3a represents initial state of the ensemble. Each dotted corner corresponds to an oscillator. As the simulation advances, observe the trajectory progresses through a spiral path (for clarity, only the trajectory of first oscillator represented by red dot is shown in the figure). In an ideal symplectic integrator:

- Each of the dots should have a circular trajectory. This would confirm that the equations of motion conserve energy. Even though each dot does not keep itself in a constant distance from the center, which typically occurs in a symplectic integrator, it should return to the same point following the same initial trajectory.
- The region delimited by the ensemble should maintain a constant area. This is connected to the conservation of area in phase space – Liouville’s theorem. If not, it would mean that on a step-by-step basis the integrator doesn’t conserve energy.

Looking at first set of results, the Vortex Studio integrator is at least not explicit in nature since it is not exploding. However, rather than returning to the initial starting point, it is falling into a spiral path by losing energy (with exponential decay mean life — time the function decays to 36.78% of its maximum — of ~7.5 seconds). Considerable decay rate is understandable because the experiment involves kinematics, a spring constraint, which expedites the decay process by contributing as an external energy sink.

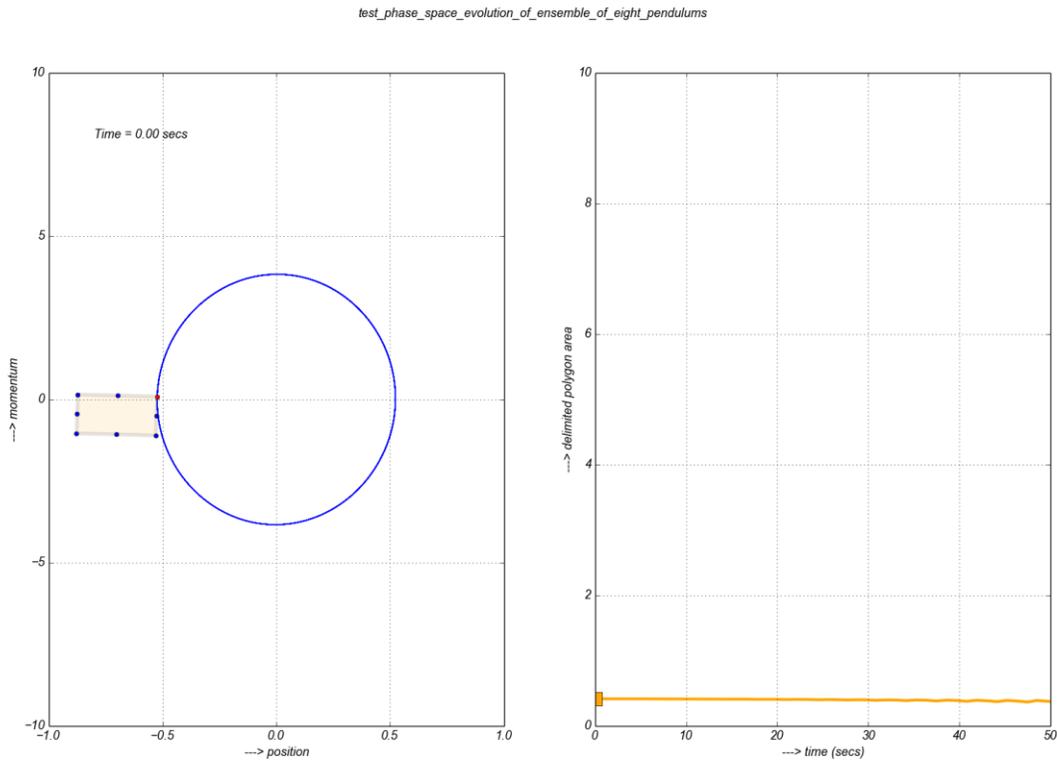


Figure 2-4: Position-momentum plot and its time-evolution for a set of pendulums.

Now, the second experiment is designed using minimal kinematics: a pendulum configuration using ball-and-socket constraint, assuming that the constraint contributes negligible amount of energy to the system. The results are encouraging that the system not only conserves energy but also does on step-by-step basis (Figure 2-4). You can see that the system energy remains unchanged even after 50 seconds of simulation.

2.2.2 Unconstrained Linear Motion: Zero Gravity

Test Case

A point particle moving from rest with an initial impulse and no gravity or other applied forces should move with constant velocity.

Model

According to Newton's first law, a point mass will move with constant velocity until acted on by a force. So, in the absence of gravity or other applied forces, a particle at rest given an initial impulse will move away from its initial position at constant velocity. The point particle is modelled as a *VxPart*; the motion of the centre of mass of a rigid body behaves as a particle.

Results

Apply impulse $\langle x = 1, y = 2, z = 3 \rangle$ at time 0 to particle of mass 1:

time: 0, velocity: $\langle x = 0, y = 0, z = 0 \rangle$	position: $\langle x = 0, y = 0, z = 0 \rangle$
time: 1, velocity: $\langle x = 1, y = 2, z = 3 \rangle$,	position: $\langle x = 1, y = 2, z = 3 \rangle$
time: 2, velocity: $\langle x = 1, y = 2, z = 3 \rangle$,	position: $\langle x = 2, y = 4, z = 6 \rangle$
time: 3, velocity: $\langle x = 1, y = 2, z = 3 \rangle$,	position: $\langle x = 3, y = 6, z = 9 \rangle$

The output from Vortex Studio is exactly as expected: the impulse at time zero results in a constant velocity from rest; the velocity is equal to the impulse divided by the mass; and the velocity produces a constant change of position between steps. So, under zero gravity the impulse response is correct.

2.2.3 Unconstrained Linear Motion: Constant Gravity

Test Case

A point particle moving from rest under constant gravity

Model

A particle with position q moving under constant gravity g undergoes a constant acceleration

$$\ddot{q} = g \quad (2.3)$$

The dot notation implies the rate of change. The solution for the motion is found by integrating twice.

With zero initial conditions $q(0) = 0$ and $\dot{q}(0) = 0$, the integration constants vanish to give

$$q = \frac{1}{2}gt^2 \quad (2.4)$$

Results

Timestep $h = 1$ and gravity $g = -10$ are chosen.

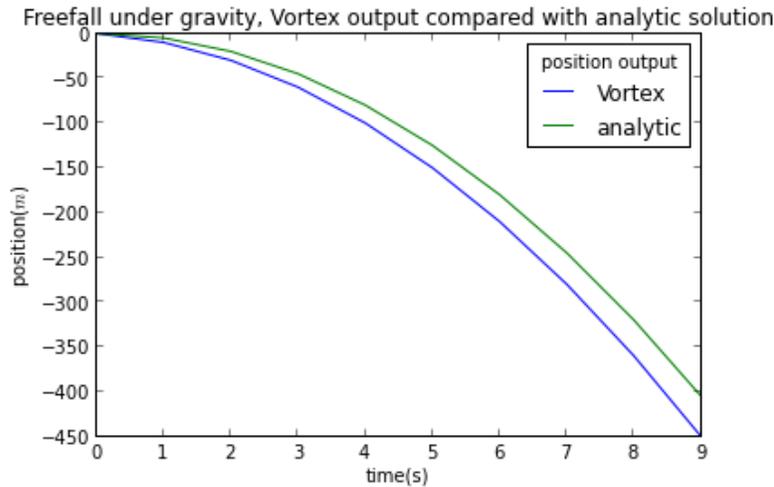


Figure 2-5: A linear motion under constant gravity

The analytic solution for position, $q = \frac{1}{2}gt^2$, sampled at times $t = nh$, is the sequence: $0, -5, -20, -45, -80, -125, -180, -245, -320, -405$. This sequence is proportional to the 'half-square' numbers $\frac{n^2}{2}$ (multiplied by gh). The Vortex Studio position output is the sequence: $0, -10, -30, -60, -100, -150, -210, -280, -360, -450$. This is proportional to sequence: $0, 1, 3, 6, 10, 15, \dots$ --the 'triangular' numbers $\frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2}$. The difference, proportional to $\frac{n}{2}$, is due to discretization error, as explained below.

Assumptions/notes

Vortex Studio, like other simulators, computes an approximation to the motion of a mechanical system by numerically integrating its equations of motion. This process inherently involves a discretization in which the continuous equations of motion are replaced by a discrete-time approximation according to a chosen time stepping scheme.

This section explains the behaviour of the stepping scheme used within Vortex Studio for the case of a particle falling under gravity. The equation of motion $\ddot{q} = g$ is a single second order equation. Introducing a variable v for velocity allows splitting the second order equation into a pair of first order equations $\dot{q} = v, \dot{v} = g$, the usual starting point for discretization.

The continuous equations of motion on the left below are approximated within Vortex Studio by the discrete stepping scheme on the right:

$$\begin{aligned} \dot{v} &= g & \Rightarrow & \frac{v_{n+1} - v_n}{h} = g \\ \dot{q} &= v & & \frac{q_{n+1} - q_n}{h} = v_{n+1} \end{aligned} \quad (2.5)$$

Here, h represents the timestep and the subscripts on the discrete variables represent the time in units of timestep; for example v_n approximates $v(t_n)$ with $t_n = nh$. The time stepping scheme above is termed semi-implicit; it is explicit in the velocity update and implicit in the position update. The discrete velocity update equation is

$$v_{n+1} = v_n + hg \quad (2.6)$$

With zero initial condition, $v_0 = 0$, it has the solution $v_n = nhg$. In this case values of the discrete velocity are equal to values sampled from the exact solution; $v_n = v(t_n) = gt_n$. Now for the discrete position solution, the update equation is $q_{n+1} = q_n + hv_{n+1}$. This is simplified by substituting $v_{n+1} = (n+1)hg$ (the velocity solution for the next step) to give $q_{n+1} = q_n + h^2(n+1)g$. With initial condition $q_0 = 0$, the solution is

$$q_n = h^2g \frac{n(n+1)}{2} = \frac{1}{2}gt_n^2 + \frac{1}{2}gt_nh \quad (2.7)$$

where the final term $\frac{1}{2}gt_nh$ is the discretization error. The error is proportional to simulation time t_n and the timestep h . The simulation trajectory is a parabolic 'ballistic' curve (Figure 2-5) identical to the exact solution but shifted a little earlier in time and appearing to start from a slightly higher point in space. For visual simulation purposes the difference is hard to perceive.

2.2.4 Impulse Force Response

Test Case

An impulse force should impose a stationary object to move with a constant velocity.

Model

Time integration of the forces applied on a body during its trajectory gives its momentum.

$$\int_0^t F(t)dt = mv(t) + \text{initial momentum} \quad (2.8)$$

For an impulse force as input, i.e. $F(t) = K\delta(t)$, the integration results in a step-momentum response.

$$\int_0^t F(t)dt = K \quad (2.9)$$

As a result, the object attains a constant velocity of IMPULSE/EFFECTIVE MASS.

$$mv(t) = K \Rightarrow v(t) = K/m \quad (2.10)$$

This is a case of linear motion in a plane with no gravity and no couplings to other degree of freedoms (DOFs).

Results

- a) Negative added mass case: Vortex Studio replaces all added mass coefficients smaller than 1.0 by 1.0 in order to avoid negative force due to added mass.
- b) Zero added mass case: when the impulse is applied, the velocity goes straight to impulse/ mass on the next time step. The velocity response is as close to a step response as it can be - the correct response.
- c-d) The other plots have added mass in increasing proportions. The plot shows that a body with added mass over-reacts to impulsive forces, moving off too quickly when the impulse is applied and then only gradually slowing down to the correct velocity. The overshoot gets worse as the added mass gets larger relative to the body.

This will not be a surprise to those who know the implementation: added mass is simulated via an applied force which is computed based on an estimate of acceleration extracted from a linear Infinite Impulse Response (IIR) filter of the velocities. More analysis of the IIR filter to show the exact stability criteria can be done. It can be seen in Figure 2-6d, where the added mass is much larger than the body mass (analogous to simulating something like a bubble), that the response eventually oscillates towards instability. As an exercise for the reader try simulating a bubble in the editor. In ROV applications, added mass is chosen around a half of the body mass so that the overshoot is not too pronounced.

Added mass is not taken into account at all in the first time step. Then, the effect of the added mass is gradually incorporated to bring the velocity back down close to what it should be.

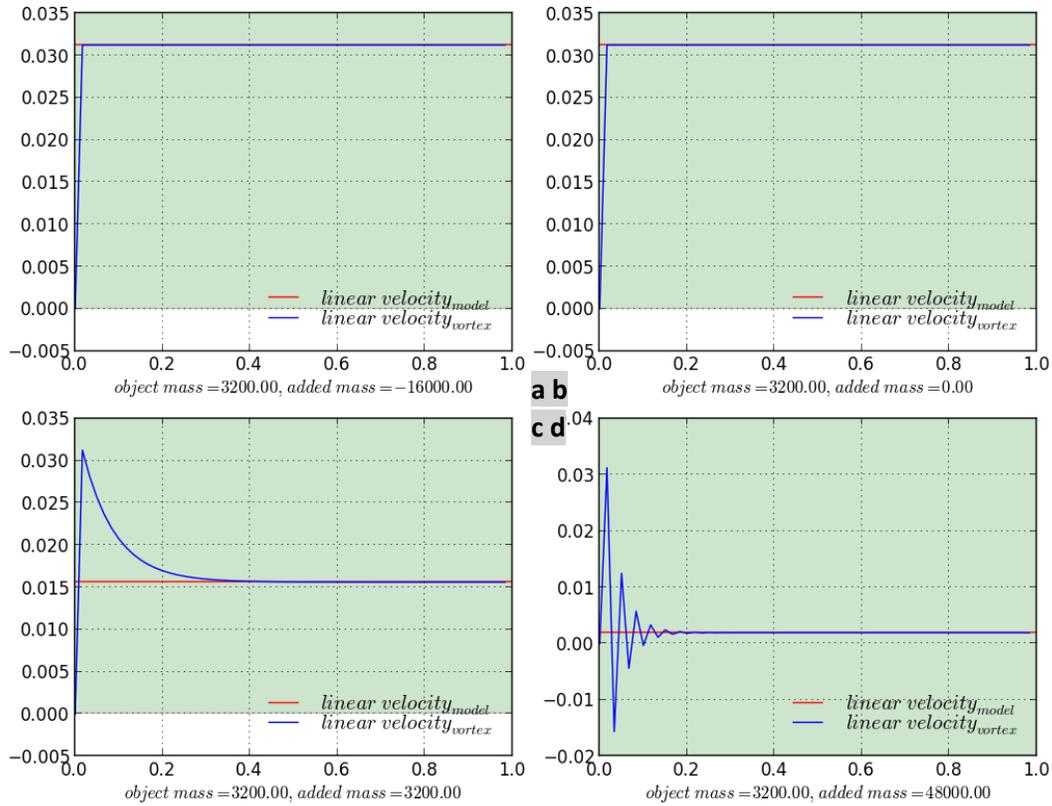


Figure 2-6: Validating dynamics by verifying correctness of impulse force response: constant velocity response due to an impulse force

2.2.5 Mass-Inertia Matrix

Test Case

The objective of this experiment is to show and verify that the fluid interaction forces, added mass and propeller effect can be parameterized through the user defined mass-inertia matrix which is exposed at the *VxPart* level in the API.

Context

A rigid body in three-dimensional space can be fully constrained by fixing its six degrees of freedom: three rotational and three translational. The Newton–Euler equations, in classical mechanics, describe the combined translational and rotational dynamics of a rigid body.⁷

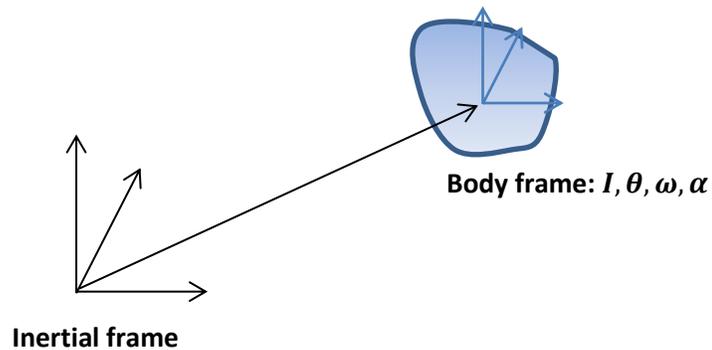


Figure 2-7: A rigid body and its two reference frames

Net force acting on a body causes the acceleration (2.33). Combining linear and angular parameters together with the associated mass and inertia, the Newton-Euler equation for a single rigid body can be written in a matrix form:

$$\begin{pmatrix} \mathbf{F} \\ \boldsymbol{\tau} \end{pmatrix} = \begin{pmatrix} m\mathbf{1} & 0 \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} 0 \\ \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \end{pmatrix} \quad (2.11)$$

where,

- \mathbf{F} – Net force acting on centre of mass (COM)
- $\boldsymbol{\tau}$ – Net torque acting on COM
- \mathbf{a} – Linear acceleration about COM
- $\boldsymbol{\alpha}$ – Angular acceleration about COM
- $\boldsymbol{\omega}$ – Angular velocity
- m – Mass of the body
- $\mathbf{1}$ – 3X3 Identity matrix
- \mathbf{I} – 3X3 inertia tensor computed about COM

We call $\mathbf{M} = \begin{pmatrix} m\mathbf{1} & 0 \\ 0 & \mathbf{I} \end{pmatrix}$ the mass-inertia matrix as it contains the mass and inertia information. Among the thirty six elements in the mass-inertia matrix, by default, only twelve of them are filled. We assume that the

⁷ Hubert Hahn (2002). *Rigid Body Dynamics of Mechanisms*. Springer. p. 143. ISBN 3-540-42373-7.

mass is a constant and does not depend on reference frame and that the linear and angular accelerations are not coupled.

$$\mathbf{M} = \begin{bmatrix} m & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & m & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & m & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & I_{ii} & I_{ij} & I_{ik} \\ \cdot & \cdot & \cdot & I_{ji} & I_{jj} & I_{jk} \\ \cdot & \cdot & \cdot & I_{ki} & I_{kj} & I_{kk} \end{bmatrix} \quad (2.12)$$

The Vortex Studio API, however, offers flexibility by providing an interface to edit any element of the matrix. The user can address their kinematics needs such as added mass. For example, let the user input added-mass tensor be \mathbf{M}_{user} , then the mass-inertia matrix in equation (2.12) is replaced internally by $\mathbf{M}_{kirchoff} = \mathbf{M} + \mathbf{M}_{user}$ which is also called Kirchhoff tensor⁸.

The inertia tensor in equation (2.11) is defined in the local frame about part COM. Rotation of the inertia tensor and Kirchhoff tensor in each simulation frame has to be reflected when it is referred from the global frame

$$\mathbf{I}_R = \mathbf{R}\mathbf{I}\mathbf{R}^T \quad (2.13)$$

$$\mathbf{M}_R = \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \mathbf{M}_{kirchoff} \begin{pmatrix} \mathbf{R}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^T \end{pmatrix} \quad (2.14)$$

where rotation matrix \mathbf{R} is computed in each simulation frame from the instantaneous angular position (2.18), the acceleration is computed by using the matrix operation

$$\mathbf{A}_i = \mathbf{M}_R^{-1}\mathbf{F} - \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\omega} \times \mathbf{I}_R \boldsymbol{\omega} \end{pmatrix} \quad (2.15)$$

Then velocity is estimated, followed by the position:

$$\mathbf{V}_{i+1} = \mathbf{V}_i + dt.\mathbf{A}_i \quad (2.16)$$

$$\mathbf{P}_{i+1} = \mathbf{P}_i + dt.\mathbf{V}_{i+1} \quad (2.17)$$

⁸ Underwater Rigid Body Dynamics, by Weissmann Steffen and Pinkall. SIGGRAPH 2012 Conference -Volume 31, Issue 4, July 2012, Article No. 104

Note that the velocity for the next frame depends on the previous state – explicit Euler method – (2.16) and position for the next frame depends on the next frame’s velocity – implicit Euler method⁹. Collectively it is named as a semi-implicit scheme.

Experiment

A rigid box is fully submerged in water with the following parameters. It is simulated with the given initial state, position, and velocity. Each coordinate data is logged throughout the simulation.

Dimension: 1.0x2.0x0.5 m3
 Density: 7870 kg/ m3 (~ aluminum)
 Fluid density: 1000 kg/ m3 (~ water)
 Initial linear & angular positions: <100, 100 100> & <0, 0, π/6>
 Initial linear & angular velocities: <20, 20, 20> & <0.0, 0.0, 0.0>
 Added mass & coupling amount: 10.0 & 100.0

To inject added mass through the mass-inertia matrix and to simulate linear and angular (along z-axis) accelerations coupling, all fluid properties such as buoyancy, drag, added mass and lift force are disabled and mass-inertia matrix is modified to $M_{equivalent}$ as

$$\begin{bmatrix} m + 10 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & m + 10 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & m + 10 & \cdot & \cdot & 100 \\ \cdot & \cdot & \cdot & I_{ii} & I_{ij} & I_{ik} \\ \cdot & \cdot & \cdot & I_{ji} & I_{jj} & I_{jk} \\ \cdot & \cdot & 100 & I_{ki} & I_{kj} & I_{kk} \end{bmatrix}$$

Since fluid properties are disabled, the object faces added mass force – simulated through mass-inertia matrix – and torque about the z-axis and the gravitational force.

Results

⁹ http://en.wikipedia.org/wiki/Backward_Euler_method

As expected the horizontal position along the x and y axes are linearly increasing, which implies a constant velocity. The vertical component is a perfect parabola (relative errors in mean $\langle x, y, z \rangle$: $\langle 0.166\%, 0.166\%, 1.347\% \rangle$) implying a linear velocity, and therefore, a constant acceleration. Similarly, the propeller effect (relative error in mean θ_z : 0.974%) can be seen due to acceleration coupling in the z-axis orientation in Figure 2-8d.

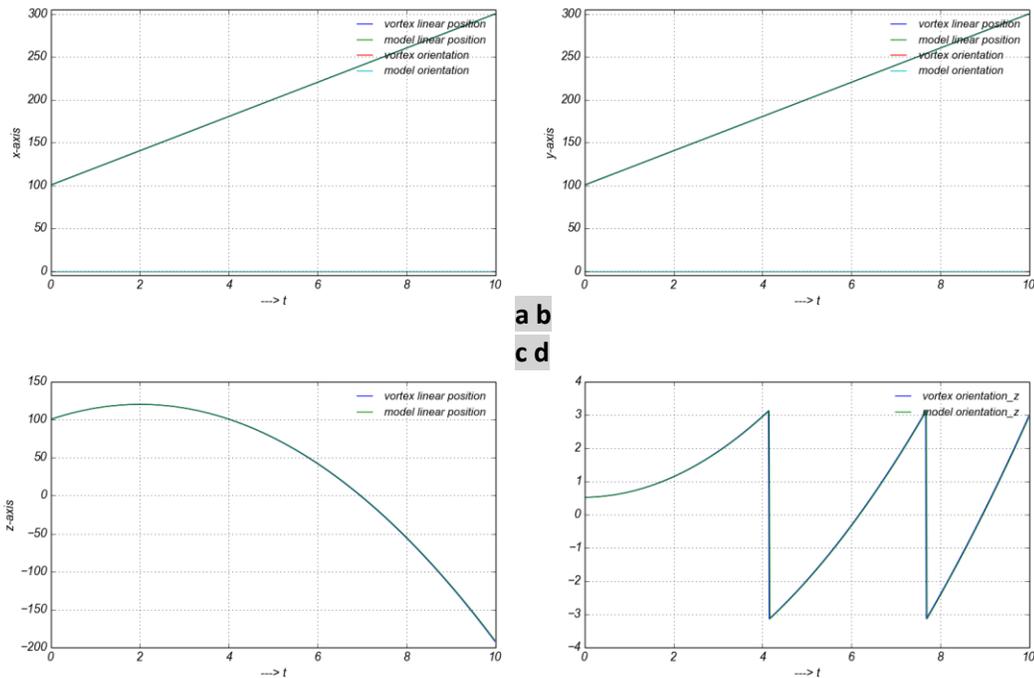


Figure 2-8: added mass and propeller effects on a projectile

2.2.6 Inertia Tensor Translation and Rotation

Test Case

Inertia tensor should be time invariant and should be correctly computed in Vortex Studio for any rotated and/or translated frame.

Model

Like the way mass is the attribute for the resistance to the linear acceleration, the inertia tensor, I , is used for angular acceleration. The inertia tensors, expressed with respect to the Center Of Mass (COM), of different primitive shapes are presented (Table 2-1). Their derivation and additional information can be found in rigid body dynamics literature:

Table 2-1: Inertia tensor for different primitives

Cube:
$$I = \begin{bmatrix} \frac{1}{12}m(h^2 + d^2) & 0 & 0 \\ 0 & \frac{1}{12}m(w^2 + d^2) & 0 \\ 0 & 0 & \frac{1}{12}m(w^2 + h^2) \end{bmatrix}$$

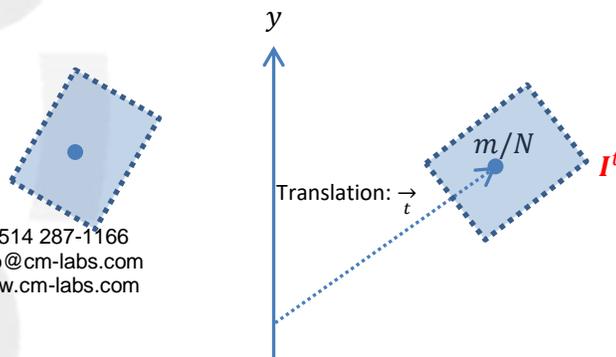
Sphere:
$$I = \begin{bmatrix} \frac{2}{5}mr^2 & 0 & 0 \\ 0 & \frac{2}{5}mr^2 & 0 \\ 0 & 0 & \frac{2}{5}mr^2 \end{bmatrix}$$

Cylinder:
$$I = \begin{bmatrix} \frac{1}{12}m(3r^2 + h^2) & 0 & 0 \\ 0 & \frac{1}{12}m(3r^2 + h^2) & 0 \\ 0 & 0 & \frac{1}{2}mr^2 \end{bmatrix}$$

Hemisphere:
$$I = \begin{bmatrix} \frac{83r^2}{320} & 0 & 0 \\ 0 & \frac{83r^2}{320} & 0 \\ 0 & 0 & \frac{2r^2}{5} \end{bmatrix}$$

where different symbols attribute to the respective meaning:

- m : Mass of the primitive
- h : Height
- w : Width
- d : Breath
- r : Radius



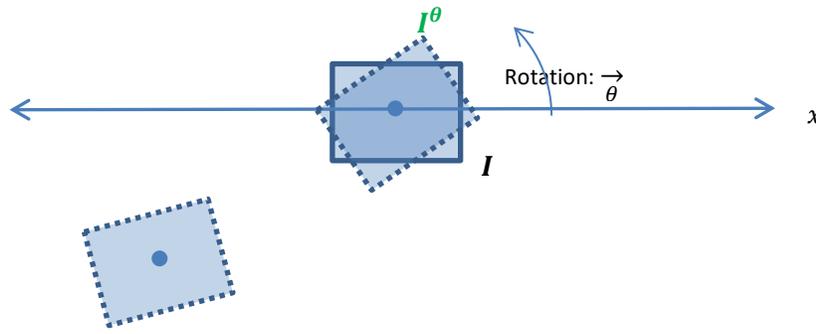


Figure 2-9: Transformation of its inertia tensor due to rotation & translation of a part

Let $\theta = (\alpha, \beta, \gamma)$ and $t = (x, y, z)$ represents rotation (Euler angles: roll, pitch, and yaw) and translation respectively. The corresponding rotation matrix becomes

$$R_{\theta} = R_z(\gamma) \cdot R_y(\beta) \cdot R_x(\alpha) \tag{2.18}$$

Hence, rotated and translated inertia can be computed as

Rotated inertia: $I^{\theta} = R_{\theta}^T I R_{\theta}$ (2.19)

Translated inertia: $I^t = I - md^2$ (2.20)

where d^2 , square of skew-symmetric matrix $d = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$, is derived from translation vector t . After

knowing the necessary translation matrix, inertia tensor of a capsule can be computed as

$$I_{capsule} = I_{cylinder} + I_{upper_hemisphere} + I_{lower_hemisphere} \tag{2.21}$$

$$I_{capsule} = I_{cylinder} + I_{hemisphere}^{(0,0,offset_z)} + I_{hemisphere}^{(0,0,-offset_z)} \tag{2.22}$$

Since capsule inertia is computed considering it as a cylinder having two hemispheres at ends, the offset of $\left(\frac{1}{2} + \frac{3r}{8}\right)$ along cylinder length axis is necessary.

Results

The Figure 2-10 shows the inertia tensor of different parts (each of them has two collision Geometries (CGs) and has undergone a fixed translation and rotation). To display a stream of 3×3 matrix parameter on a 2D figure, each segment with 100 units are allocated to represent I elements in the order of $I_{xx}, I_{xy}, I_{xz} \dots \dots I_{zx}, I_{zy}, I_{zz}$. Notice that Vortex Studio inertia computation fairly closely follows the model. However, for a large translation, corresponding inertia element deviates more from the actual value.

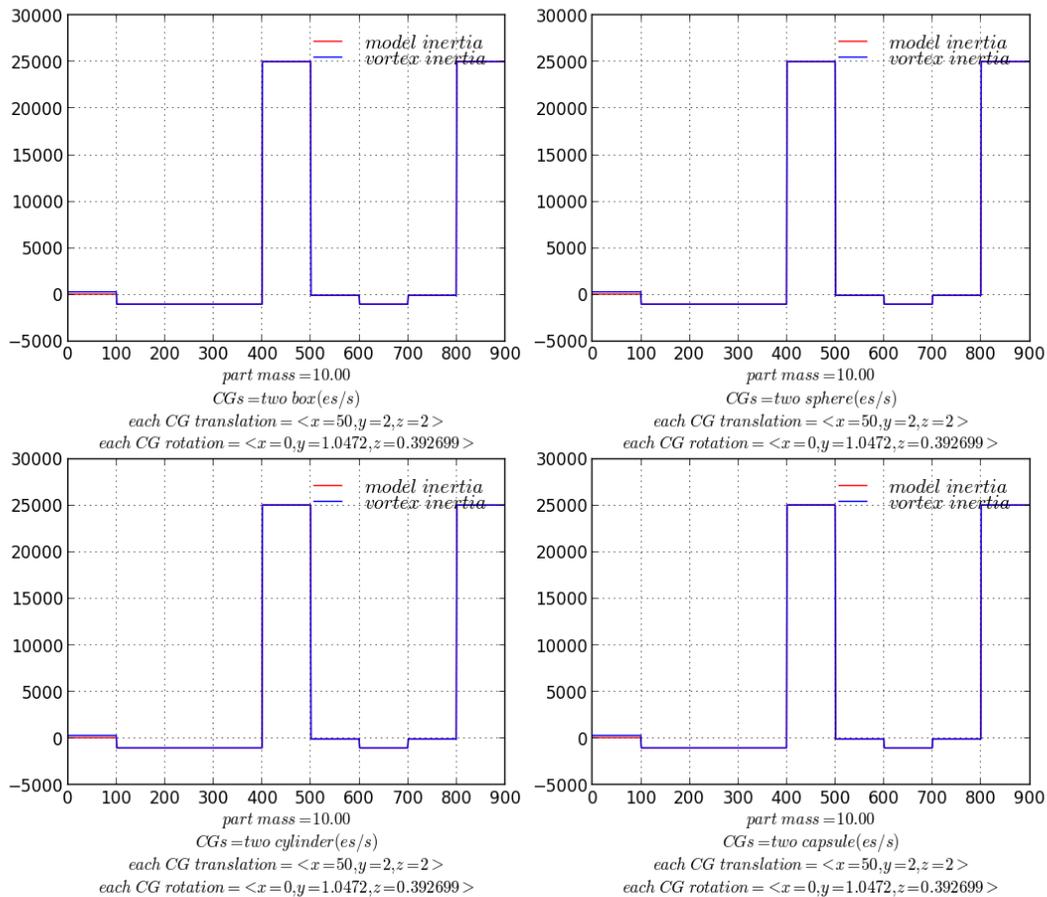


Figure 2-10: Validating inertia tensor property: transformation of part inertia tensor

2.2.7 Friction

Test Case I: Friction Angle

The test should verify that a mass resting at a plane should start sliding if the plane inclination is larger than angle of friction.

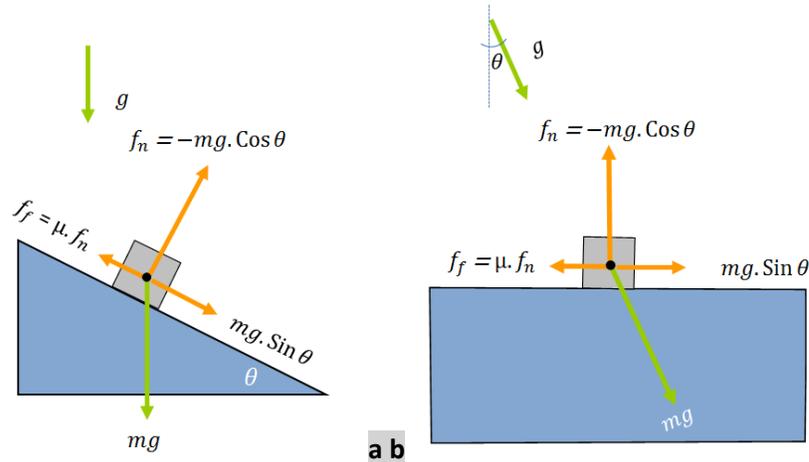


Figure 2-11:

acting on a body in an inclined plane: (a) vertical gravitational force (b). inclined plane simulating by rotating the gravity.

Different forces

Model

From the Coulomb friction model¹⁰, an object in a plane will start sliding when the sliding force is equal or greater than friction force ($f_f = \mu \cdot f_n$). For the above configuration, the sliding force can be altered either by changing the value of gravity, g , or by changing the plane angle, θ .

For the given friction coefficient, μ , there is a unique plane angle associated with the configuration. The object slides for all inclinations beyond what is called the angle of friction, θ_f . Assume that the object starts sliding at θ_f :

$$f_f = \mu \cdot f_n \tag{2.23}$$

$$mg \cdot \sin \theta_f = \mu \cdot mg \cdot \cos \theta_f \tag{2.24}$$

$$\theta_f = \tan^{-1} \mu \tag{2.25}$$

In the verification experiments, it is investigated whether the object stays stationary when the plane inclination is less than θ_f , and that it starts moving if the plane inclination is beyond θ_f .

¹⁰ http://en.wikipedia.org/wiki/Friction#Dry_friction

Results

Figure 2-12 depicts the position of the box (which is at the bottom of the stack of 5 boxes glued together by RPRO constraint) in an inclined plane. When plane inclination equals to the angle of friction (2.25), the box just starts moving, and for all other greater angles, the horizontal displacement is parabolic –a constant acceleration.

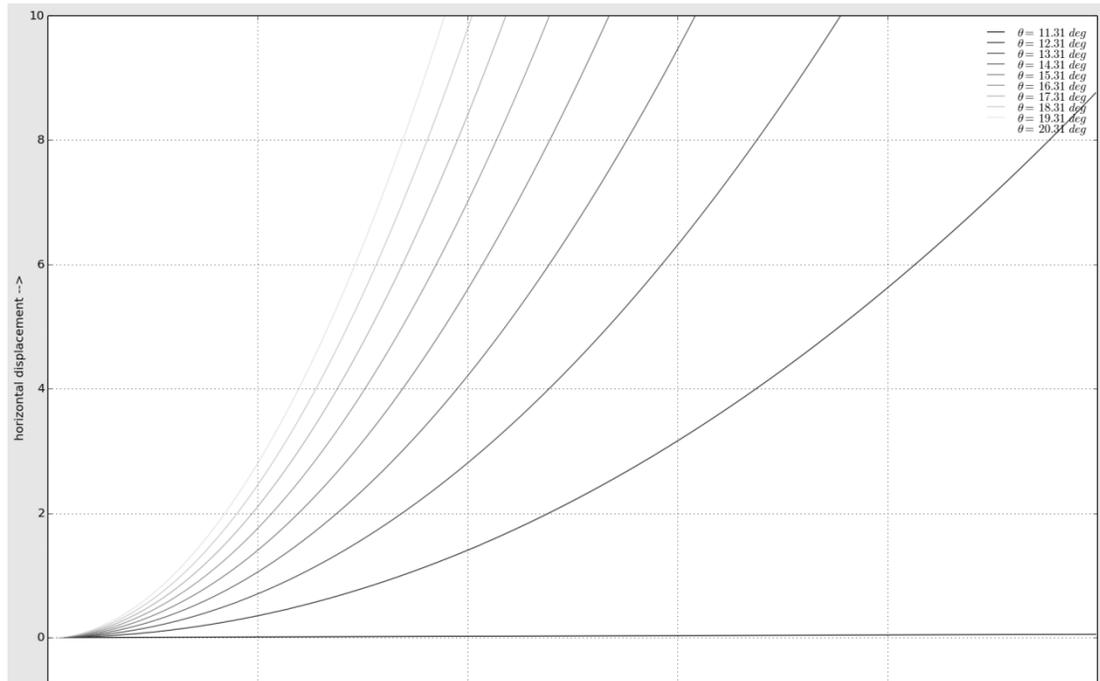


Figure 2-12: Validating static friction property: motion of a box in an inclined plane for different plane angles.

Test Case II: Effect of Friction Coefficients on Trajectory

Since Vortex Studio users can choose distinct orthogonal friction coefficients, this test case is designed to show and verify how their choices affect the trajectory of the object.

To mimic the terrain surface where friction in one direction is different than in its orthogonal direction, Vortex Studio allows users to set the friction in each direction to different coefficients. In the example of a plane surface, where friction force is independent of direction of object motion, friction can be simulated with both coefficients set to same value. In other words, in a given time the object should cover equal distance irrespective of its direction of motion, provided that the initial velocity is the same.

To verify it through experiment, multiple boxes (each with mass 10 kg and not colliding with each other) are allowed to slide along different angles with a constant initial velocity of 10 m/s.

Results

The trajectory plot (Figure 2-13a) shows all boxes cover the same distance irrespective of the direction of their motion. Also, it is verified that they take the same time and follow the same velocity profile to cover that distance (Figure 2-13b). For a cross check, let us estimate the distance covered by a box:

$$s = vt - 0.5at^2 = 10 * 2 - 0.5 * 4.9 * 2^2 = 10.2 \tag{2.26}$$

While the other symbols have their usual meaning, a is the deceleration caused by friction force, $\mu f_n = 4.9 N$. It is very close to the radius of the arbitrary circle $10.0878 m$, after which each box stops moving.

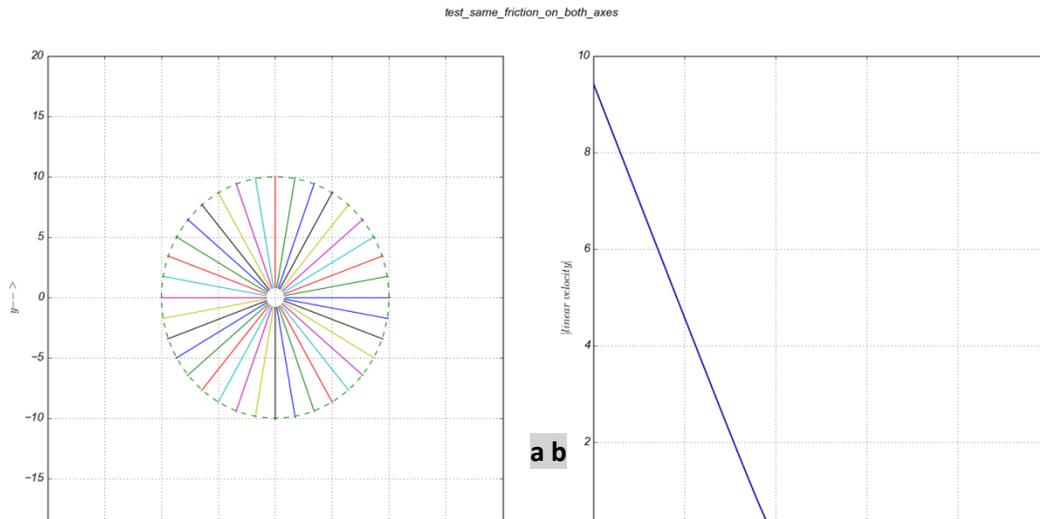


Figure 2-13: Trajectory of the several parts (a) and their velocity profiles (b), with same friction components.

A

small consideration is taken into account by moving COM to the lower face of the box to avoid torque from being developed that retards the object from covering the expected distance.

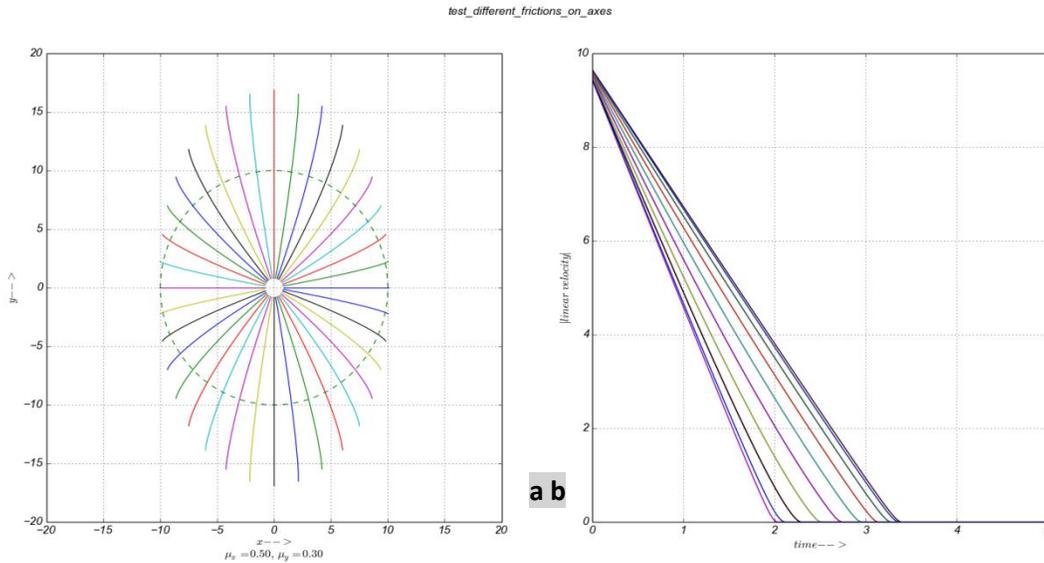


Figure 2-14: Trajectory of the several parts (a) and their velocity profiles (b), with different friction components.

Similarly, to verify that the box does not move in a straight line if different coefficients are chosen, let's alter those coefficients to $\mu_x = 0.5$ and $\mu_y = 0.3$. Since μ_x is higher than μ_y , the box faces lesser friction along y -axis, and it travels a longer distance in that direction (Figure 2-14a). It can also be observed that the box moving along the x -axis covers distance of 10.0878 m, the same distance covered when μ_x and μ_y are both 0.5. This is because μ_y has no effect on the object moving along the x -axis, just as μ_x has no effect on an object moving along the y -axis.

Test Case III: cone-of-friction

Transition from static to dynamic friction should happen at the boundary of cone-of-friction region.

Model

As mentioned before, if friction is limiting ($f_f = \mu f_n$), the resultant force, R , at COM makes an angle of $Tan^{-1}(\mu)$ with f_n . Therefore, for a given μ , there is an angle which limits a region boundary to the reaction force. The boundary is also valid for dynamic friction case. If all directions along the plane surface are considered, the region bounded by the angle $Tan^{-1}(\mu)$ becomes a right circular cone with vertex at the point of contact of the two surfaces; therefore, the friction coefficient of a surface are frequently referred in terms of cone-of-friction.

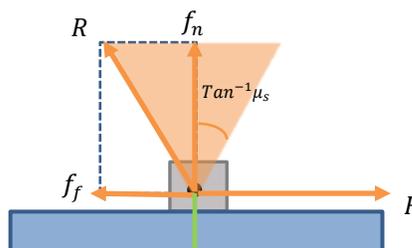


Figure 2-15: cone-of-friction

Results

The experiment is conducted such that an object, box in this case, is dragged on the surface with a constant force and recorded corresponding acceleration profile. The experiment is repeated for different forces, ranging from 45 to 55 N, to drag the part, and the average acceleration of each part is plotted against the force (Figure 2-16a). It allows us to identify the force corresponding to the boundary of cone-of-friction. The transition from static to dynamic friction condition is noticed when the pulling force is around 49 N which is exactly as expected: the object just starts moving when the external pull overcomes friction force which is $\mu \cdot mg = 0.5 \cdot 10 \cdot 9.80 = 49\text{ N}$. Note that because the terrain has finite stiffness, the object sinks little bit at the beginning (Figure 2-16b).

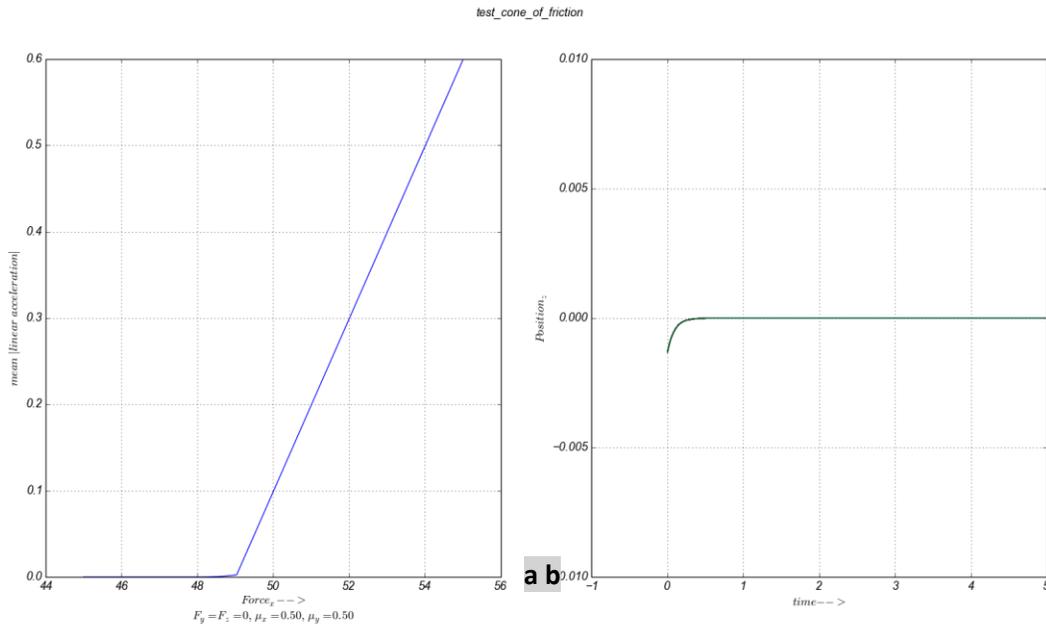


Figure 2-16: average acceleration profile showing transition from static to dynamic friction condition

2.2.8 Contacts

Test Case

To have a realistic and smooth simulation of a primitive rolling over a surface, the contact force should be continuously available, have less jitter, and be computed in a reasonable time so that the target FPS will be met at all times.

The test should verify that contact force is smooth and its computation (detection and generation) is efficient.

Scenario

The configuration consists of several primitives (convex primitives such as convex-box, convex-cylinder, convex-capsule and convex-sphere) rolling around a central box which is pivoted at a fixed point with the motorized universal constraint, and the primitives with Linear2 Angular3 constraint (angular coordinates of which are motorized and linear coordinates are locked). This way, the primitives will be rolling over the central box, facing smooth as well as edge surfaces. Note that no primitives are allowed to collide with each other, but with the central box.

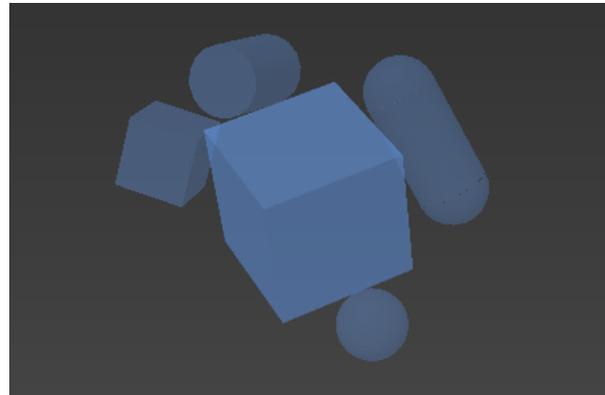
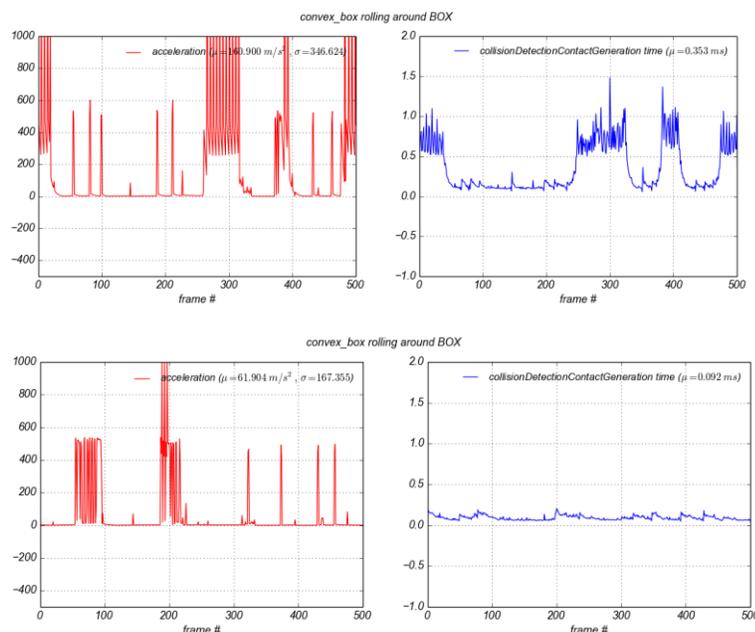


Figure 2-17: Scenario: different convex primitives (box, cylinder, capsule, sphere) rotating around a static Box.

Results

In Vortex Studio revision 6.6.0, a few technological improvements were made in contact force computation. The simplification is made mainly for convex primitives using volumetric contact models¹¹. As a result, smooth and continuous contact force is available.



(a b)

¹¹ Contact Dynamics Modelling for Robotic Task Simulation – Yves Gonthier

(c d)

Figure 2-18: Acceleration and contact computation time for the convex-box primitive: (a b) in Vortex Dynamics 6.5.0 and (c d) in Vortex Dynamics 6.6.0

Acceleration and contact solving time (which includes collision detection and contact generation time) profiles of a convex-box are logged and presented. In 6.5.0 (Figure 2-18 a), mean and standard deviation of acceleration are $\mu, \sigma = 160.9 m/s^2, 346.624$ respectively, which are reduced to $\mu, \sigma = 61.904m/s^2, 167.355$ in 6.6.0 (Figure 2-18 c). Similarly, there is a noticeable average performance gain of 73.94% in contact solving time (Figure 2-18 bd). Lower values of both attributes not only imply that the object is moving smoothly but also indicate that there is less jitter.

The experiment is repeated for other convex primitives. The results are equally encouraging, and they are tabulated below (Table 2-2).

Table 2-2: performance improvement indicators

Acceleration	$\mu, \sigma @ 6.5.0$	160.9, 346.624	112.0, 277.273	41.044, 134.78	228.387, 312.585
	$\mu, \sigma @ 6.6.0$	61.90, 167.355	24.167, 98.587	25.412, 106.517	18.325, 81.488
Contact solving time	$\mu @ 6.5.0$	0.354	0.369	0.372	0.356
	$\mu @ 6.6.0$	0.093	0.105	0.096	0.096

2.3 Fluid Interaction

Object interacting with fluid forces is one of the important behaviours that CM Labs strives to simulate right. Some of the fundamental parameters and properties that govern the fluid interaction are added mass, drag, and lift. Their effects are being tested in the following sections.

2.3.1 Added Mass Property

Test Case

The test should verify that an added-mass property contributes to the dynamic behaviour of the object. An object, with mass m , inside a fluid with a mass m_a , is placed in an environment with zero gravity and no damping. If the object is constrained with the *VxSpring* (stiffness k and damping c) and is violated by a small amount to initialize it, the object will oscillate with the (undamped) frequency given by

$$\sqrt{\frac{k}{m + m_a}} \quad (2.27)$$

It attains the natural frequency of

$$\omega_0 = \sqrt{\frac{k}{m}} \quad (2.28)$$

when there is no added mass.

Model

An object hung using *VxSpring* constraint should simulate a mass-spring-damping (MSD) system.

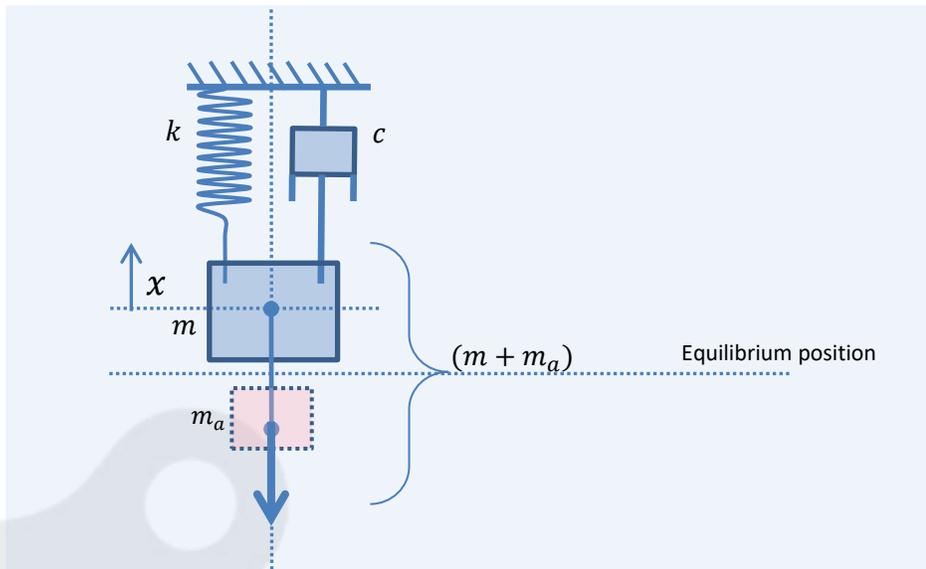


Figure 2-19: A naturally oscillating MSD system immersed into the fluid

In the absence of gravity, the MSD system is governed by homogeneous 2nd order differential equation¹².

$$(m + m_a) \frac{dx^2}{dt^2} + c \frac{dx}{dt} + kx = 0 \tag{2.29}$$

The damping ratio that describes how the oscillation decays in the system is given by

$$\zeta = \frac{c}{2\sqrt{km_{eq}}} \tag{2.30}$$

If the damping ratio is suppressed to zero, the system oscillates with (undamped) natural frequency

$$\sqrt{\frac{k}{m + m_a}} \tag{2.31}$$

Results

Figure 2-20 shows that the Vortex Studio response very closely follows the model for all masses, except for small inherited damping.

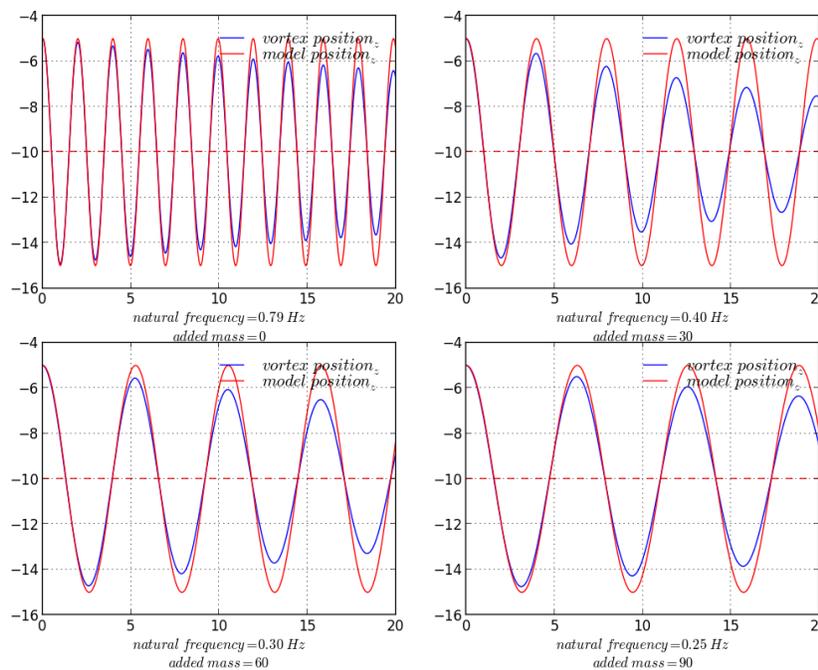


Figure 2-20: Validating added mass property: changing the behaviour of MSD system by controlling the mass

Assumptions/notes

- Fluid viscosity is not accounted for.
- Semi-implicit solver in Vortex Studio brings stability. However, this inherits the extra damping which can be improved at the cost of higher step size.

¹² <http://en.wikipedia.org/wiki/Damping>

- It is noticed that buoyancy force equals gravity force when the added mass is 1 (not when it is 0).

2.3.2 Restoring Torque due to Buoyancy

Test Case

If a half-submerged and rotated prismatic object is released, it will restore to a stable position. The restoring torque is generated by the effect of buoyancy and gravity forces.

Model

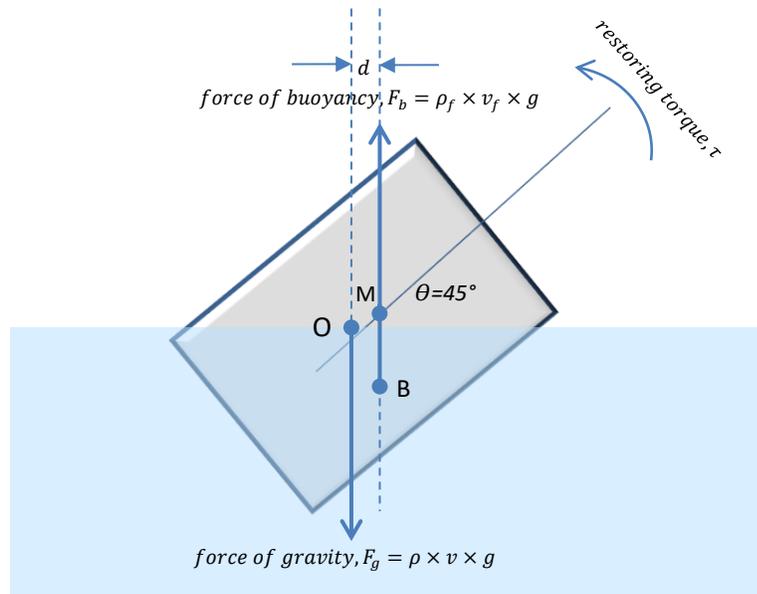


Figure 2-21: A half-submerged but rotating prism and different forces acting on it

- θ : Angular position
- O : Center of gravity
- M : Metacenter
- $v = \ell \times b \times h$: Prism volume
- ρ : Prism density
- B : Centroid of submerged volume, buoyancy center
- v_f : Displaced volume
- ρ_f : Fluid density
- g : Gravity

When there is an axial drift between gravity and buoyancy forces, restoring torque is produced. The amount of the torque is given by

$$\tau = d \times F_b \quad (2.32)$$

The object’s stability is defined where the metacenter lies above or below the CG. The metacenter is the point where F_b meets the CG axis.

Results

The Vortex Studio and the model responses are exactly overlapping each other is clear evidence of the fact that the Vortex Studio response follows the model exactly.

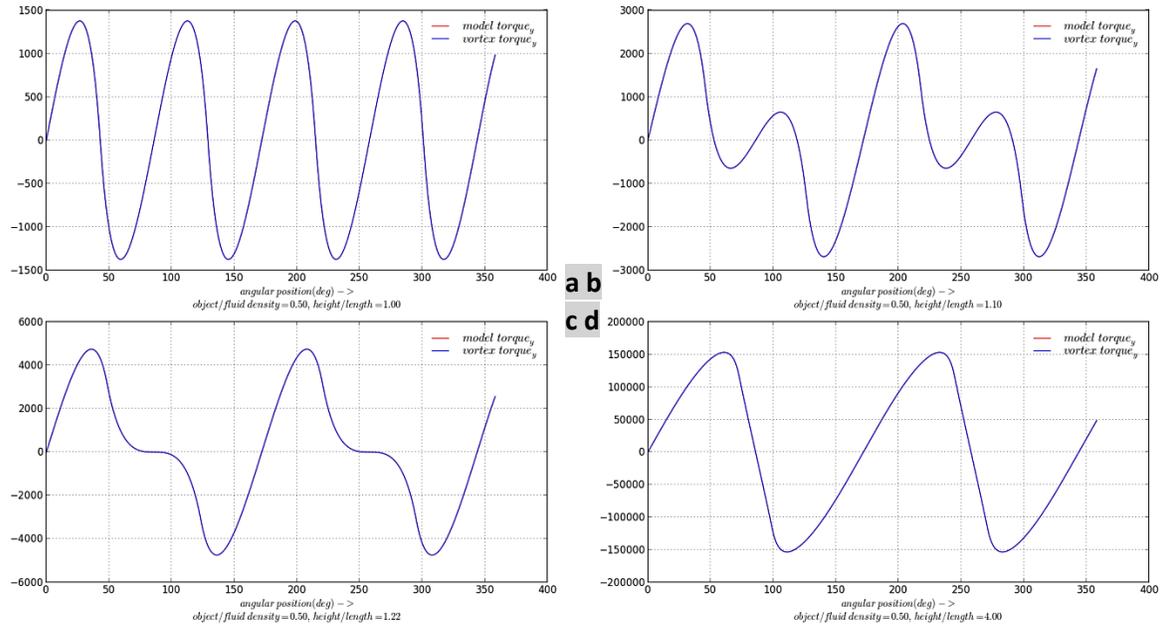


Figure 2-22: Validating buoyancy property: restoring torque on the half-submerged prism at different rotation positions

- a) *height/length=1.0*
- b) *height/length=1.1*
- c) *height/length=1.22*
- d) *height/length=4.0*

Assumptions/notes

For better accuracy, the box is modeled using triangle mesh. The higher the mesh resolution, the more accurate the result; however, two triangles per face was chosen to achieve this level of accuracy.

2.3.3 Drag and Lift

Test Case

Fluid interaction is one of the key properties that must be computed accurately to simulate equipment such as ROVs and vessels.

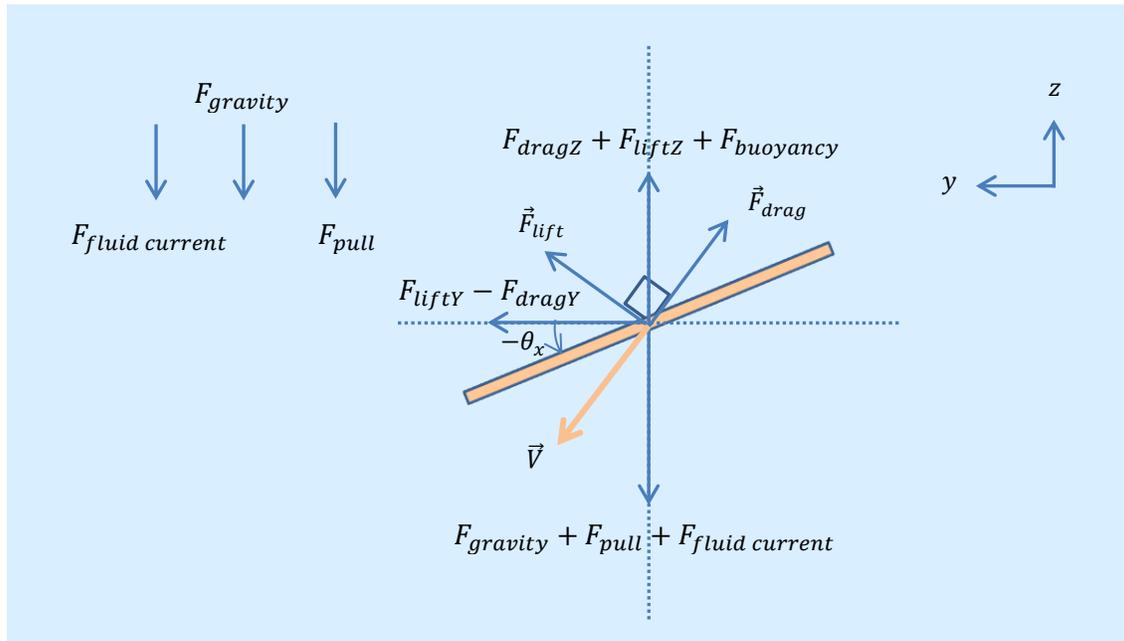


Figure 2-23: Experimental setup for Drag and Lift properties verification: a fully submerged thin plate and different forces acting on it

In Figure 2-23, a flat, thin square plate, is rotated by $-\theta_x$ degree and is fully immersed into a fluid. As a result, the plate is subject to different fluid forces, including forces due to fluid flow and external pull (both along same axis i.e. z-axis). Drag force, denoted by \vec{F}_{drag} , will be observed opposite to the velocity direction, and lift force, denoted by \vec{F}_{lift} , will appear at the orthogonal axis.

The net force causes the acceleration on the body:

$$m \frac{d^2 P}{dt^2} = \sum_{i=0}^n F_i \tag{2.33}$$

where $P = \langle x, y, z \rangle$ is the position triplet. The equation (2.33) can be resolved into its components.

$$m \begin{bmatrix} \frac{d^2x}{dt^2} \\ \frac{d^2y}{dt^2} \\ \frac{d^2z}{dt^2} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n F_{ix} \\ \sum_{i=0}^n F_{iy} \\ \sum_{i=0}^n F_{iz} \end{bmatrix} \quad (2.34)$$

Furthermore,

$$\begin{bmatrix} \frac{d^2x}{dt^2} \\ \frac{d^2y}{dt^2} \\ \frac{d^2z}{dt^2} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} 0 \\ -F_{dragy} + F_{lifty} \\ F_{dragz} + F_{liftz} + F_{buoyancy} - F_{pull} - F_{gravity} - F_{fluid\ current} \end{bmatrix} \quad (2.35)$$

Where

$$F_{drag} = 0.5\rho_{fluid}A_v C_d(V)^2 \quad V - \text{Velocity of the object}^{13}$$

A_v – Area projected along object displacement
 C_d – Drag coefficient
 ρ_{fluid} – Fluid density

$$F_{lift} = 0.5\rho_{fluid}AC_d(V)^2$$

$F_{buoyancy} = \rho_{fluid}Vg$ V - Volume of the object
 $F_{gravity} = mg$ m, g - Mass and gravity

Experiment and Methodology

As shown in Figure 2-23, a thin aluminum plate with dimension $2 \times 2 \times 0.01 \text{ m}^3$ is fully immersed into the fluid with density of 1000 kg/m^3 . Other related parameters are as follows:

Plate density: 2740 kg/m^3 (~ aluminum)
 Rotation about x-axis: $-\pi/10 \text{ rad}$
 Fluid laminar flow along z-axis: -0.5 m/s
 Drag coefficient: 0.41
 External pull along z-axis: -200 m/s

The plate is denser than the same amount of fluid. The downward force is greater than the upward force. Therefore, the plate is assumed to move in third quadrant. The motion would create a drag force in the

¹³ http://en.wikipedia.org/wiki/Drag_coefficient

opposite direction and lift in the orthogonal direction, as indicated by \vec{F}_{lift} and \vec{F}_{drag} respectively. The resulting object motion in space caused by cumulative forces is described by Equation (2.35). These are set of second order nonlinear nonhomogeneous ordinary differential equations – nonlinear because drag and lift forces are function of velocities squared $(dP/dt)^2$. To solve these equations, an ODE solver from open-source Scipy¹⁴ Python package is used, with initial position and velocity being $P_{t=0}$ and $V_{t=0} = \langle 0, 0, 0 \rangle$ respectively.

The position and velocity of the plate are logged throughout the simulation. Similar data are populated by running the model at given time instances. Finally they are compared against each other.

Results

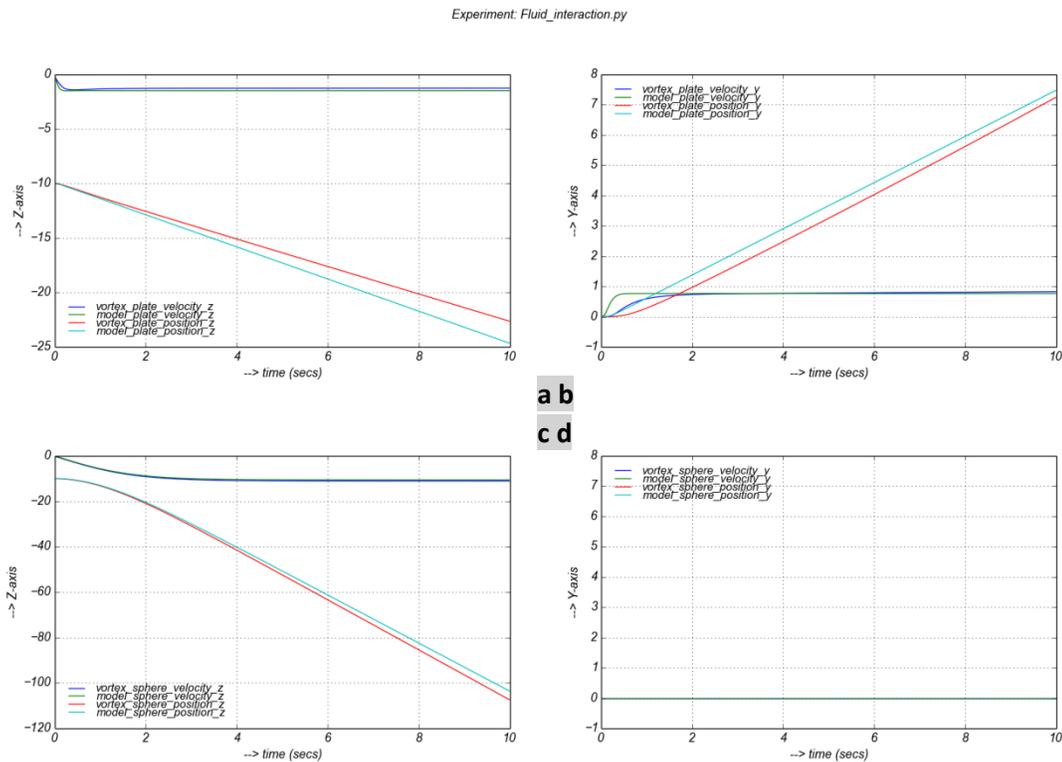


Figure 2-24: Object trajectory along drag (z-axis) and lift (y-axis) direction.

The result shows that both vertical and horizontal components of the trajectory follow the model:

Relative error in mean in position_Y = 0.106462687752
 Relative error in mean in velocity_Y = 0.0326385883212
 Relative error in mean in position_Z = 0.0580228956313
 Relative error in mean in velocity_Z = 0.160687231762

The object attains a terminal velocity in both directions, which is an expected behaviour: when opposite forces along the motion are equalized, a body moves with zero acceleration (Figure 2-24ab). To take the experiment one step further, a sphere with 2m diameter is also subjected to the same environment. It is noted that the

¹⁴ <http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.integrate.odeint.html>

sphere drops $\sim 100m$ into the fluid while plate has just covered $10m$ at the same time (Figure 2-24c). This verifies that the sphere faces less drag force as compared to the plate. The sphere also does not react to the lift force (Figure 2-24d) because it is assumed that omni-symmetric objects, such as a sphere, are negligibly affected by lift and therefore ignored.

Notes

- Drag and lift effects due to the thickness ($0.01m$) are ignored.
- To simplify the model, it only takes projected areas into account when computing drag and lift, which is insufficient for omni-symmetric objects. Therefore, projection area along y-axis is explicitly set to zero to address the fact that sphere has no lift.
- For simplifying the model, a constant drag coefficient is chosen although it is a property that varies with several factors such as fluid properties, its velocity and direction, and object geometry.

2.4 Dynamics Modules

Vortex Studio provides modules such as Cable Systems, Vehicle Systems and Earthwork Systems to offer high-level custom interfaces to the API. This allows greater flexibility and rapid application creation opportunities for our customers.

2.4.1 Cable Systems: Catenary

Test Case

Cable, having its own mass and freely hanging at two fixed points, should follow a catenary shape.

Model

The catenary shape of a flexible and uniform weight cable is a function of different parameters:

- w : Uniform load (or weight per unit length), N/m
- T_H : Horizontal component of the tension, N
- S : Span length, m

If the ‘maximum sag point, D, is considered as origin, then the catenary shape is modeled using a hyperbolic function¹⁵:

$$y(x) = \frac{T_H}{w} \left[\cosh\left(\frac{x}{T_H/w}\right) - 1 \right] \tag{2.36}$$

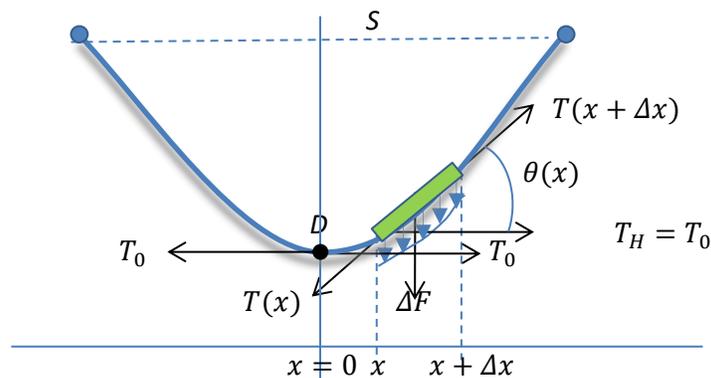


Figure 2-25: A flexible cable taking Catenary shape under its own weight

Tension profile is also an interesting parameter to look upon. An incremental length Δx is subjected under forces $\Delta F (= \Delta x \cdot w \cdot g)$, $T(x)$ and $T(x + \Delta x)$. At equilibrium, summation of forces along vertical as well as horizontal direction should be equal zero:

$$-T(x)\cos\theta(x) + T(x + \Delta x)\cos\theta(x + \Delta x) = 0 \tag{2.37}$$

$$-T(x)\sin\theta(x) + T(x + \Delta x)\sin\theta(x + \Delta x) - \Delta F = 0 \tag{2.38}$$

The (2.38) follows that horizontal component of tension is always a constant, equal to maximum tension.

¹⁵ <http://en.wikipedia.org/wiki/Catenary>

$$T(x)\cos\theta(x) = \text{Constant} \Rightarrow T(x)|_{x=0} = T_H \tag{2.39}$$

So, the tension profile throughout the cable is given by

$$T(x) = \frac{T_H}{\cos\theta(x)} \tag{2.40}$$

Results

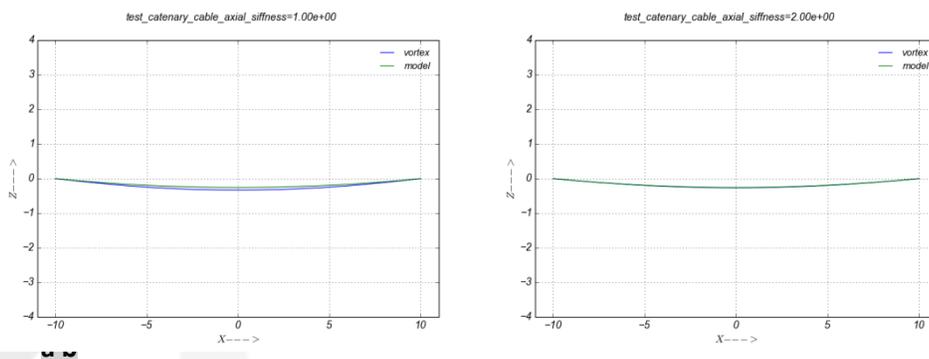
A 40 mm diameter *Surelift35* hoist cable from Wire Rope¹⁶ having linear density of 7.95 kg/m and maximum breaking load of 1521 kN is selected as a simulation candidate. A quick look at the simulation configuration is presented below:

```

simulation_time = 10.0           # secs
pivot_separation = 20.0
cable_radius = 0.02             # m
preferred_section_length = 0.1
max_number_of_sections = 2000
max_elongation_percentage = 50.0
load_for_relative_elongation = 1559.2335 # N
breaking_load = 1521            # kN
relative_elongation = 5e-3      # m
youngs_modulus_Gpa = 200        # GPa
number_of_wires = 1
cable_linear_density = 7.95     # kg/m
    
```

Case I: as a first test, a catenary (the shape of a curved cable takes under its own weight when only supported by its end pivots) cable is created, and the resulting immersive behaviours are analyzed for verification. The ends are separated by 20 m.

The objective is to verify whether it follows a symmetric catenary shape (2.36). The result (Figure 2-26) shows that the cable system strictly follows the model profile, keeping the symmetry intact, but with a negligible deviation in segment positions: mean relative error of 0.2800 and sag absolute error of 0.0761 m.



¹⁶ <http://www.wirerope.com/docs/ctg-cranes.pdf>

Figure 2-26: Validating cable system: symmetric catenary shape

Since the cable is simulated as a chain of lumped mass elements, a small deviation from the actual model is expected. However, it can be tuned through input parameters such as axial stiffness scale, relative elongation and load. For instance, as the axial stiffness scale is changed to 2.0 from 1.0 (default value), Vortex Studio gets closer to the catenary model (Figure 2-26): relative error in mean and absolute error in sag are reduced to 0.02 and 0.0081, respectively.

The horizontal tension becomes lowest at the pivots and gradually increases due to self-weight, and reaches its maximum at the bottom ($x = 0$) (refer to Figure 2-25). The maximum tension logged is 14240.79 N , while the supplied value at the configuration time is 15210.0 N , which is only a small relative error of 6.8%.

Updated in Vortex Dynamics 6.6.0:

One of the immersive behaviours of a simulated cable is its symmetric shape in a catenary setup. In this release, an improvement is made for a better symmetric shape. It helps us to feel confident that the axial and bending parameters are properly interacting with each other during the simulation. How is it different than the experiment just before? Well, the previous test case was designed to determine how overall shape is off from the actual model; this time, we wanted to see how symmetric the cable is about the vertical axis.

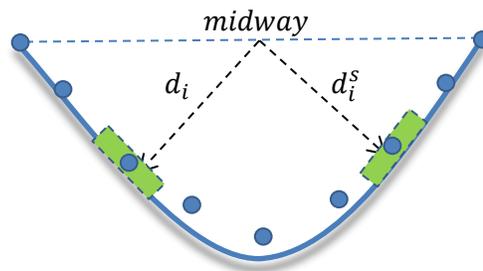


Figure 2-27: Sketch showing how symmetrical cable shape is quantified

The absolute error in distance from midway to a section, d_i , and its symmetric counterpart, d_i^s , is accumulated over all the sections (where $2n$ is number of sections) shown by dots (Figure 2-27), and it is considered as error indicator for symmetric shape - $\sum_{i=0}^n |d_i - d_i^s|$.

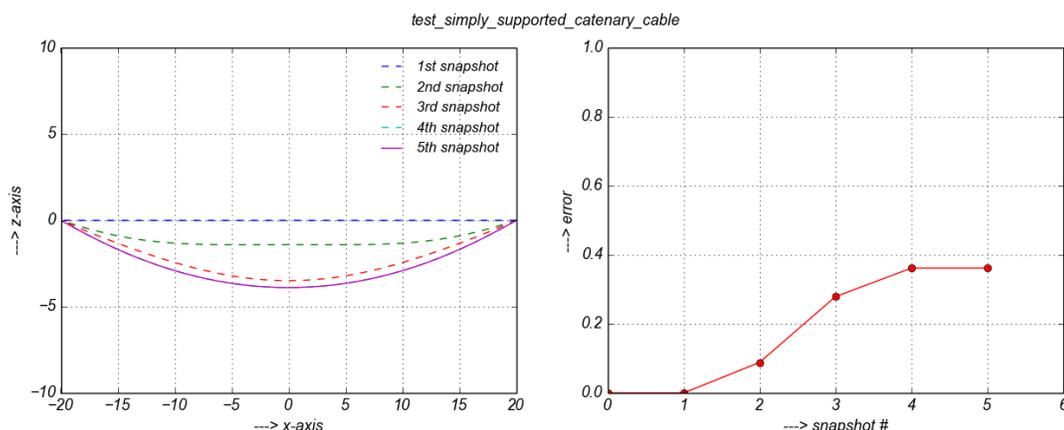


Figure 2-28: Snapshots during cable forming a catenary shape (a) and the error profile (b)

It is found that there is almost no error at the beginning and slowly increases as the cable stretches (Figure 2-28). At settling (5th snapshot), it is around 0.36. This small error is still present because bending points between segments shift due to finite stiffness (underneath it is a change in limits' position of the prismatic constraint). The fact that a stiffer cable deviates less can be shown by simulating a cable hinged at the middle. This improvement makes cable simulation a more realistic: absolute error in sag is brought down to 0.0076 from 0.0081 (see results from previous experiment).

Case II: It is also interesting to see when one of the pivots is lowered, the lowest point in the catenary - where the horizontal tension is at its maximum - shifts towards lowered pivot.

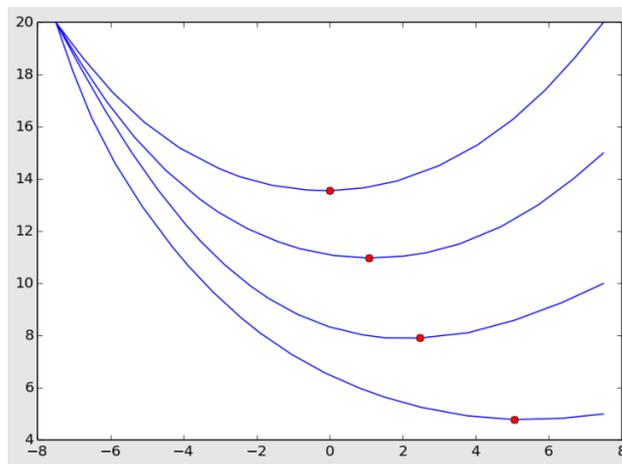


Figure 2-29: Validating cable system: there is a gradual shift in maximum tension point when right pivot point is lowered

We can visualize (Figure 2-29) that the lowest point (marked by red dots) shifts towards the right when the right pivot is lowered at discrete levels.

Assumptions/Notes

Since the cable system doesn't usually take use-case specifications as parameters, it is obvious to see the emerging behaviour of the API being slightly off from the model. Yet, fine tuning the default API parameters could result in a cable system with a more realistic response.

2.4.2 Cable Systems: Pendulum

Test Case

A pendulum made with a thin cable should behave like a simple pendulum.

Model

Besides making sure that the behaviour is realistic, the experiment is also to see if the pendulum time period is comparable to that of a simple pendulum: an ideal pendulum with zero cable and point mass.

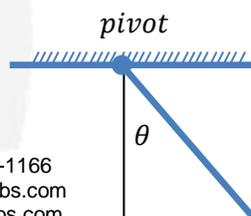


Figure 2-30: An illustration of a pendulum and its component forces

A simple pendulum can be treated as a special case of a physical pendulum with moment of inertia (MOI) $I = I_0 + ml^2$, where I , m and l are pivot referred to MOI, mass and cable length respectively.

The angular acceleration of the sphere is due to gravity generated torque:

$$I \frac{d^2\theta}{dt^2} = -mgl \sin\theta \quad (2.41)$$

$$I \frac{d^2\theta}{dt^2} + mgl \sin\theta = 0 \quad (2.42)$$

This is a non-linear second order homogeneous equation and its natural frequency is given as:

$$\omega_0 = \sqrt{\frac{mgl}{I}} \quad (2.43)$$

and the Time period as:

$$T = \frac{2\pi}{\omega_0} = 2\pi \sqrt{\frac{I}{mgl}} \quad (2.44)$$

Check:

If we consider that the sphere is very 'small' in the sense that $I_0 \ll ml^2$ (2.44) time period of a point mass pendulum is given as

$$T = 2\pi \sqrt{\frac{l}{g}} \tag{2.45}$$

Results

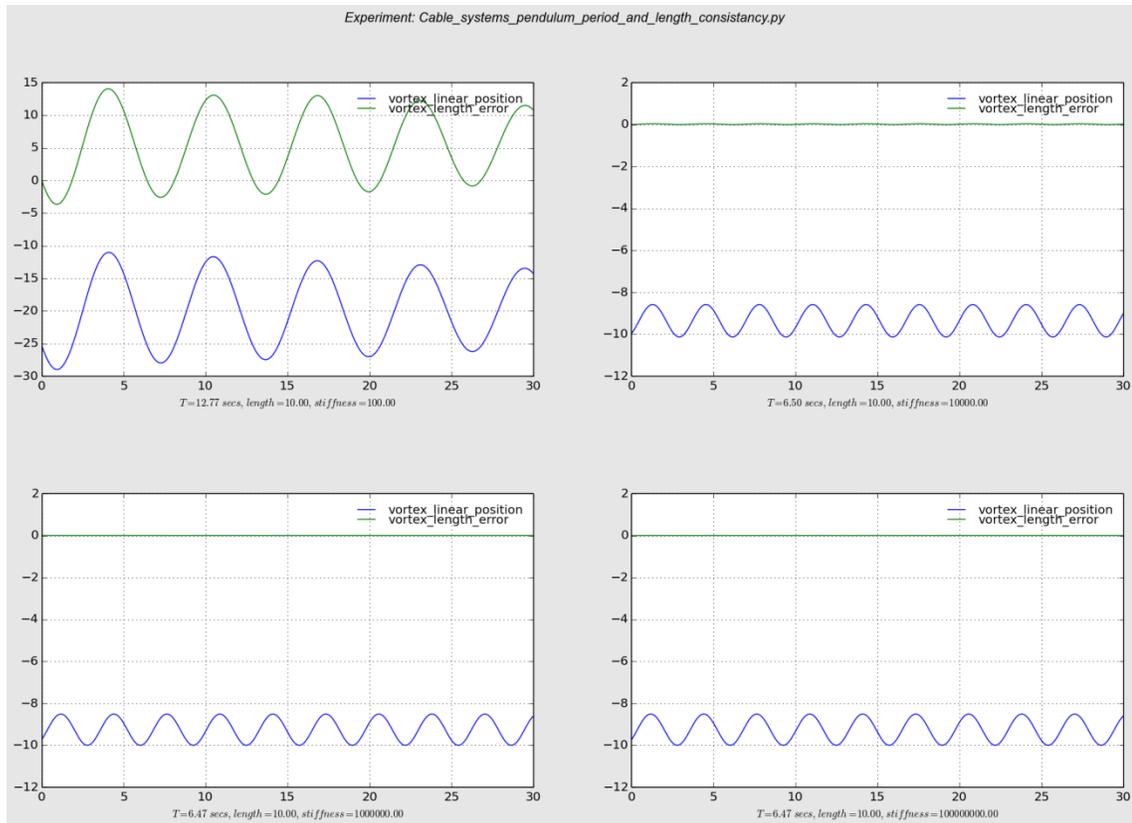


Figure 2-31: Pendulum position along z-axis and length error

For the given length, the oscillation time period of 6.47 secs (highest stiffness case) is reasonably closer to model period $T = 2\pi \sqrt{\frac{10}{9.81}} \approx 6.34 \text{ secs}$, resulting in a relative error of 2.05%. A change in the pendulum length Δl during simulation is what we call length error.

- As we move from the top to bottom row in Figure 2-31, the cable stiffness is increased, length error is reduced, and pendulum time period is decreased. Maximum length error occurs when the pendulum bob is at its lowest position.
- It is obvious that there will be less vibration among segments when stiffness is increased and the cable section acts like a single rod. This would allow the pendulum to complete its cycle in less time.
- Pendulum load at end has direct influence on pendulum length. The length rises linearly to mass, as shown in Figure 2.33, since the stiffness of the cable is finite. However, the length error does not depend on the initial cable length for a stationary pendulum.

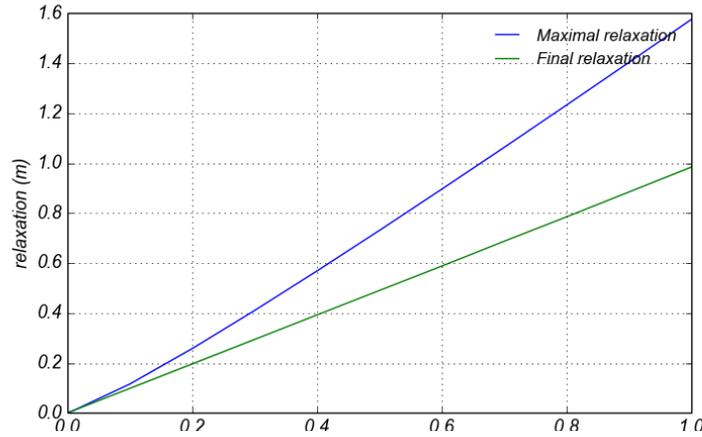


Figure 2-32: Length error relative to the mass for a stationary pendulum

Assumptions/notes

- Cable radius is considered to be negligible.
- The model formulation is valid for point mass. However, in practice, it is not realizable.

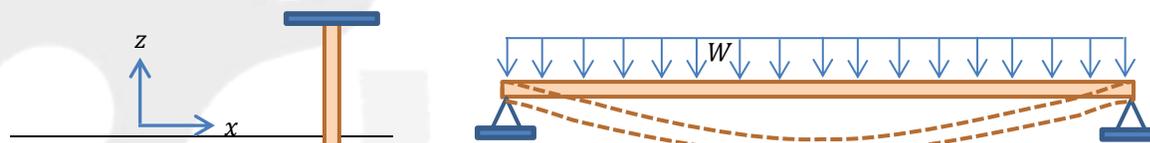
2.4.3 Cable Systems: Elongation and Bending

Test Cases

A cable stretching and bending under loading should mimic the beam bending phenomenon in a miniature scale.

Model

A cable in Vortex Studio is simulated by connecting uniformly spaced parts through constraints, so the self-load of the cable is contiguously distributed throughout the length. In order to verify that the behaviour of the cable bending is physically correct, we have chosen to use the Euler-Bernoulli beam theory¹⁷ as the comparison model. We will restrict our analysis in the cases where the Young’s modulus is high enough to be in a range where the beam theory is valid.



¹⁷ E. A. Witmer (1991–1992). "Elementary Bernoulli-Euler Beam Theory". MIT Unified Engineering Course Notes. pp. 5–114 to 5–164.

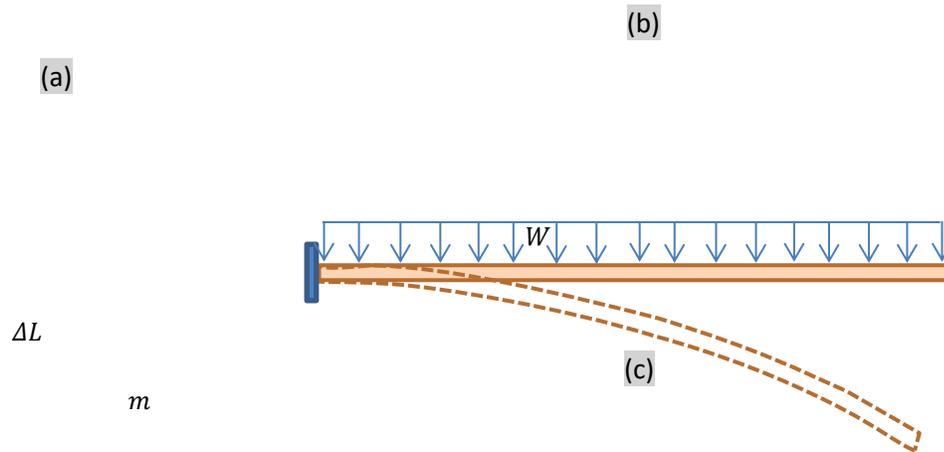


Figure 2-33: Cable loading cases – stretching (a), simply supported (b), and cantilever (c).

Three test cases with different loading conditions are designed. Stretching cable due to a mass hanging on its open end is the first case; simply supported cable, subjected under self-weight, is the second case; and for the third case, cantilever cable under its self-weight is analyzed. Analytical models for such loading conditions mentioned in Figure 2-33 can easily be found in any structural analysis textbook, and are stated below:

Stretching (Mass Hanged at One End):

- i. Elongation

$$\text{Elongation: } \Delta L = \Delta L_R * L \tag{2.46}$$

Note relative elongation, ΔL_R , for the given Load, mg , are input parameters to the Cable Systems' *Dynamic Cable Property* which internally sets the axial stiffness of the constraints connecting internal lump elements. The objective of this experiment is to make sure that same elongation is achieved when loaded with mg .

Simply Supported (Both Ends Fixed):

- ii. Uniform self-load

Deflection:

$$z(x) = -\frac{wx(L^3 - 2Lx^2 + x^3)}{24EI} \tag{2.47}$$

Max deflection value:
$$z(x)|_{L/2} = \frac{5wL^4}{384EI} \quad (2.48)$$

Cantilever:

- i. Uniform self-load

Deflection:
$$z(x) = -\frac{wx^2(6L^2 - 4Lx + x^2)}{24EI} \quad (2.49)$$

Max deflection value:
$$z(x)|_{L/2} = \frac{wL^4}{8EI} \quad (2.50)$$

The symbols used in analytical models have the following meanings:

- w : Uniform weight per unit length
- L : Cable initial length
- ΔL : Cable elongation due to hanging mass m
- E : Young's modulus
- I : Area moment of inertia about the cable lateral axis

Results

As shown earlier (Figure 2-33), there are three test cases. In the first case, a mass is attached at the free end of a dangling cable. A cantilever cable is bending under its own weight in the second case. Finally, in the third case, a cable is supported at two ends and subjected under its self-weight. Some of the parameters common to both of the experiments are as follows:

```
simulation_time = 10.0           # secs
cable_length = 10.0
cable_radius = 0.02             # m
preferred_section_length = 0.1
max_number_of_sections = 1000
max_elongation_percentage = 50.0
relative_elongation = 5e-3
hanging_load = 9801.77*9.80     # N
youngs_modulus_Gpa = 200        # GPa
number_of_wires = 1
volumetric_density = 7800       # iron: 7800 kg/m^3
cable_linear_density = volumetric_density*(pi*cable_radius^2) # kg/m
```

For each test case, the simulation runs for a given time, and the final shape of the cable, along with the corresponding model defined by equations (2.47) to (2.50), are displayed (Figure 2-34-Figure 2-36).

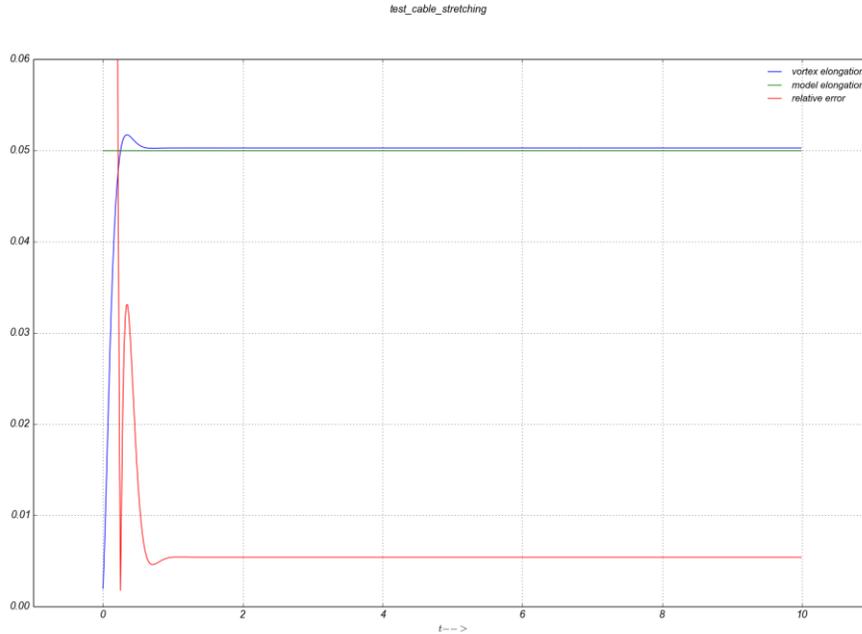


Figure 2-34: Simulation of a stretching cable.

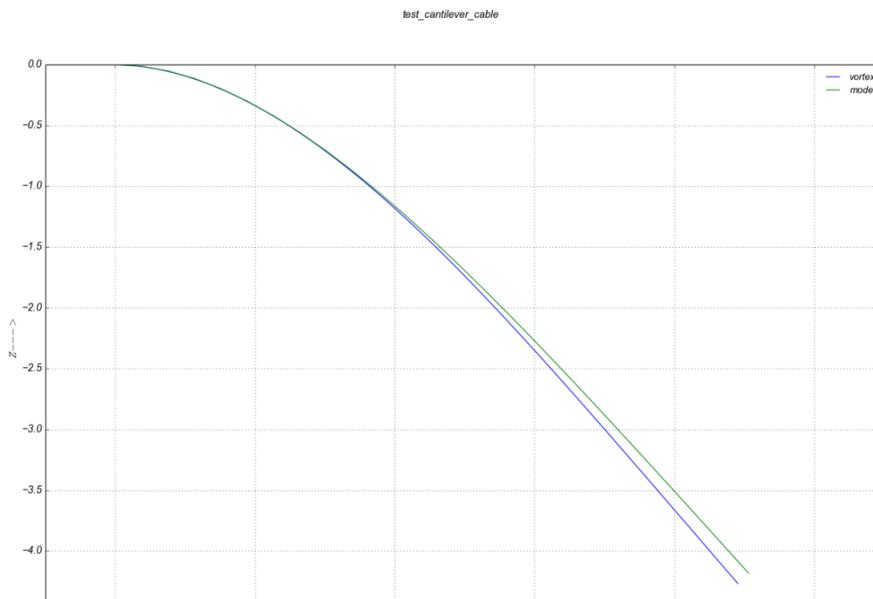


Figure 2-35: Simulation results of a cantilever cable.

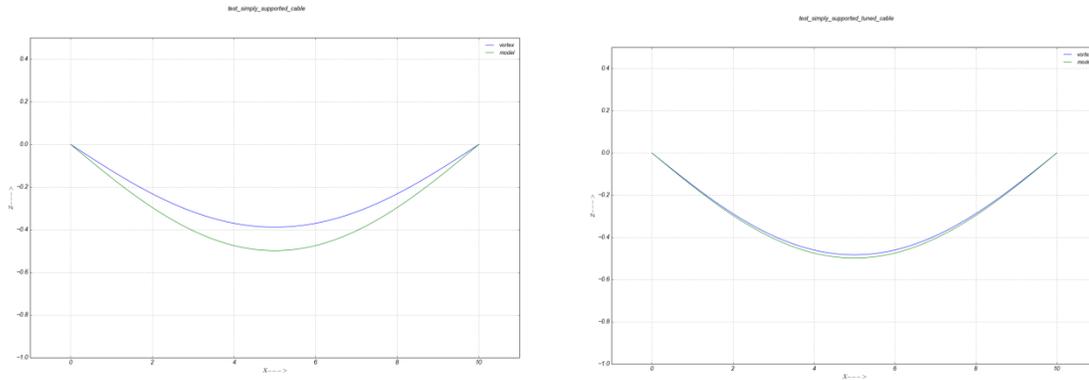


Figure 2-36: Simulation results of cables under self-weight, with axial stiffness scale of **1.0** (a) **0.1** (b).

The cable in Vortex Studio is

simulated as a chain of lumped mass elements connected by constrains; therefore, unlike in a solid beam, where the mass distribution is uniform, in a Vortex Studio cable, mass distribution is discrete therefore we cannot expect the cable taking the exact shape suggested by the model. However, Vortex Studio always offers flexibility to our users. In this case (Figure 2-36), the users change the ‘axial stiffness scale’ parameter in Dynamics Cable Properties so as to match the simulated cable with the model.

The simulation results are fairly close to the model (Table 2-3).

Table 2-3: Error between simulation and model.

Test	Absolute error in mean	Relative error in mean	Relative error in max sag
Stretching cable	0.0002	0.0046	0.9605
Simply supported cable	0.0690	0.2191	0.2208
Cantilever cable	0.0787	0.0473	0.0210

We do expect to be closer and closer to the Bernoulli model as the lumped element number increases. With a higher number of lumped elements, the simulation result (e.g., cantilever cable experiment) is found to resemble the real cable represented by the model more closely. For instance in the cantilever experiment, the relative error decreases from **0.7244** to **0.0548** when section length is changed from **2.0** to **0.125** (Figure 2-37), and of course in expense of simulation resources.

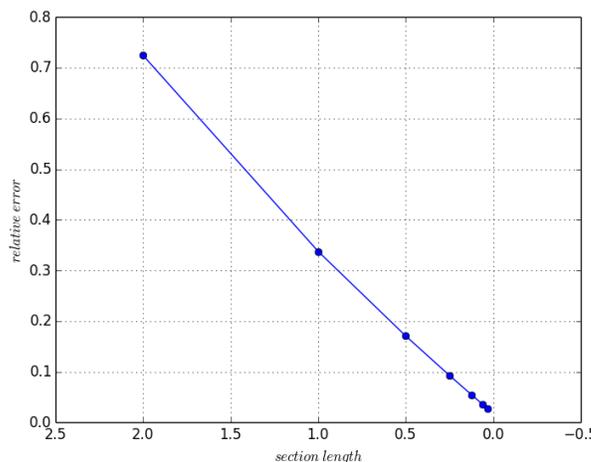


Figure 2-37: Results showing how smaller section length contributes towards better representation of cable.

2.4.4 Cable Systems: Adaptive

Test Case

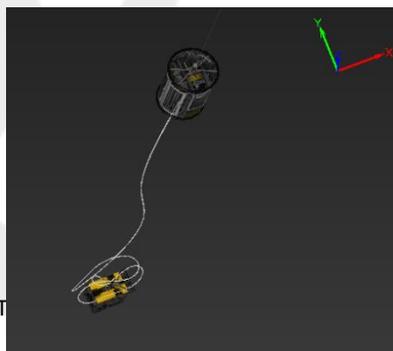
There should be a performance improvement when cable is adaptive, without a lot of compromise of fidelity.

Model

When a cable becomes longer, for its adequate representation the number of sections have to be increased accordingly. For real-time simulation, it could consume more resources. When simulation a long cable, especially when the cable is in an underwater environment, a large sections of the cable could be stationary or moving with a very small relative velocities. Such case provides a window of opportunity if we group the stationary (or with a small relative velocity) sections and simulate them as a single unit. This is the basic concept of adaptive cable.

We need to make sure that computational advantage is achieved by not compromising the fidelity much.

A scenario to mimic the real world underwater simulation is created in Vortex Studio Editor: it involves resource intensive elements such as very long cable, low and high relative velocity sections and fluid interaction. A Remote Operated Vehicle (ROV) is connected to a Tether Management System (TMS) which is connected to the ship (it is not show in the Figure 2-38) floating on the sea surface. The TMS is under a low frequency sinusoidal force to mimic the pulling force by the ship moving with the ocean surface, and the ROV is guided along a helical path.

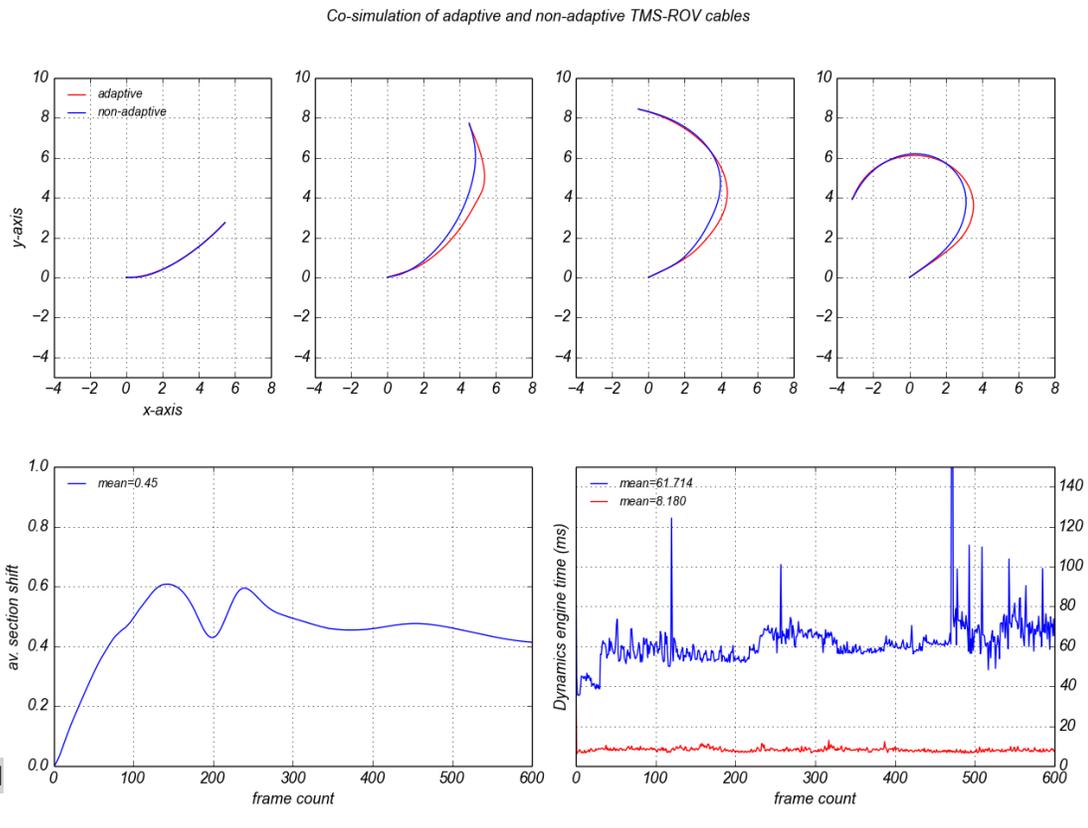


a b

Figure 2-38: Vortex Studio Editor snap shot from an underwater ROV simulation: ROV and TMS are connected through a cable (a), and details around ROV (b)

Results

The simulation runs for **10** seconds. The data from the cable connecting ROV and TMS is logged and analyzed for both adaptive and non-adaptive cases.



a b c d
e f

Figure 2-39: Comparison between adaptive and non-adaptive cables showing performance improvement.

Figure 2-39 summarizes the results from the experiment. The first row shows the cable top-view snapshots at 2, 4, 6 and 8 seconds respectively. Similarly, profiler data logged during the simulation are presented in the second row.

The comparison clearly shows that there is a noticeable performance improvement (Figure 2-39f). The dynamics engine time is reduced from **61.714** to **8.18** ms on average when the non-adaptive cable is replaced by its

adaptive counterpart. It is a significant average performance gain of 654 %. The performance gain is prominent when the system is more complex. For instance, around 8 seconds, when the cable is wrapped around the ROV (Figure 2-38b and Figure 2-39d), the solver is taking too much time in the case of the non-adaptive cable. The same complex situation is solved easily when the cable is made non-adaptive. On the fidelity front, the cable sections are deviated in position by 0.45 on average (Figure 2-39e). The deviation is expected because it needs inevitable operations such as enabling and disabling of constraints, merging and unmerging of sections, and enabling and disabling of joints when non-adaptive cable is chosen.

With the help of this experiment, it is verified that a significant performance gain can be achieved, while respecting the fidelity, by using adaptive cable.

2.4.5 Cable Systems: Fluid Interaction

Test Case

Cable system should interact realistically with the fluid. To verify the property, configurations are set up and the simulation results are presented and analysed. Rather than verifying by comparing against complex analytical model, we decided to do a visual verification.

Configuration-I: cable under fluid current

When a flexible cable segment that is tied at two ends is subjected to a fluid current, a catenary shape, in the direction of fluid flow, is observed.

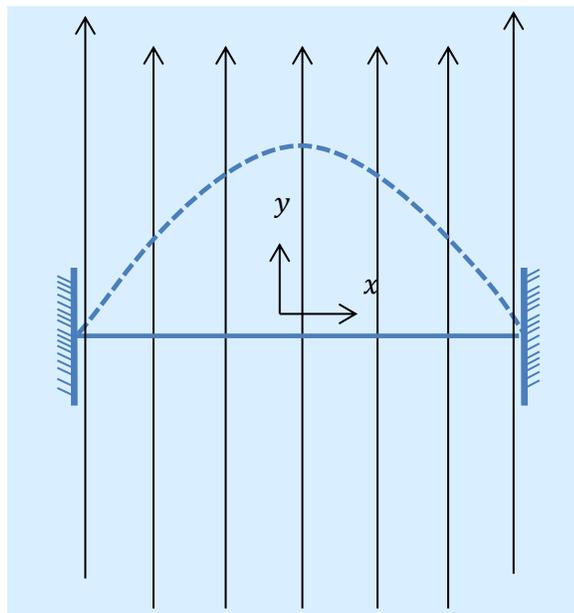


Figure 2-40: Cable catenary due to fluid current

Simulation snapshots are presented (Figure 2-41):

- The cable hangs under the gravity and takes the catenary shape when fluid interaction in the cable systems is disabled (Figure 2-41a). A sphere (initialized at origin) is also added into the simulation which eventually lands and rests on the cable. It helps to ensure that the cable is reacting to the gravitational pull.

- When the fluid interaction is enabled, the cable will be subjected under two forces: gravity and force due to fluid flow. Since the fluid velocity is dominant in the setup, the cable takes a catenary shape along the fluid flow, i.e., along $+y$ (Figure 2-41b). Also note that the sphere is out of the camera frustum because there is nothing to stop it from falling.

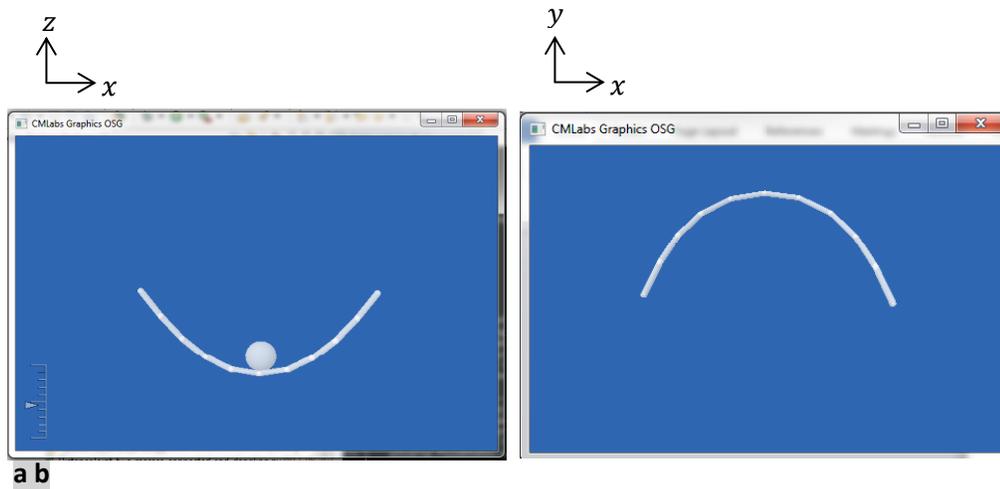


Figure 2-41: Snapshots of cable fluid interaction

Configuration-II: floating the cable

An experimental setup inside the fluid consisting of a sphere and a flexible cable hanging from a fixed attachment is created. The cable and sphere are made neutrally buoyant because a common density of 1000 is assigned to the fluid, cable, and sphere. Since the cable and sphere are fully immersed in the fluid, they stay stationary (Figure 2-42a). When the cable fluid interaction flag is suddenly disabled at runtime, the free end of the cable swings under gravity and hits the sphere (Figure 2-42b). This verifies that the cable system behaves to the relative density when it is inside the fluid.

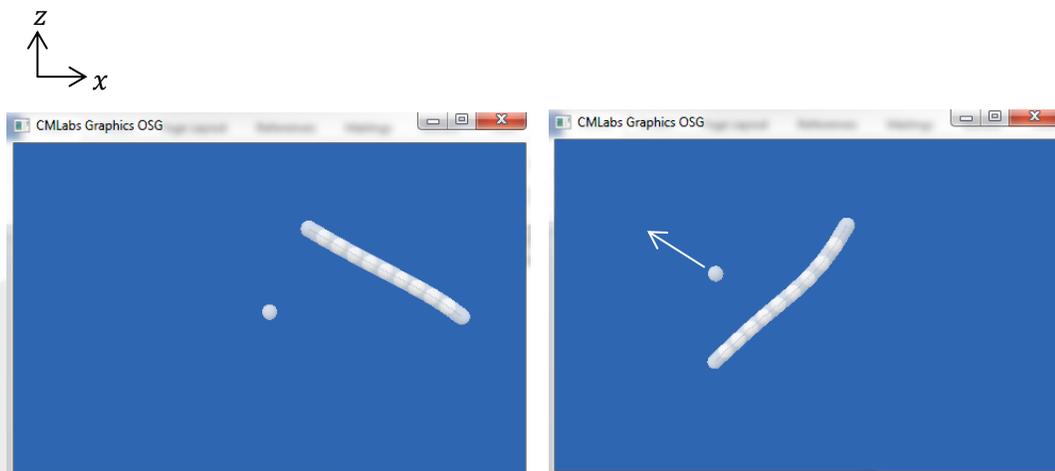


Figure 2-42: Snapshots of the experiment when the density of the fluid, containing neutrally buoyant cable and a sphere, is suddenly disabled in runtime

Configuration-III: twisting the cable

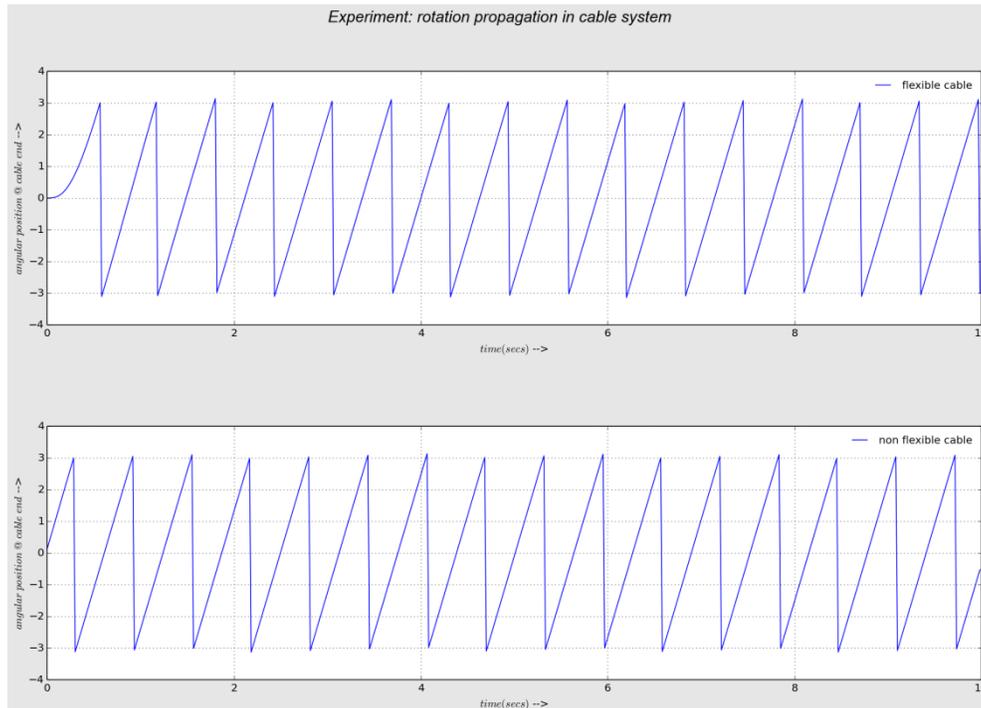


Figure 2-43: Rotation propagation in cable system

Since the cable has been realized through number of connected sections, it is interesting to see if cable rotation at one end gets transferred to the other end. The experiment is performed on both flexible and non-flexible cable types. The continuous rotation throughout the simulation helps us to confirm that the cable systems can propagate the rotation from one end to the other without jittering. It is worth noting that a flexible cable (Figure 2-43a) introduces a small initial delay because it conserves a small energy at the spring joints.

2.4.6 Cable Systems: Ring

Test Case

One of the important cable point types is ring. It can be a dynamic waypoint through which the cable passes. To affirm realistic ring simulation, it should correctly handle basic use cases such as a ring should not add extra force on the cable, nor it should alter the cable length when the ring is moved.

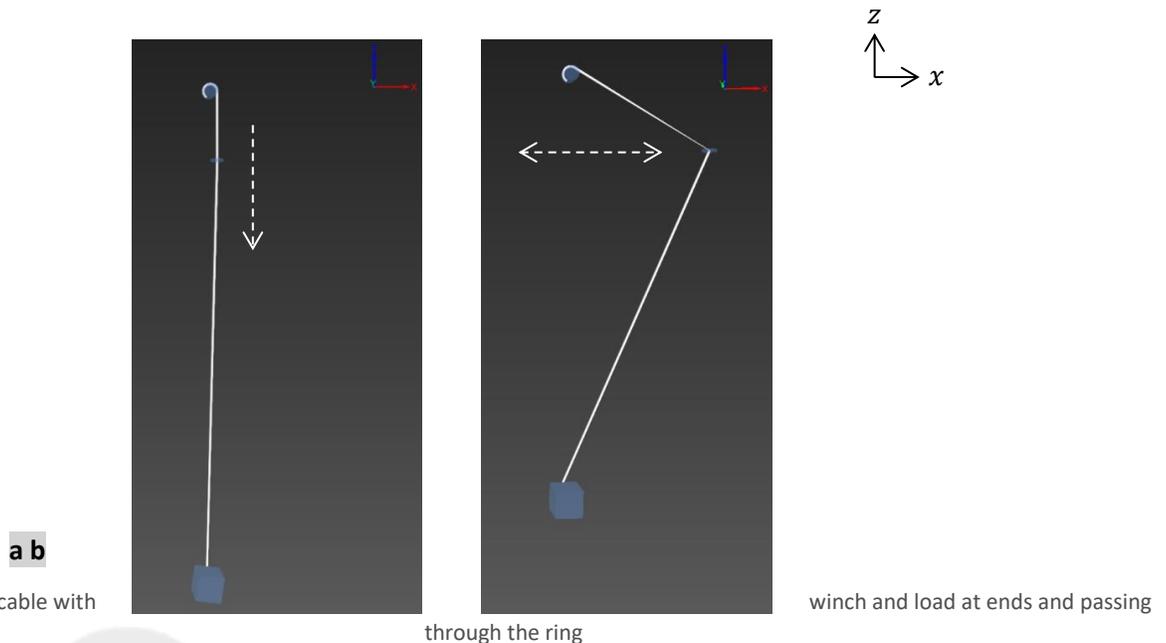
There are two scenarios and both of them contain a cable coming out of the winch, passes through a ring in the middle, and attaches to a load at the end. When the ring falls freely (indicated by the dotted arrow) under gravity, it is expected that the ring should not cause the winch to release the cable, implying the ring does not add extra force on the cable (Figure 2-44a). Similarly, if the winch is locked and the ring is moved laterally along a prismatic axis, the cable should pass through the ring but the total length of the cable should remain constant all the time (Figure 2-44b). The prominent parameters used in the scenario are as follows:

Cable: radius=0.01 m; linear density=2 kg/m; stiffness= 10×10^{20} N/m

Winch: position= $\langle x=0, y=0, z=30 \rangle$ m; radius=0.2 m; mass=50 kg

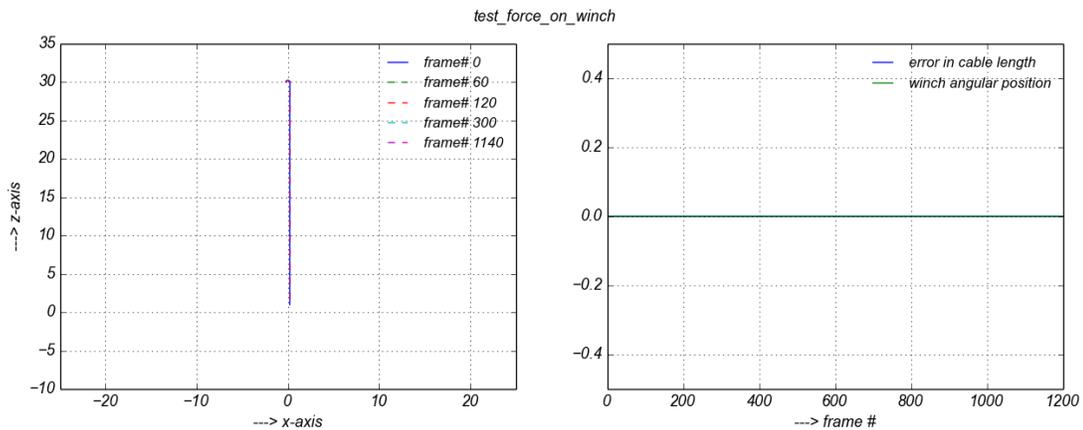
Ring: position= $\langle x=0, y=0, z=25 \rangle$ m; mass=5 kg

End load: position= $\langle x=0.2, y=0, z=0 \rangle$ m; mass=100 kg

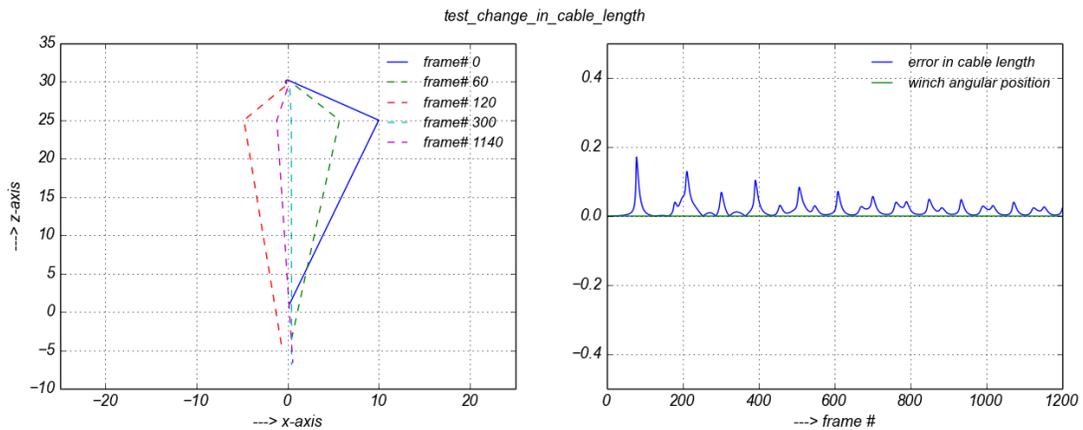


Results

The results from both of the test cases show that the ring simulation is realistic. The first test case results suggest that a free falling ring has almost no effect on a winch as its angular position is unaffected throughout the simulation (Figure 2-45a right), with only and average absolute error of 1.78×10^{-07} m. Similarly the length of the cable also remained the same.



a. Force on winch test



b. Change in cable length test

Figure 2-45: The figure showing cable snapshots at different frames and respective winch angular position and error in length

In the second test, the ring on the middle of the hanging cable is constrained to move along a prismatic axis which is orthogonal to the vertical axis. The simulation is initialized with the ring positioned a bit off the vertical axis (see frame # 0 in Figure 2-45b left), then allowed to swing under its self-load and the load at the end. Few snapshots of the cable during the simulation are presented on the left while the right figure summarizes the error in cable length. Since the cable is composed of beads of sections, while swinging, it undergoes a higher order oscillation which introduces a small error in total length (Figure 2-45b right), with an average error of 0.0186 m. These results affirm that the fidelity in cable simulation is fairly preserved. In addition, further parameter tuning in Cable Systems allows achieving custom behaviour.

2.4.7 Vehicle Systems: Ackermann Steering

Context

Testing a vehicle system is a very general task. It is generally driven by the requirements. Customers want to make sure that vehicle behaviour lies within an acceptable error tolerance. It is good practice to verify individual components so that deviation in results or possible error can be narrowed down to the specific component being tested.

The steering system could be the first place to start verifying since it plays crucial role in guiding the whole vehicle dynamics. An experiment is designed to verify if the vehicle circulates about a fixed pivot point when driven by an Ackermann steering system. For simplicity, a front-wheels-driven (FWD) vehicle turning towards the left is chosen. The variables relevant for this experiment are indicated in Figure 2-46.

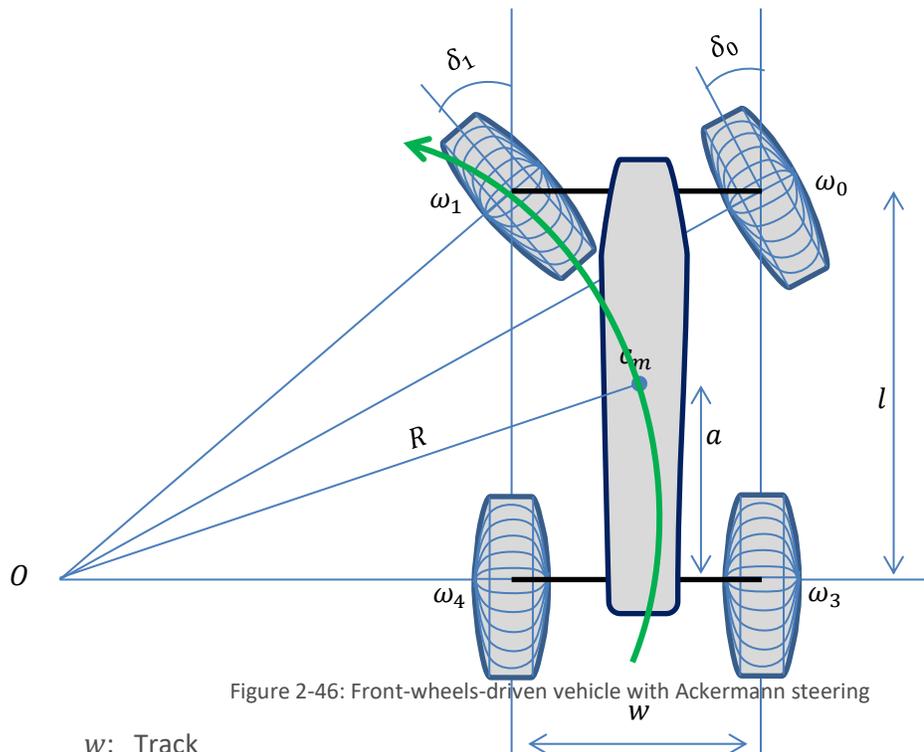


Figure 2-46: Front-wheels-driven vehicle with Ackermann steering

- w : Track
- l : Wheelbase
- a : Wheelbase to center of mass (COM) distance
- R : Radius of COM
- ω : Angular velocity of a wheel

There are various implementations of the Ackermann steering mechanism, but all of them follow a kinematic condition: the lateral axis of each wheel passes through the common center O , so that the vehicle always turns about a fixed pivot point, optimizing each wheel's trajectory. The *Ackermann condition* is expressed as

$$\text{Cot}\delta_0 - \text{Cot}\delta_1 = \frac{w}{l} \tag{2.51}$$

where δ_0 and δ_1 are the steering angles for the outer and inner wheels respectively. After few steps, the rotation radius of the center of mass, R_m , referred to as 'model radius', can be derived as¹⁸

$$R_m = \sqrt{a^2 + l^2 \cot^2 \delta} \quad (2.52)$$

where

$$\cot \delta = \frac{\cot \delta_0 + \cot \delta_1}{2} \quad (2.53)$$

¹⁸ http://www.idsc.ethz.ch/Courses/vehicle_dynamics_and_design/11_0_0_Steering_Theroy.pdf

Experiment

A vehicle preset with the following power flow scheme is generated.

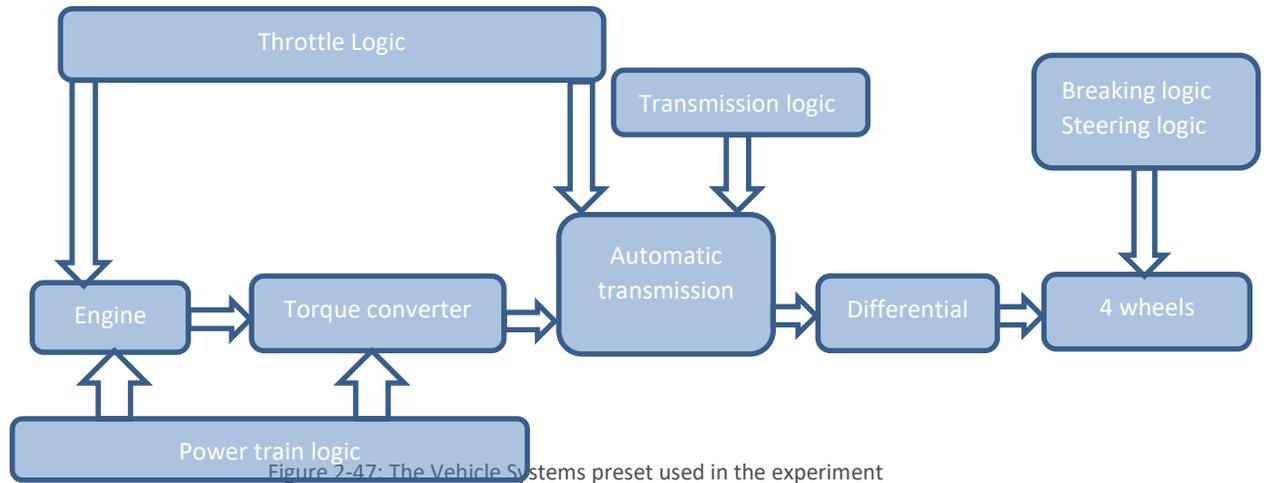


Figure 2-47: The Vehicle Systems preset used in the experiment

The instantaneous rotation radius of the vehicle, r_v , referred to as the 'Vortex Studio radius', is computed frame to frame. Let $\vec{P}_0, \vec{\theta}_0$ and $\vec{P}_1, \vec{\theta}_1$ be the position and orientation vector sets at two frames. Assuming it to be large, r_v is estimated by using simple trigonometric rule.

$$r_v = \frac{|\vec{P}_0 - \vec{P}_1|}{\vec{\theta}_0 \cdot \vec{\theta}_1} \quad (2.54)$$

Methodology

After the vehicle has reached a stable speed, different properties such as chassis speed and Vortex Studio radius are logged. The model radius is also derived for the given steering input.

The Vortex Studio radius data, which is list of radii retrieved on each frame, is not scattered around the model radius which is a single number. To gain insight into how much the vehicle deviates from the actual value when doing a circular motion, the absolute mean difference, $\left| R_m - \frac{1}{n} \sum_{i=0}^{n-1} r_{vi} \right|$, is chosen as a statistical measure.

Results

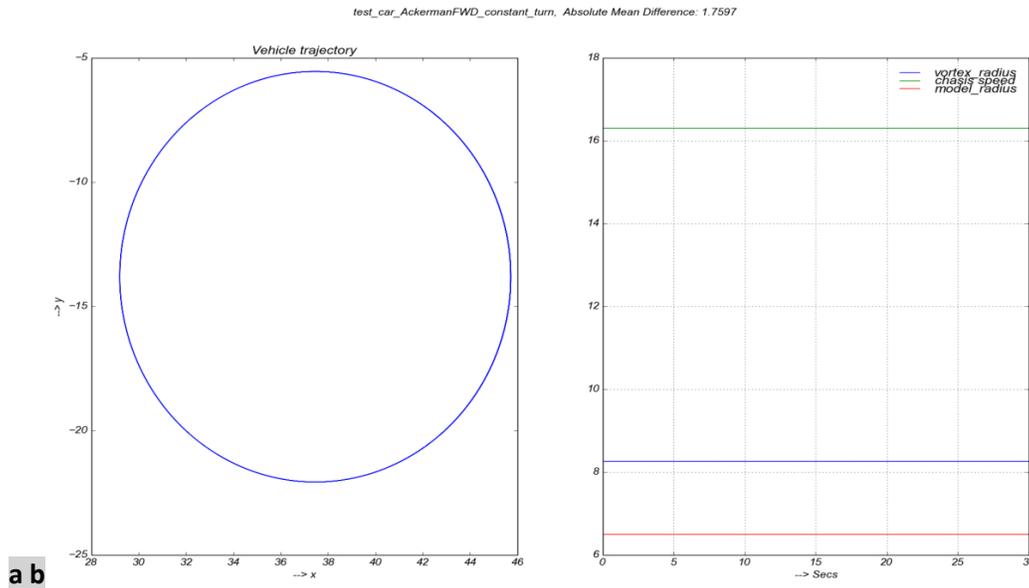


Figure 2-48: Trajectory, chassis speed and continuous radius plot of the vehicle doing a skid free circular motion

As seen in the vehicle trajectory (Figure 2-48) and the *chassis speed* on the right, the vehicle is found to be moving with a constant speed in a perfect circle. The average Vortex Studio radius is 8.2578 m which is close to what the model predicted: 6.4981 m. Absolute mean difference of 1.7597 m is therefore insignificant. There could be various factors contributing to the difference such as lateral wheel slip from the interaction between the wheel and terrain, which is an expected effect.

2.4.8 Vehicle Systems: Engine Torque Table

Context

Engine is one of the main components in a fuel powered vehicle. The primary function of the engine is to generate driving torque which is continuously transported (mechanically) to the wheels. Torque, in other words, is a twisting force. It is expressed as the product of force and distance, $\vec{\tau} = \vec{F} \times \vec{d}$, to consolidate the rotation effect of tangential force acting at a distance from the center of rotation.

One of the ways to verify proper working of the engine is to make sure that it, for a given throttle input, is producing the expected torque at observed rotations per minute (RPM). This torque data of an engine is tabulated as specification and is called a torque table. Depending on where the engine is going to be used, companies can make engines using different torque table characteristics. As an example, the torque table of the 'sports car' preset in Vehicle Systems is presented below.

Table 2-4: The torque table used in the experiment

RPM\Throttle	1	0.75	0.5	0.25	0
0	3.4	3.4	3.4	3.4	3.4
666.67	144	115.2	86.4	50.4	28.8
1000	214.4	171.5	128.6	75	0
1625	267.8	211.6	160.7	80.4	0
2250	304	237.1	182.4	76	0
2875	312	240.2	187.2	62.4	0
3500	318.4	242	178.3	35.8	0
4125	320	240	166.4	0	0
4750	307.2	227.3	107.5	0	0
5375	284.8	207.9	0	0	0
6000	252.8	182	0	0	0

It can be seen from the table that the torque data first raise almost linearly, and then it drops slowly once it reaches the peak. This restates the fact that high torque at high RPM is a contradictory demand. For all throttle sets, RPM versus torque plot forms a discrete surface, with the independent variable being throttle and RPM.

Experiment

When the vehicle preset is loaded in Vehicle Systems, the corresponding torque table is also loaded at the same time. During the simulation, the vehicle should produce the same amount of torque for a given throttle and RPM. For that, a simplified vehicle preset with minimum components is created.

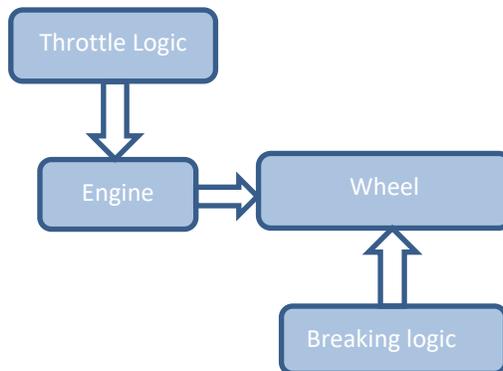


Figure 2-49: The Vehicle Systems preset used in the experiment

Methodology

The simulation is run with throttle set, $\mathbb{T} = \{1.0, 0.75, 0.5, 0.25, 0.0\}$ and RPM set

$\mathbb{r} = \{0.00, 666.67, 1000, 1625, 2250, 2875, 3500, 4125, 4750, 5375, 6000\}$, Therefore the torque table, a two dimensional function $f(\{t, r | t \in \mathbb{T}, r \in \mathbb{r}\})$, is a surface plot.

Each time the engine, starting from rest, is run at throttle t until it attains r RPM, and the corresponding torque τ_v value is read back. Because simulation is run for discrete time steps, the engine may not be running exactly at r RPM during the simulation. In such situation, the torque is estimated by linear interpolation. Now to compare the list of (t, r, τ_v) triplets against the model supplied through torque table (table 2.17) and to see how the engine torque follows the model, Root Mean Squared Error (RMSE),

$$\sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (\tau_v - \tau_m)^2} \tag{2.55}$$

where τ_m is the model torque, is chosen.

Results

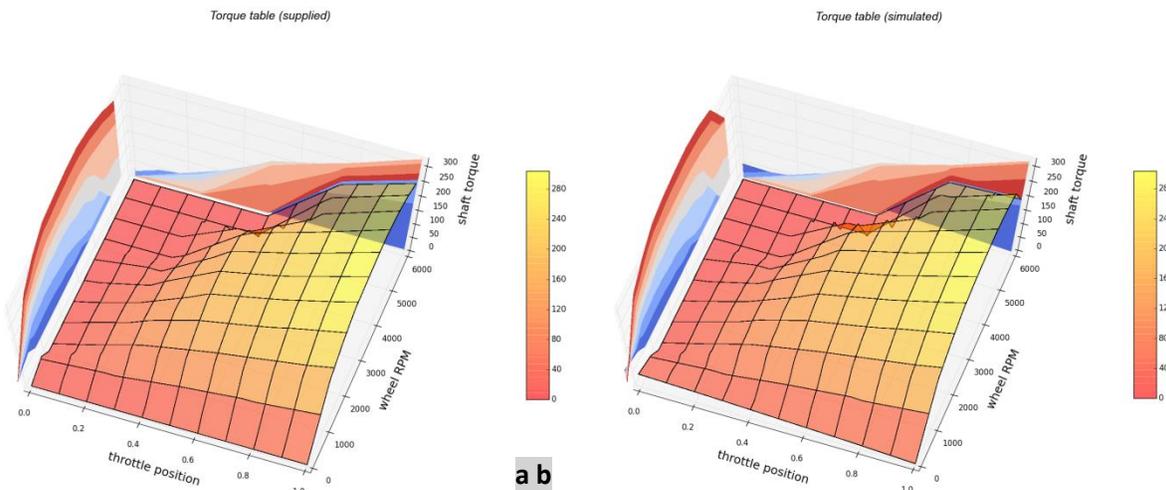


Figure 2-50: Engine torque tables: supplied and simulated

Figure 2-50 refers to the torque table which is supplied along with the preset. It is considered the model, while Figure 2-50b is the torque data logged during the simulation. It is impressive that the surface profiles are almost identical, except for small discrepancies around higher RPM and throttle values, which are summarized by RMSE of 7.1607. This is acceptable for an engine that runs (0-6000 rpm) and produces the torque in the range of (0-320 Nm).

2.4.9 Vehicle Systems: Synchronizer

Context

In the transmission mechanism, input and output shafts will be, in general, spinning at different speeds. During gear shift, if we try to engage them directly, gears' teeth will clatter, possibly wear, and not engage. To overcome this mechanical difficulty, modern gearbox has a synchronization mechanism for smoother and faster transition of the rotational speed of the input and output shaft.

Synchronizer is a complex mechanical arrangement. However, from the Vehicle Systems perspective, its functionality can be modeled as

$$Transmission\ status = \begin{cases} Synchronized & \text{if } \omega_r \leq e_\omega \\ failed & \text{otherwise} \end{cases} \quad (2.56)$$

where,

ω_r : Relative speed between transmission input and output $\omega_r = |\omega_o - \omega_i|$

e_ω : Speed tolerance of the synchronizer. It is governed by several physical parameters such as material, gear teeth shape, operating temperature and transmission fluid viscosity.

When the gear is shifted, friction developed synchronization torque, τ_s , is applied to the shafts for a duration of synchronization time, t_s . Using the fact that the application of external torque causes the change in angular momentum, angular speed, after t_s seconds on both sides, are projected:

$$\begin{aligned} \tau_s &= \frac{I \cdot \omega_t}{t_s} - \frac{I \cdot \omega_0}{t_s} \\ \Rightarrow \omega_t &= \frac{\tau_s \cdot t_s + I \cdot \omega_0}{I} \end{aligned} \quad (2.57)$$

where

ω_0 : Angular speed at the beginning of synchronization

ω_t : Angular speed after t_s seconds

I : Shaft inertia

During the synchronization, the low speed shaft undergoes acceleration and the high speed shaft slows. Once both input and output final angular speeds estimates are in calculated (2.57), the condition $|\omega_t^{input} - \omega_t^{output}| \leq e_\omega$ is checked to forecast the synchronization state: synchronized or not.

Experiment

The verification of the synchronization process in manual transmission logic is the objective of this experiment. The synchronization is active when the lever position is changed from one gear to other. During the synchronization, transmission shafts rotating at different angular speeds will be brought closer in speed by the application of synchronization torque.

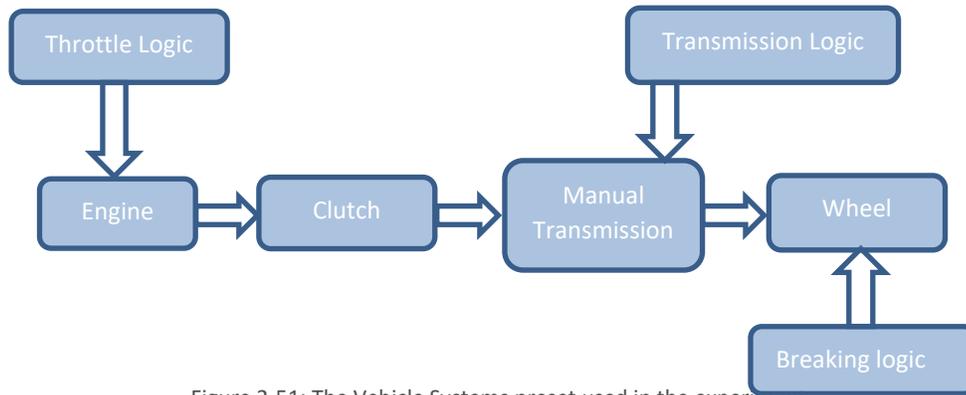


Figure 2-51: The Vehicle Systems preset used in the experiment

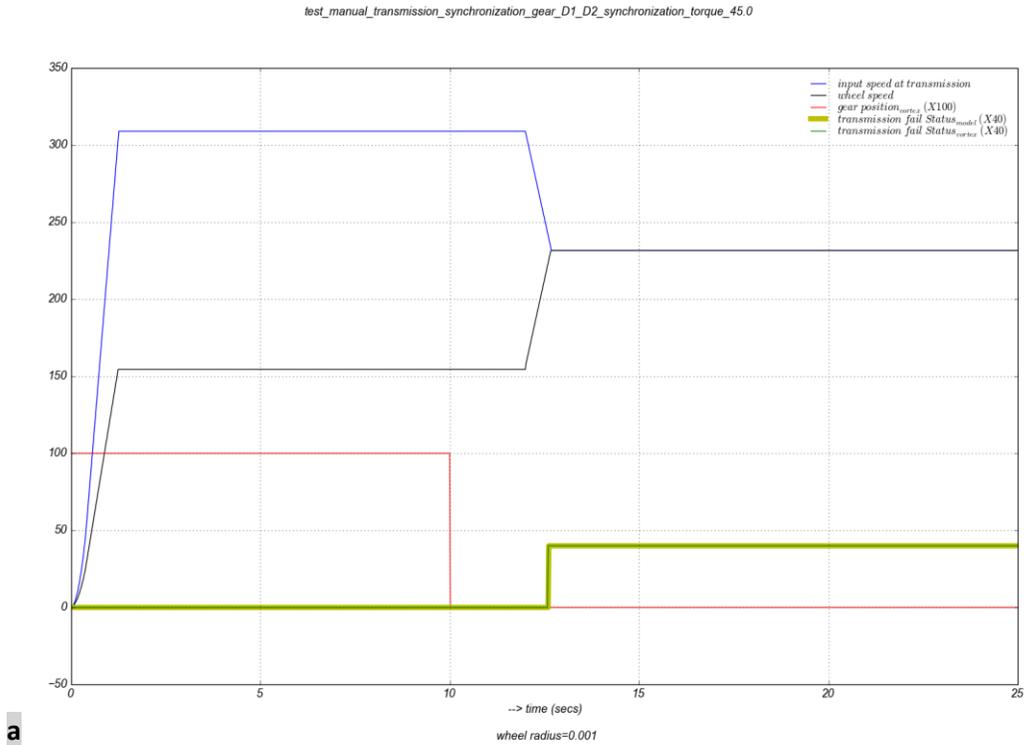
Methodology

A Vehicle Systems preset is loaded with default parameters, and the simulation is started in first gear.

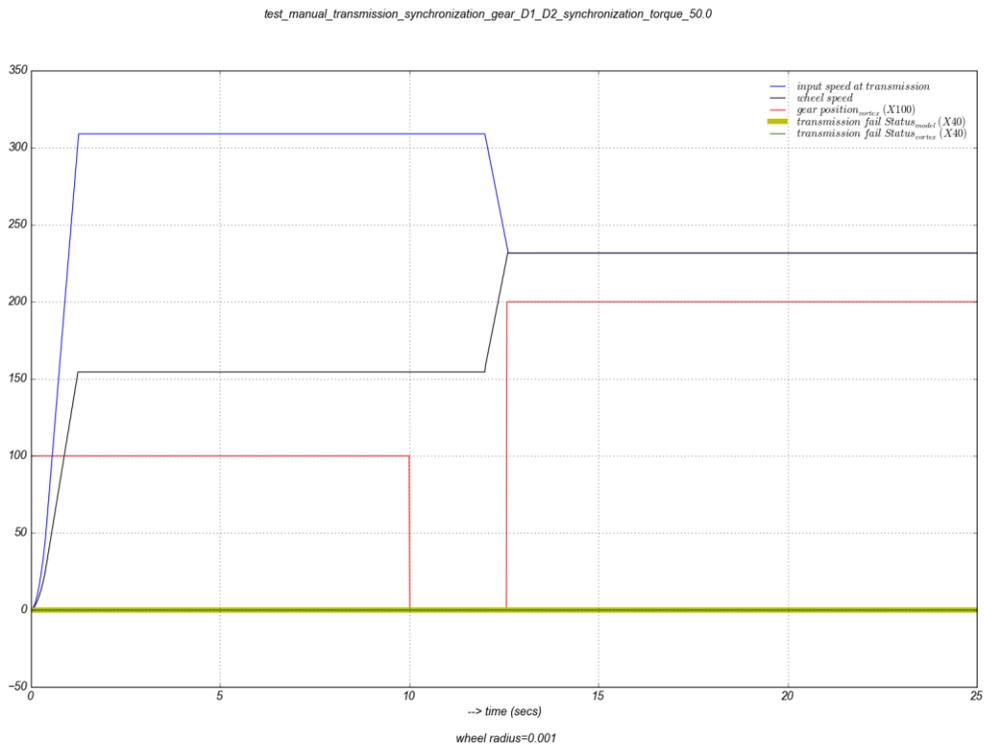
Once a stable speed is reached, the clutch is disengaged to isolate the transmission from the engine. The lever is first changed to neutral, and after a second it is moved to the second gear. The experiment is repeated for different synchronization torque set $\mathbb{T} = \{30, 40, 45, 50, 55, \dots\}$. Relevant information such as the transmission input and output speed, gear position, and transmission status is logged throughout the simulation.

Results

Note that the synchronization process is completed in less time as the increase in synchronization torque. Figure 2-53 depicts two cases: not synchronized and synchronized. In the first case, the synchronization failed and the gear position is not successfully shifted to the target gear because, within synchronization time, the synchronization torque is not enough to bring input-output relative shaft speed within tolerance. However, in the other case when the synchronization torque is increased, the shafts are successfully brought together enough to be synchronized (see sample calculation). Since synchronization state is the outcome of binary decision, the decision is correct for a large range of synchronization torque values. For the given configuration, there always exists a minimum torque value which guarantees the synchronization. Around this minimum synchronization torque, it is also interesting to note that in a very narrow torque window of 1.26 (=49.58-48.32), the simulation is unable to follow the model. Considering several factors such as the number of rich computations involved in cascaded modules and the chosen frame rate, this deviation is considered acceptable.



a



synchronization torque on transmission

Figure 2-52: Effect of

Sample calculation:

Case-I: NOT synchronized (Figure 2-53a)

```

synchronization torque: 45.0
synchronization time: 0.6
synchronization tolerance: 10.0
gear ratio: 1.0
input shaft inertia: 0.4
output side inertia: 0.4125
input shaft speed (actual)@ the beginning of transmission: 308.9232775
wheel speed (actual)@ the beginning of transmission: 156.332197711

input shaft speed(actual) @ the end of transmission 241.423277114
wheel speed(actual) @ the end of transmission 223.537198169
shaft relative velocity(actual) @ the end of transmission 17.8860789453
transmission fail flag status(actual)@ end of transmission 1.0 (Failed)

input shaft speed(predicted) @ the end of transmission: 241.4232775
output shaft speed(predicted) @ the end of transmission: 221.786743166
shaft relative velocity(predicted) @ the end of transmission: 19.6365343341
transmission fail flag status(predicted): 1.0 (Failed)

```

Case-II: **synchronized** (Figure 2-53b)

```

synchronization torque: 50.0
synchronization time: 0.6
synchronization tolerance: 10.0
gear ratio: 1.0
input shaft inertia: 0.4
output side inertia: 0.4125
input shaft speed (actual)@ the beginning of transmission: 308.9232775
wheel speed (actual)@ the beginning of transmission: 156.540037606

input shaft speed(actual) @ end of transmission 233.923277106
wheel side speed(actual) @ end of transmission 231.537628318
shaft relative velocity(actual) @ the end of transmission 2.38564878758
transmission fail flag status(actual)@ the end of transmission 0.0 (Synchronized)

input shaft speed(predicted) @ the end of transmission: 233.9232775
wheel speed(predicted) @ the end of transmission: 229.267310333
shaft relative velocity(predicted) @ the end of transmission 4.65596716663
transmission fail flag status(predicted): 0.0 (Synchronized)

```

Notes

- For easy visualization, some plots are scaled
- SI units are chosen for all parameters
- All speeds are in angular form

2.4.10 Vehicle Systems: Clutch**Context**

The clutch is a vehicle component that rests between the engine and transmission, and facilitates the engagement and disengagement of the power transmission. In the Vehicle Systems, it is modeled on Coulomb's friction model¹⁹. For simplification purposes, thermal effects do not have an impact since they have a low dynamic. The transfer function can be written as

$$T_{out} = \begin{cases} T_{in} & \text{if } \omega_r = 0 \\ T_d \cdot \text{Sign}(\omega_r) & \text{otherwise} \end{cases} \quad (2.58)$$

where

T_{in} : Input torque

ω_r : Relative RPM

T_d : Dynamic friction torque, T_d , which is computed using a normalized clutch position signal, c , as a fraction of the maximum friction torque: $T_d = c \cdot T_{max}$

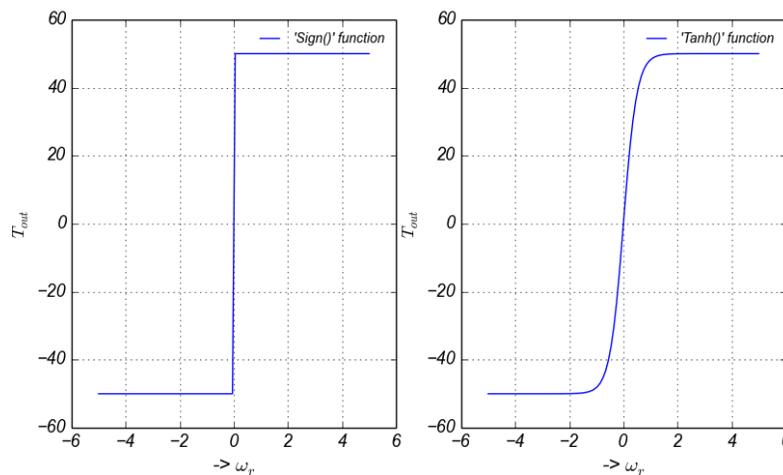


Figure 2-53: Friction torque variation w.r.t relative velocity

Since

creates the non-causality issue, the model is often simplified by using hyperbolic function, $Tanh$, to ensure the smooth transition around origin. It makes the system not only casual and numerically stable but also mimics the viscous friction model. So the model is modified to

$$T_{out} = T_d \cdot \text{Tanh}\left(\frac{2\omega_r}{\omega_0}\right) \quad (2.59)$$

where ω_0 is the parameter that determines the speed of the transition from -1 to 1. Figure 2-53 shows how the clutch output torque (with $c = 1$) differs when it is approximated by $Sign$ and $Tanh$ functions.

Experiment

A vehicle preset with the following power flow scheme is generated and used for the experiment.

¹⁹ BĂȚĂUȘ, Marius, et al. "Automotive Clutch Models for Real Time Simulation." *Proceedings of the Romanian Academy, Series A: Mathematics, Physics, Technical Sciences, Information Science*.

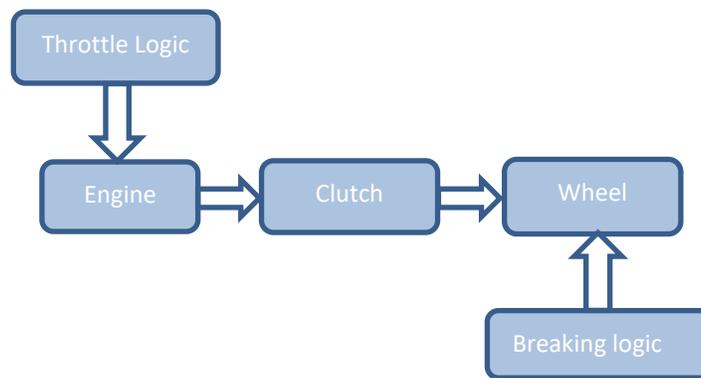


Figure 2-54: Vehicle systems preset used in the experiment

Methodology

The engine is simulated for a given duration and with full throttle, and the clutch output torque, T_{out} , and relative RPM, ω_r , which is the PRM difference in clutch input and output, are logged. The experiment is repeated for set of clutch positions $\mathbb{c} = \{1.0, 0.75, 0.5, 0.25, 0.0\}$.

Results

Figure 2-55 shows the torque output against the relative RPM plot and torque RPM profile against the simulation time plot respectively.

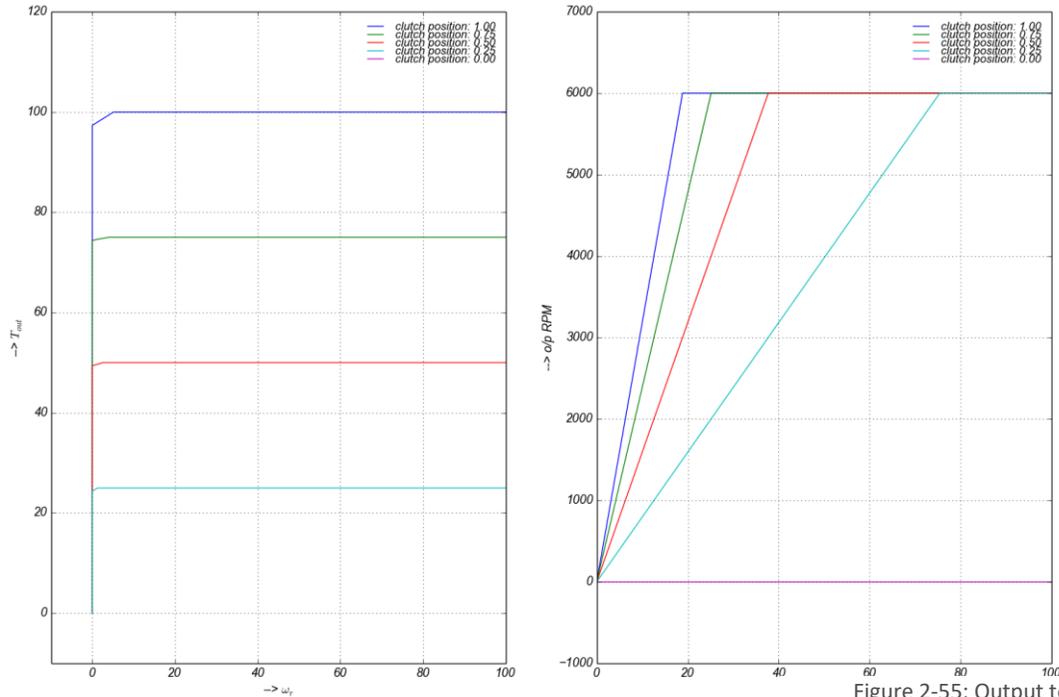


Figure 2-55: Output torque and RPM plot of a clutch

a b

- For all clutch positions, the steady state torque closely follows the model (2.58) except for discrete transitions around low relative RPM, which is an expected behaviour in a discrete time step simulation (Figure 2-55). For instance, when clutch (maximum torque of 100) is positioned at 0.5, the RMS error (2.55) of only 0.6374 in clutch torque compared with the model is found.
- For clutch position 0.0, torque output is zero since there is no coupling (Figure 2-55).
- The fact that wheel at clutch output should be constantly accelerating until it attains the input RPM is verified and depicted in Figure 2-55.
- The higher the clutch position, the coupling takes place in lesser time (Figure 2-55).

2.4.11 Vehicle Systems: Transmission

Context

Engines provide torque. Depending on the driving condition, torque has to be transported to the wheels in different proportion. Since power is the product of torque and speed, higher torque and higher speed outputs are contradictory demands in a system with constant power input. So, transmissions play an important role in

controlling the torque by altering the speed. A typical transmission containing gears and gear trains assembly is presented in Figure 2-56²⁰.



Figure 2-56: a typical example of gears - a five-speed + reverse gearbox from the Volkswagen Golf 1600cc (2009)

According to the speed requirements at the output end, the input power is routed through corresponding gear mechanism offering an input output speed ratio – gear ratio. While doing so, the torque is compromised. If there is a speed gain, torque at the output will be reduced and vice versa. In a lossless system, power remains constant.

$$\tau_i \cdot \omega_i = \tau_o \cdot \omega_o \quad (2.60)$$

$$\text{Gear ratio} = \frac{\tau_i}{\tau_o} = \frac{\omega_i}{\omega_o} \quad (2.61)$$

where

τ_i and ω_i : input torque and angular speed
 τ_o and ω_o : output torque and angular speed

In other words, gear ratio is how many times the input shaft should rotate to get a single rotation at output shaft. To achieve different speeds, input and output shafts are routed through a mechanical arrangement that offers the defined gear ratio for the given lever position.

²⁰ Picture source: [http://en.wikipedia.org/wiki/Transmission_\(mechanics\)](http://en.wikipedia.org/wiki/Transmission_(mechanics))

Experiment

Vehicle systems offer two types of transmission mechanisms: manual and automatic. In manual transmissions, the user should manually change from one gear to another to meet his torque and speed requirements while gear shifting automatically takes place at pre-defined RPM set in an automatic. These gear shifting points are normally selected at some percentile of the maximum RPM that the engine can offer. For instance, the vehicle preset in this experiment uses the following *gear change parameters*:

- Gear quick change up: 0.95
- Gear change up: 0.70
- Gear quick change down: 0.70
- Gear change down: 0.45

Quick change parameters are used when a high speed gradient is requested. To verify the proper working of these transmission systems, three experiments are conducted:

- a. **Automatic gear up down:** The experiment is to verify the gear change in automatic transmission is happening at pre specified RPM sets, as defined by the *gear change parameters*.
- b. **Automatic transmission gear ratio** and
- c. **Manual transmission gear ratio:** In these experiments, the simulation is run to see if at a given lever position, the input and output speeds of the transmission are in line with corresponding gear ratio.

Figure 2-57 shows schematic diagrams of vehicle preset that are used in the experiments.

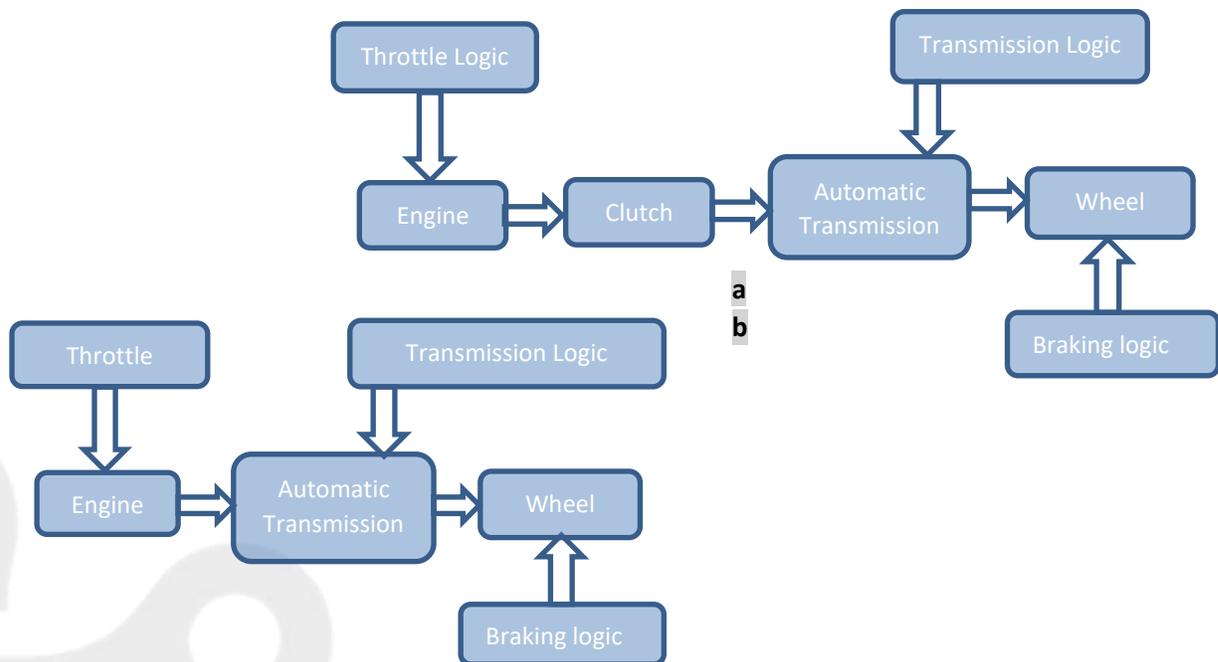


Figure 2-57: Vehicle systems presets used for transmission verification for: (a) manual transmission and (b) automatic transmission

Methodology

The vehicle systems presented in Figure 2-57 are run with a constant throttle for the given duration and relevant parameters are logged for analysis. In both transmission types, wheel RPM is considered as output RPM since transmission output shaft is directly connected to it. Gear ratio set

{-3.1, 0.0, 3.2, 2.1, 1.5, 1.0} are set at lever positions {R1, N, D1, D2, D3, D4}.

Results

Automatic gear up/down experiment

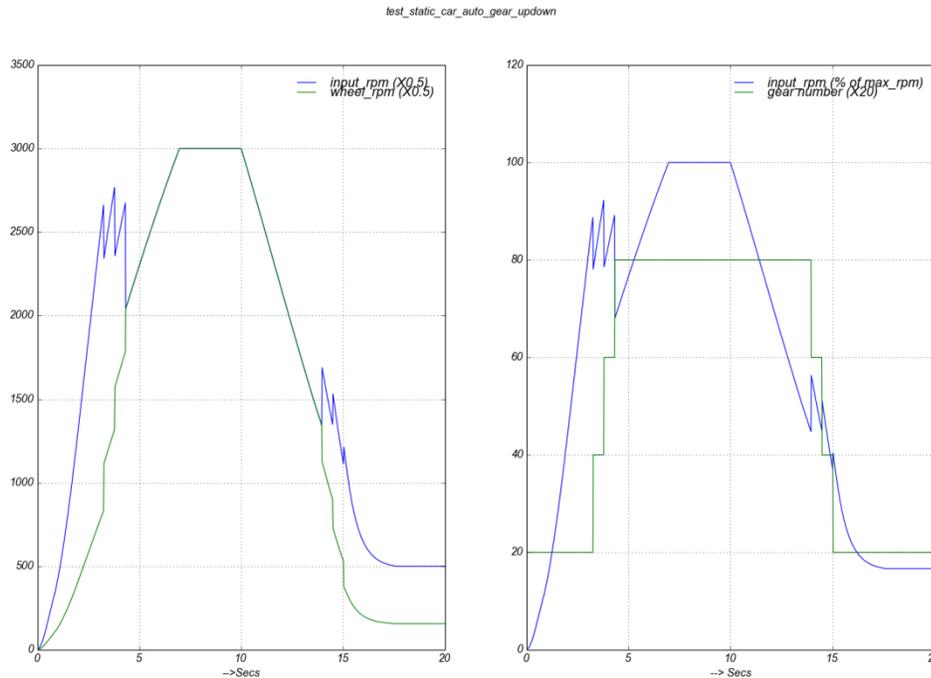


Figure 2-58: Figure 2-59: Effect of gear change in automatic transmission on RPM

The throttle is set to 0.9. The vehicle is allowed to accelerate and to shift the gear up quickly, and at the halfway the throttle is stopped so that it decelerates and gear will shift down automatically. The attributes before and after the gear change are logged and presented in Table 2-5. As expected by setting parameters (gear quick change up: 0.95, gear change down: 0.45), the gear quick up and regular down are occurring around 95% and 45% of the maximum engine speed with relative errors or 5.2% and 6.0% respectively, which is acceptable considering the complexity of the system.

Table 2-5: automatic gear shift

Input RPM	Gear	up/down	% of max RPM	throttle
5325.68906	1		0.88761484	0.9
4686.0135	2	Up	0.78100225	0.9
5532.72478	2		0.9221208	0.9
4716.26742	3	Up	0.78604457	0.9
5349.19497	3		0.89153249	0.9
4084.00151	4	Up	0.68066692	0.9
2688.07509	4		0.44801251	0
3378.18092	3	Down	0.56303015	0
2698.19632	3		0.44969939	0
3063.37651	2	Down	0.51056275	0
2224.74115	2		0.37079019	0
2425.18358	1	Down	0.40419726	0

a. Automatic transmission gear ratio experiment

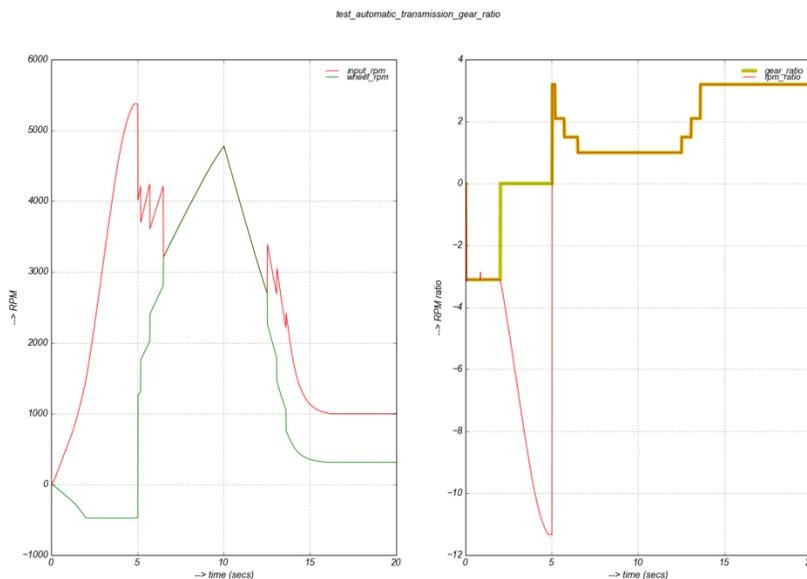


Figure 2-60: Effect of gear change in manual transmission on RPM

The input / ratio at any instant and gear ratio associated with the selected lever position should be the same as tested in this experiment. Similar to the previous experiment, the vehicle is simulated with constant throttle, and allowed to accelerate and decelerate. The plot on the right verifies the claim as the RPM ratio and gear ratio plots throughout the simulation are exactly overlapping, except during neutral. In neutral, the lever is brought to neutral from reverse gear. The decoupled wheel therefore will still be rotating freely, as the wheel is not touching the ground, in the configuration.

basic fact that output RPM

b. Manual transmission gear ratio experiment

The simulation is repeated for each lever positions: {R1, N, D1, D2, D3, D4}. Depending on the selected lever position, output settles to a RPM value (Figure 2-61a) such that its ratio with the input belongs to the gear ratio set $\{-3.1, 0.0, 3.2, 2.1, 1.5, 1.0\}$ (Figure 2-61b). Transition at the beginning of each plot is because of engine ramp up time.

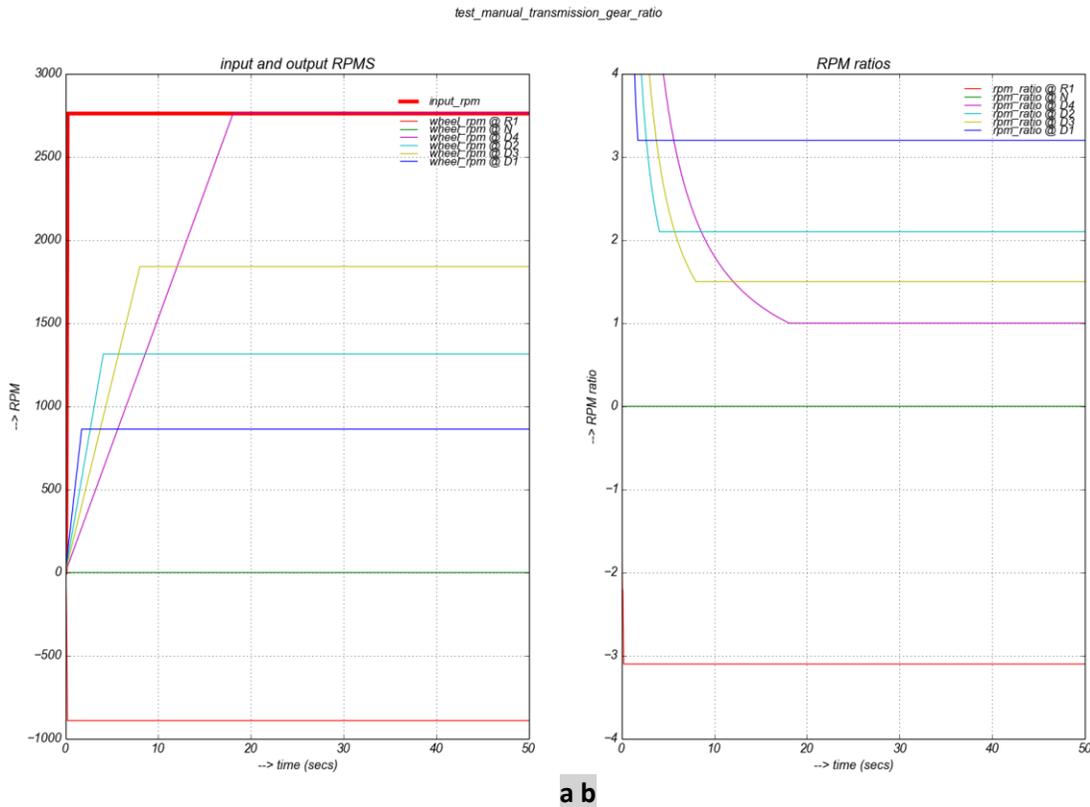


Figure 2-61: relation of gear and rpm ratio in manual transmission

2.5 Constraints

This section explains how several constraints available in the Vortex Studio API are verified. The scenarios in each experiment are designed using the constraint to be verified. This way the underlying constraint can be verified by examining its behaviour.

2.5.1 VxBallAndSocket

Test Case

A pendulum configured using a ball and socket constraint should be consistent on its length and period of oscillation, irrespective of its initial position.

The objective of this test is not to check the accuracy of a single run, but to see the consistency among different runs using different initial positions. No analytical model is involved.

Results

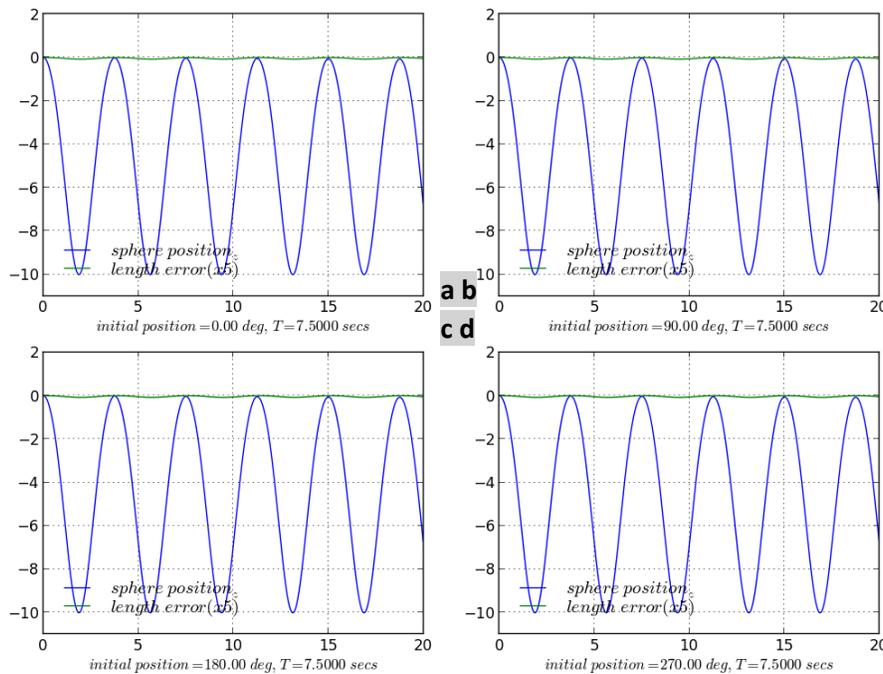


Figure 2-62: Validating ball-and-socket constraint: consistency in Time period & length of the pendulum

As expected all Figure 2-62a-d are exactly same implying the initial angular position has no effect in oscillation, also the error in the pendulum length due to non-ideal stiffness on the limits is negligible. The sag is at maximum when the bob is just below the pivot. Because the vertical component of gravity becomes the highest at this position, the attachment point will have the maximum shift.

2.5.2 VxDistanceJoint

Test Case

Distance constraint should maintain the specified distance between attachments.

Two masses are hung in a chain, each using the distance constraint: first part is at 5 units distance from the world origin, and the other is at the same distance from the first part.

Results

The distance between the attachments is maintained until a very large mass is used. When the mass is extremely large (~1 million tons), there is a damped oscillation before settling, and final resting position is slightly shifted down by ~ 2 (from -10 to 12) units. This is because of the non-ideal default stiffness and damping of the constraint.

Check: As the default stiffness of the constraint limit is very large, the solver overwrites it. So, stiffness for this check has to be derived from constraint linear compliance ($=1 \times 10^{-10}$) from the Solver Parameters. Let us restate that a spring displacement due to a load is

$$mg = k_{eq} \cdot \Delta x \Rightarrow \Delta x = \frac{mg}{k_{eq}} \tag{2.62}$$

where k_{eq} is the equivalent stiffness of two springs. Now default solver linear stiffness ($=1 \times 10^{+10}$) when put twice in series equates $k_{eq} = \frac{k}{2} = 0.5 \times 10^{+10}$. Now the deviation computed using (2.62) becomes $\frac{1000001 \times 10^3 \times 9.81}{0.5 \times 10^{+10}} = 1.962 \approx 2$. NB: symbols have usual meanings.

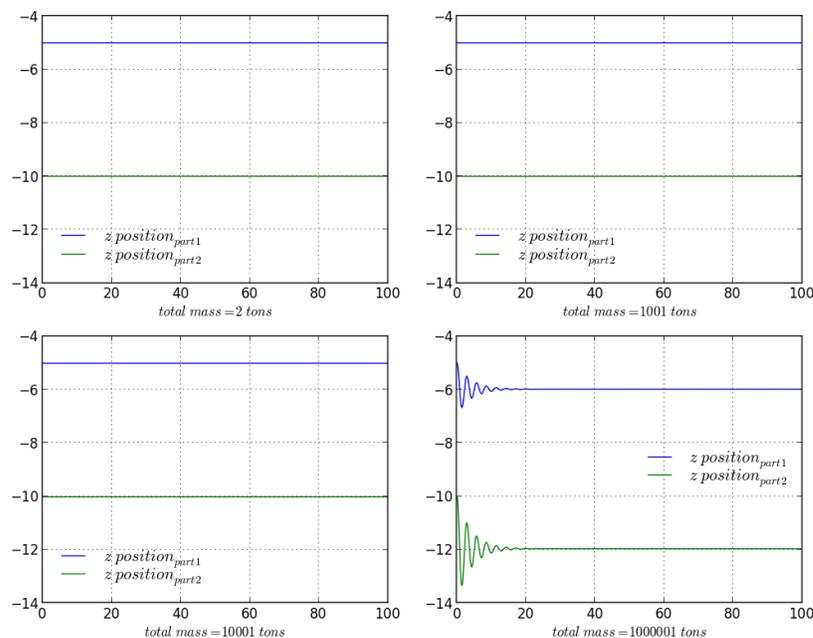


Figure 2-63: Validating VxDistanceJoint constraint: positions of connected mass using the constraint

$$I' \frac{d^2\theta}{dt^2} + c \frac{d\theta}{dt} + k\theta = mg\cos\theta.l \tag{2.63}$$

If both stiffness and damping are considered zero, it models the part before it hits the limit.

$$I' \frac{d^2\theta}{dt^2} = mg\cos\theta.l \tag{2.64}$$

Now, by controlling damping ratio (2.30), different behaviours can be seen. If the part is bounced back outside the limit, it again follows ballistic motion, with the exit position and velocity as its initial condition.

Results

Overlapping model and Vortex Studio responses for all the damping ratios is the proof that Vortex Studio is following what is expected by the model.

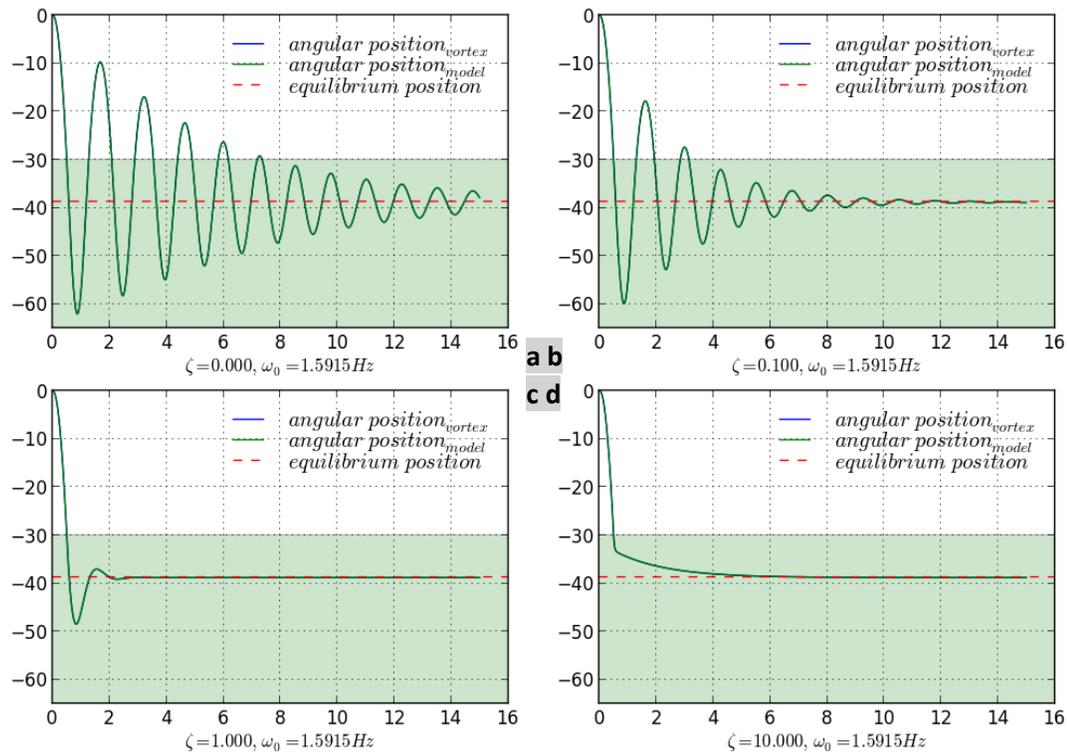


Figure 2-65: Validating VxHinge constraint: all four behaviour cases (governed by damping ratio ζ) of a mass around the limits

- a) undamped, $\zeta = 0$
- b) damped, $0 < \zeta < 1$
- c) critically damped, $\zeta = 1$
- d) over damped, $\zeta > 1$

Assumptions/notes

No air resistance is considered.

2.5.4 VxHomokinetic

Test Case

Velocity mapping between parts connected using *VxHomokinetic* should be linear.

Model

VxHomokinetic is a constant velocity joint: the input and output shaft would couple linearly irrespective of the axial shift. In other words, the same instantaneous angular velocity from input shaft is transferred to the output shaft.

$$\forall \beta: \omega_o(\beta) = \omega_i(\beta) \tag{2.65}$$

Where,

- ω_i : Input velocity
- ω_o : Output velocity
- β : Angle between planes

Results

It can be seen from the Figure 2-66 that the Vortex Studio and model outputs are overlapping with each other implicating that Vortex Studio follows the model for all axial shifts.

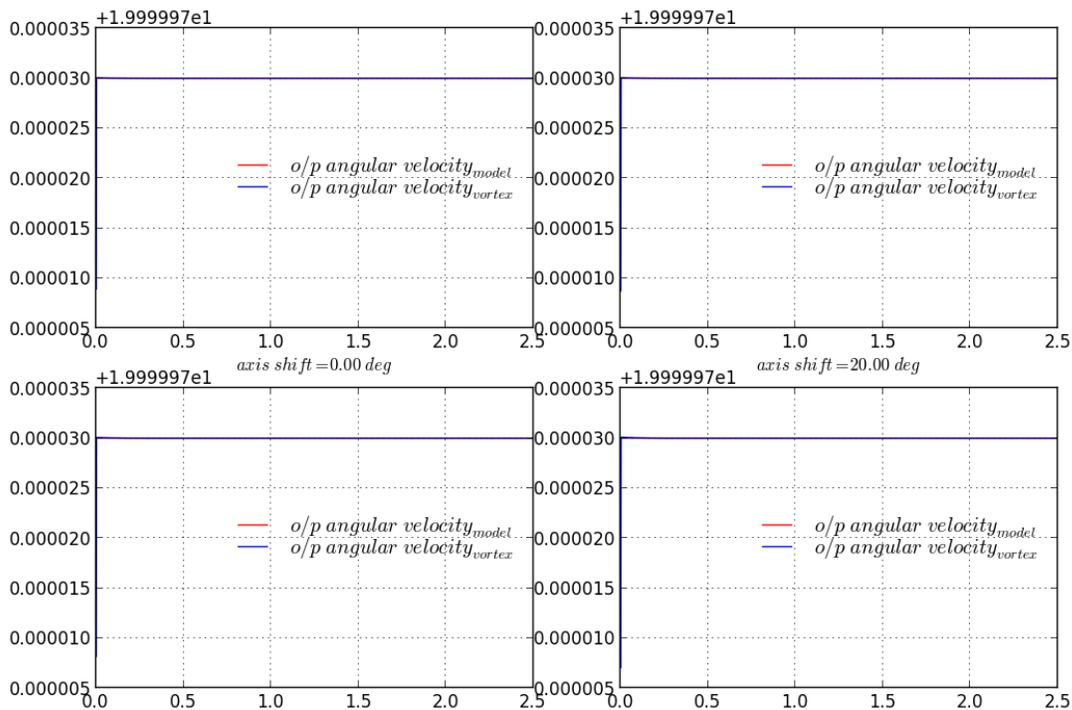


Figure 2-66: Validating VxHomokinetic constraint: making sure there is a linear coupling between input and output velocities. Configuration: input velocity = 20 rad/sec

2.5.5 VxPrismatic

Test Case

The test should validate springy behaviour of *VxPrismatic* constraint limits.

Model

The trajectory of an object, hanging from a static pivot using vertically configured *VxPrismatic* constraint, can be analyzed into two systems: ballistic and Mass-Spring-Damper (MSD).

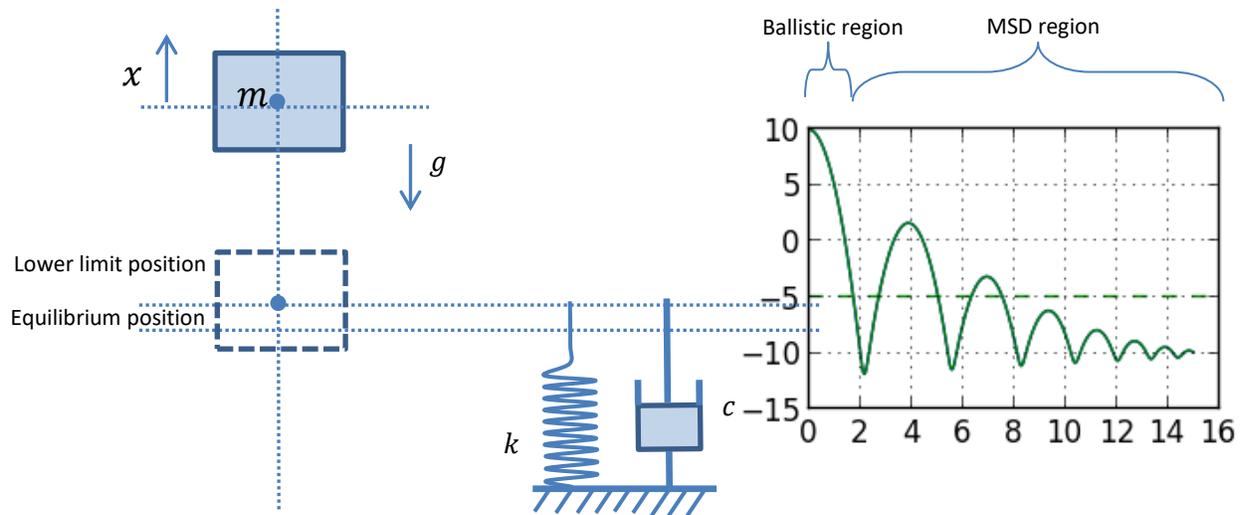


Figure 2-67: Validating VxPrismatic constraint: behaviour of a part around the constraint limit

- m : Part mass
- k : Stiffness of the limit
- c : Damping of the limit
- g : Gravity
- $x(t_i)$: Position at $t = t_i$

The object shows the ballistic behaviour until it hits the limit. The position of the projectile is modeled by classic motion equation $m \frac{dx^2}{dt^2} = mg$, and the solution is

$$x(t_i) = x(t_0) + (t_i - t_0) \cdot v(t_0) + \frac{1}{2} g (t_i - t_0)^2 \tag{2.66}$$

When it is inside the limit, it becomes a MSD system. The dynamics is, therefore, a solution of 2nd order linear differential non-homogeneous equation.

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = mg \tag{2.67}$$

Now, by controlling damping ratio ζ (2.30) we should see different behaviours. If the part is bounced back outside the limit, it again follows ballistic motion, with the exit position and velocity as its initial condition.

Results

Figure 2-68 shows the part position for different damping ratio values. As expected, when damping ratio is zero, there is a natural oscillation (some damping is due to the implicit solver in use) and the oscillation damps increase as damping ratio is increased.

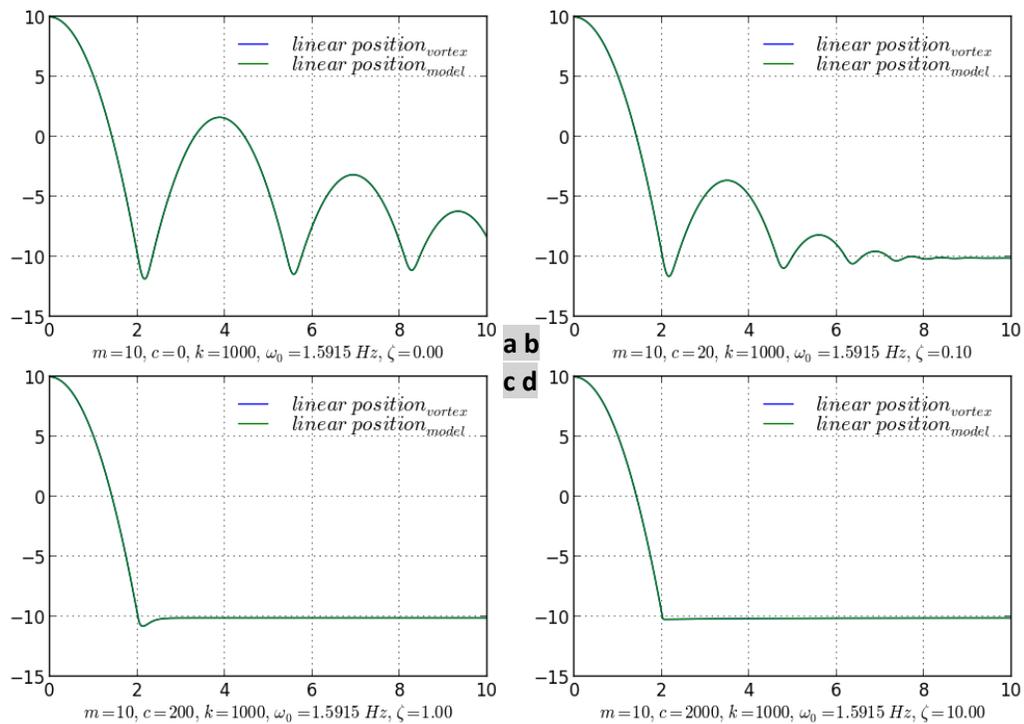


Figure 2-68: Validating VxPrismatic constraint: the behaviour of a mass around the limits

- a) undamped, $\zeta = 0$
- b) damped, $0 < \zeta < 1$
- c) critically damped, $\zeta = 1$
- d) over damped, $\zeta > 1$

Assumptions/notes

No air resistance is considered.

2.5.6 VxRPRO

Test Case

VxRPRO should constrain all DOFs.

VxRPRO (unless relaxed) constrains all DOFs. This is equivalent that two parts connected through *VxRPRO* are welded together. When a part is fully constrained to the world with an *VxRPRO* joint, the torque value read back from the constraint should be equal to the applied torque at the corresponding DoF at steady state.

Results

The force read back from the constraint equations are shown first in Figure 2-69. There is a perfect alignment between the applied quantity and the read back.

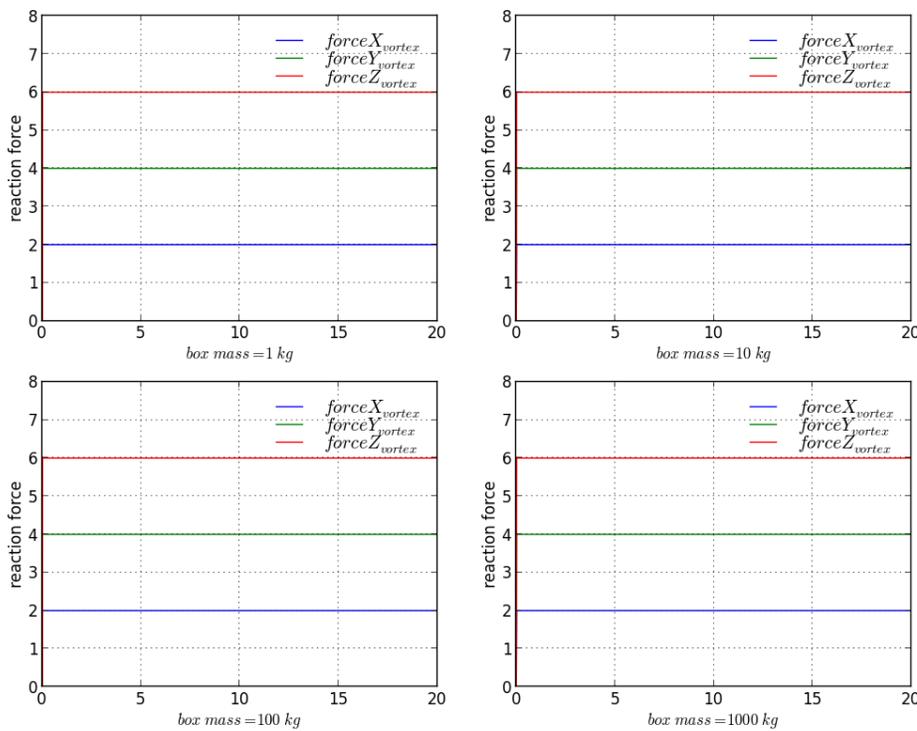


Figure 2-69: Validating VxRPRO constraint: force (2, 4, 6) is applied at a part constrained to the world, and the reaction from the constraint is read

The same is true for the torque as shown in Figure 2-70.

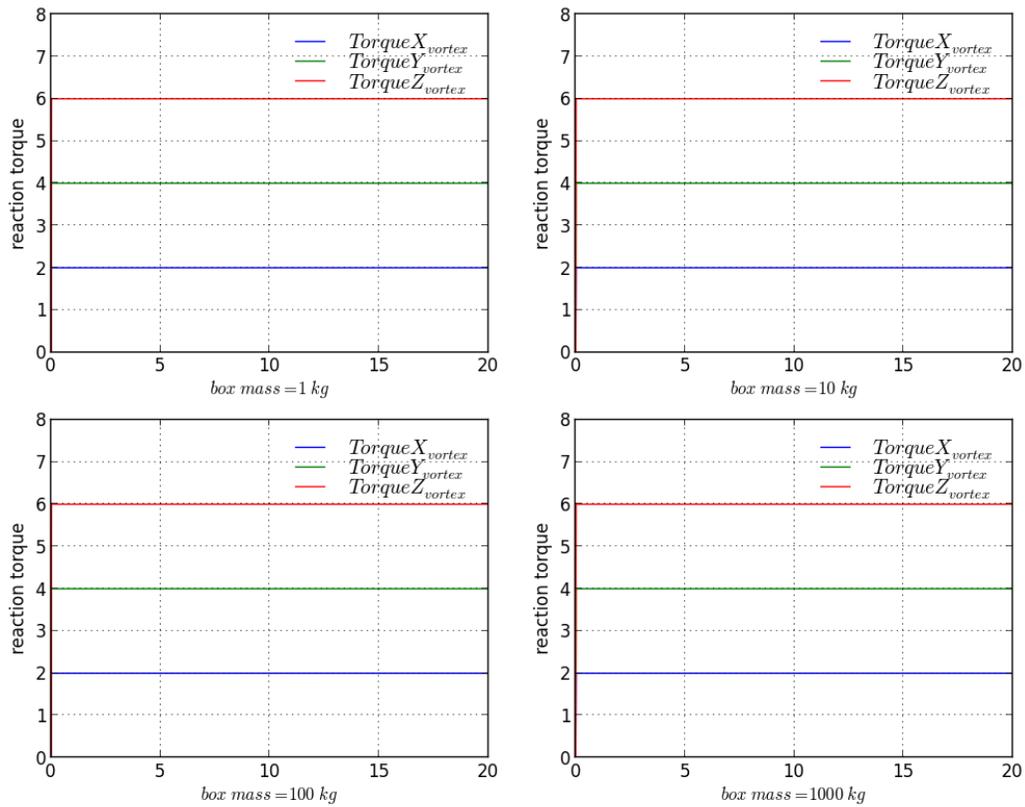


Figure 2-70: Validating VxRPRO constraint: torque (2, 4, 6) is applied at a part constrained to the world using RPRO, and the reaction from the constraint is read

Assumptions/notes

There is a frame delay because the force is computed using acceleration that is available only after a frame.

2.5.7 VxSpring

Test Case

The test should verify that a mass hung from a static pivot using *VxSpring* constraint should simulate a mass-spring-damping (MSD) system. The first attachment is used as a static pivot, and the other goes to the dynamic mass. By controlling the mass (m), stiffness (k), and damping (c) of the constraint, we should see different behaviours.

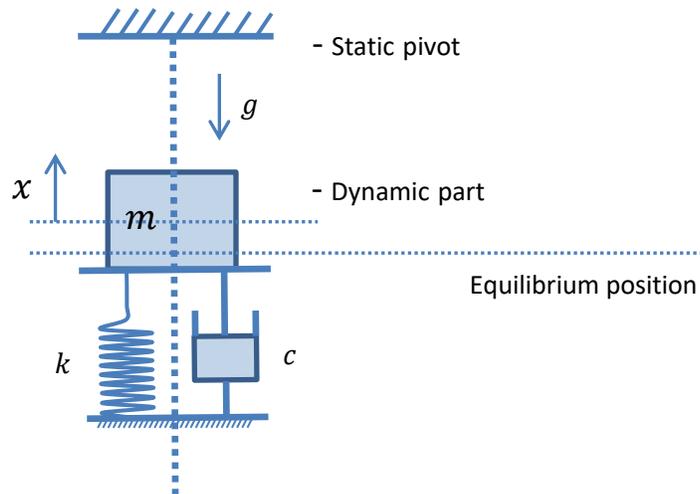


Figure 2-71: MSD system: a hanging mass under VxSpring constraint

From classical mechanics, the MSD system is governed by non-homogeneous second order differential equation (2.67), where m , c and k are mass of dynamic parts, damping constant, and stiffness of the limits respectively. Depending on the damping ratio ζ (2.30), the system shows different behaviours: undamped, damped, critically damped and over damped.

Results

Figure 2-72 shows that Vortex Studio responded closely with the model results for all damping ratios ζ except an additional small damping.

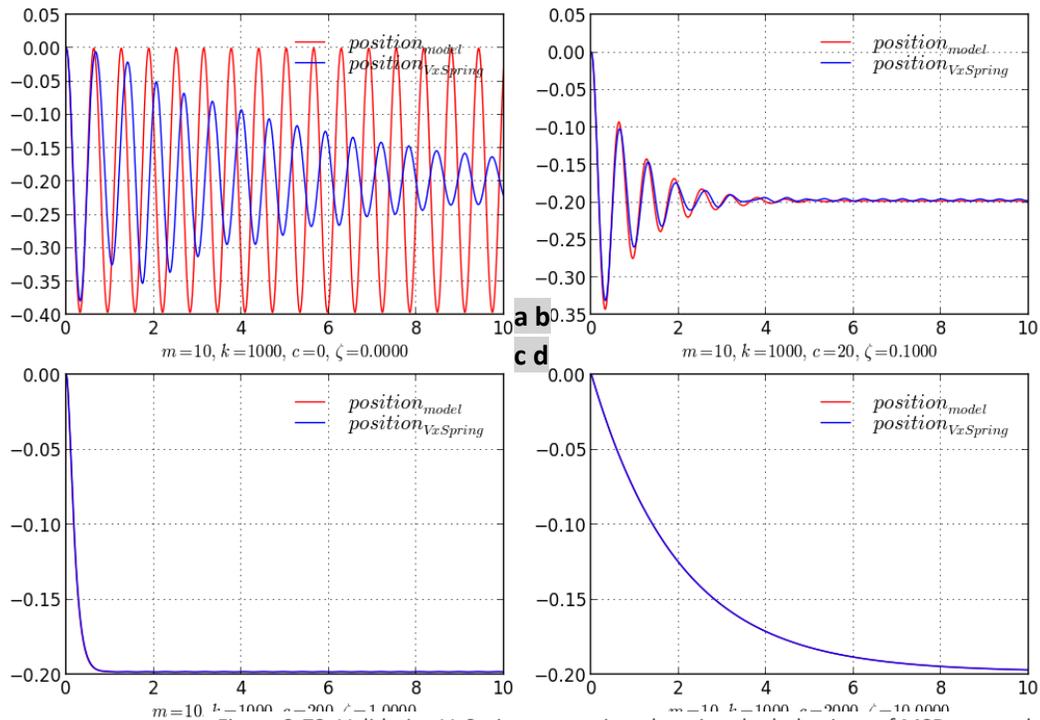


Figure 2-72: Validating VxSpring constraint: changing the behaviour of MSD system by controlling the damping ratio

- a) Undamped, $\zeta = 0$
- b) Damped, $0 < \zeta < 1$
- c) critically damped, $\zeta = 1$
- d) over damped, $\zeta > 1$

Assumptions/notes

To have a stable simulation, the implicit solver is chosen. However, this results in extra damping, which can be improved at the cost of a smaller step size.

2.5.8 VxUniversal

Test Case

When two parts are connected through *VxUniversal* constraint, the velocity coupling should be non-linear unless there is no axial shift between the axis.

Model

The velocity coupling is governed by the formula

$$\frac{\omega_0}{\omega_i} = \frac{\cos(\beta)}{1 - \sin^2\beta \cos^2\gamma_1} \quad (2.68)$$

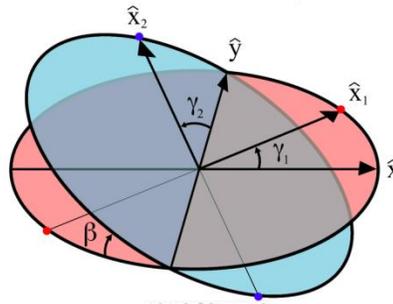


Figure 2-73: planes perpendicular to the input axis (in red) and to the output plane (in blue), and their orientations

- ω_i : Input velocity
- ω_0 : Output velocity
- β : Angle between planes
- $\gamma_{1,2}$: Angular displacements
 - Angles of unit vectors \hat{x}_1 and \hat{x}_2 w.r.t their initial positions along the \hat{x} and \hat{y} axes
 - These vectors are constrained to remain perpendicular.

Results

From Figure 2-74 a-d (four axial shift cases), it is evident that the higher the axial shift, the output velocity becomes more non-linear; however, Vortex Studio follows the model closely for all axial shifts.

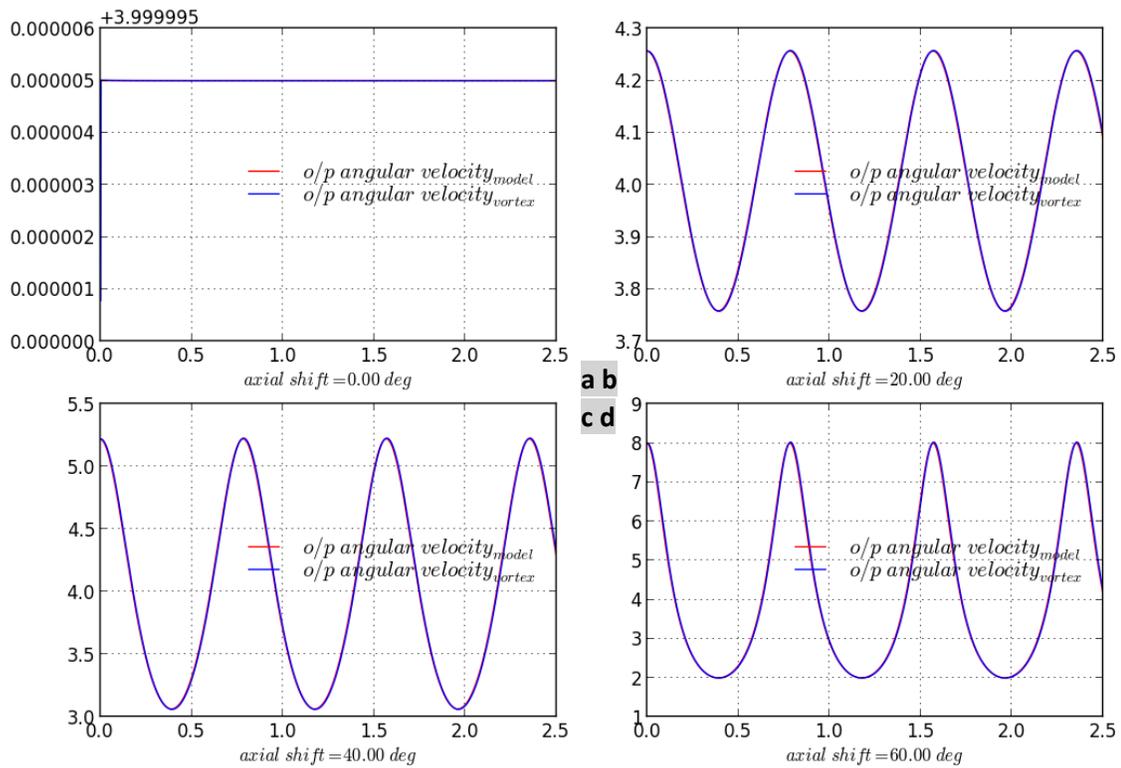


Figure 2-74: Validating VxUniversal constraint: effect on velocity coupling due to angular shift between shafts. Configuration: input velocity = 5.0 rad/sec

3 VALIDATION OF A VORTEX STUDIO-BASED SIMULATION

Validation answers the question of fitness to purpose. As Vortex Studio is a very flexible toolkit with wide-ranging functionality, validation is done on particular simulations for particular client requirements.

The first question to answer is what level of quality is desired? Anything on the simulated machine may be measured, but measuring the aggregate behavior of a real vehicle can be difficult. Therefore, the first step is to identify to what degree the simulation must model the actual vehicle and this will depend upon the purposes that the vehicle simulation will be put to: a training simulator may be more accurate than a research prototype, but doesn't need to be as accurate as a simulation used for design.

Vortex Studio can simulate the motions of articulated rigid bodies faithfully and in great detail but there are other major vehicle components that are much more complex and can only be handled using approximate models. Examples are soil interactions, track behavior, engines, transmissions and drive trains, and interactions with fluids. In many of these cases, the physics is too complex for real-time simulation, and in some cases (e.g., soil) the physics is not fully understood, so that empirical models must be used.

Once a level of quality has been clearly identified, it is necessary to ask how much time and money is available to take measures on the vehicle in question? Finally, there are other issues which will affect the level and confidence of the validation:

- Full vehicle measurements are not always available for reasons of government security, corporate trade secrets or access to the vehicle.
- It may be impossible to get the driver to do the sorts of actions required for reasons of difficulty or danger.
- Movement under gravity and potential forces.

3.1 Aggregate Behaviors

Aggregate behaviors are motions of the vehicle that emerge from the underlying properties of the vehicle. It is impossible, for example, to directly set braking distance of a vehicle in Vortex Studio. This emergent property is dependent on the mass distribution, the constraints between bodies, any controllers that are acting within the simulation, as well as coupling with environmental factors.

Using a tool like Vortex Studio requires starting with a clear idea of the aggregate behaviors that will be sought from the simulation. Early creation of unit tests that can determine whether these behaviors are achievable is essential; however it can be difficult to construct true/false unit tests for every desired behavior.

3.2 Data-based and Statistical Analysis

Frequently data sampled from a real vehicle under real conditions must be tested against data generated in a Vortex Studio simulation. The real data will be dirty and different each time the experiment is run, the Vortex Studio simulation is deterministic and saves exact data. Therefore the analysis will be statistical relative to the sampled data.

3.3 Subject Matter Expert Review

An important type of analysis is performed by a subject matter expert, usually someone who has experience with the simulated machine. These tests are usually done in a prototype of the final simulator and are concerned with correct behaviours of the vehicle, as well as achieving a good "feel," particularly with the integration to the controls and motion platform.

Subject Matter Expert reviews can be hard to do successfully. Frequently two SMEs will have conflicting analyses, and it is necessary for the development team to make judgment calls. This can be avoided by using a single SME throughout the testing.

3.4 Typical Tests during Validation

A few notes should be made on what the actual tests could be:

- Vehicle attitude change during acceleration, braking, turning, traversing obstacles.
- Engine rpm and vehicle speed during sustained acceleration and to show the behavior of automatic or manual transmissions.

A test matrix might include entries similar to Table 3-1:

Table 3-1: A typical test matrix

Test Number	Description	Velocity vs time	RPM vs time
T1a	vehicle at idle, set transmission to F1 at t=0		
T1b	vehicle at idle, set transmission to F1 at t=0, set throttle QUICKLY to 0.2, 0.4, 0.6, 0.8 wait until reach of max speed, then set throttle QUICKLY to 0		

CONTACT

For more information, CM Labs can be contacted using any of the following methods:

CM Labs Simulations - Head Office
645 Wellington, #301
Montreal, Quebec, H3C 1T2, Canada

Telephone: +1-514-287-1166

General inquiries: info@cm-labs.com

Sales: sales@cm-labs.com

Support (if under current subscription or M&S plan): support@cm-labs.com

Online discussion forum (free access): <http://support.cm-labs.com/support/discussions>

CM Labs' Contact page: <https://www.cm-labs.com/contact-us/>