Enhancements to the dynamics of a 2WD pickup truck pull

In an earlier paper, we looked at a simple model of a 2WD pickup truck pulling a loaded sled in a "tractor pull" competition. The simplified model was simple enough that we were able to derive a single equation for the distance travelled by the rig as a function of time. The single equation was useful because it provided some insight into the dynamics of a run and allowed us to quantify roughly what was going on.

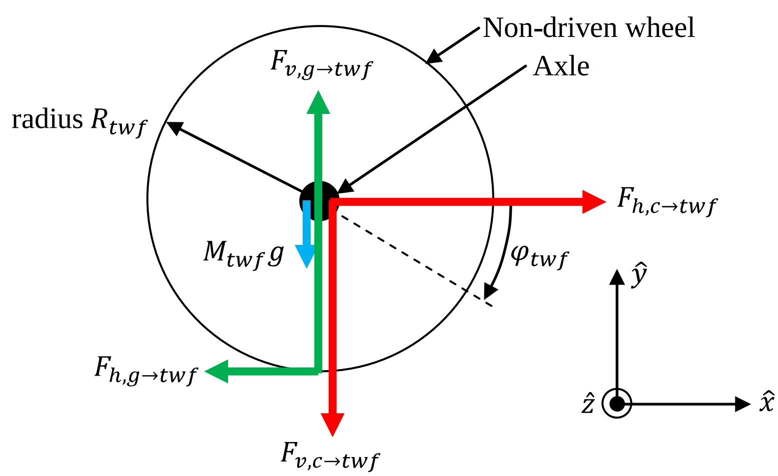
On the other hand, a single equation suffers from two big disadvantages. Firstly, it can only describe the activity in a single "mode" of operation. The most important mode in a tractor pull has the truck or tractor firmly on the ground. But, the single equation does not describe what happens when the rig begins to operate in some other mode, such as spinning its wheels. Secondly, keeping the model simple meant that we had to ignore certain physical realities. For example, we assumed that the effective height of the sled was the same as the effective height of the truck. As another example, we assumed that the entire rig was a rigid body; its wheels translated along with the rest of the rig, but did not turn.

In this paper, we will enhance the dynamics of simplified model to make it more realistic. We will break the rig into five separate rigid bodies, being the three sets of wheels, the truck's chassis and the sled's chassis. The cost of making the enhancements is that we will no longer be able to perform closed-form integration of the equations of motion, and will have to integrate numerically instead.

Part I -- Dynamics of the truck's front wheels, which are not driven wheels

In the earlier paper, we used Newton's Law, $Force = mass \times acceleration$, to calculate the acceleration of the rig along the horizontal. I described the variant of Newton's Law which applies to rotations, $Torque = Moment\ of\ inertia \times rotational\ acceleration$, but we did not use it in the equations of motion. We did not have to use it because we assumed that nothing would rotate. It is true that the source of power in the simplified model was a rotational torque applied to the rear wheels of the truck, from which we calculated the horizontal force applied by the tires on the ground. For that purpose, we used the definition of torque, $Torque = distance \times force$. But, we did not actually allow the rear wheels to turn, so we did not need to find their rotational acceleration.

Let's consider a typical pair of non-driven wheels. Both the front wheels of the pickup truck and the rear wheels of the sled fall into this category. They are not powered, and merely turn as required to adjust to the instantaneous position of their respective vehicles. The following figure shows a free body diagram for the front wheels of the truck. Variables which relate to these wheels have the subscript twf, which stands for "truck wheels, front".



The wheels are shown in a side view and their motion will be considered only in the plane of the page. Five forces act on the wheels. As is usual, the tail of each individual force is located at the point in the rigid body at which it acts. (It would be customary to scale the lengths of the arrows in proportion to the magnitudes of the forces, but we do not yet know the numerical values.) The five forces are these:

- M_{twf} is the mass of the pair of wheels, so that $M_{twf}g$ is the force of gravity exerted on the pair of wheels. Since the wheels are rotationally symmetric, we know that the center-of-gravity lies at the center of the axle. We will assume that the force of gravity can be concentrated at that point.
- $F_{v,c\rightarrow twf}$ is the vertical force which the rest of the truck (the"chassis") exerts on the front axle. I have assumed that the weight of the chassis causes it to bear down on the axle, so that this force will normally act in the downward direction. Note that the first subscript of the forces other than gravity is either v for vertical and h for horizontal.
- $F_{h,c\rightarrow twf}$ is the horizontal force which the rest of the truck exerts on the front axle. As in the earlier paper, the direction of the rig's motion is towards the right. If the vehicle is accelerating, it will push the front wheels forward in the direction of motion.
- $F_{v,g\rightarrow twf}$ is the vertical force which the ground exerts on the front wheels. This will normally be an upwards force, as shown.
- $F_{h,g\rightarrow twf}$ is the horizontal force which the ground exerts on the front wheels. This will normally be a retarding force, as shown, with a tendency to hold the bottom of the wheels stationary. We will not get into the details of the ground-to-rim interaction for the non-driven wheels. It will be sufficient for our purposes to assume that the non-driven wheels do not slip. The maximum frictional drag which the ground can exert on the rim of the tire is equal to the coefficient of static friction multiplied by the vertical force acting through the ground-to-rim boundary. We will assume that this maximum retarding force is always greater than the actual force $F_{h,g\rightarrow twf}$ required to hold the bottom of the wheels in place.

We will assume that the lines-of-action of all of these forces, other than $F_{h,g\rightarrow twf}$, pass through the center of the front axle. We will assume that $F_{h,g\rightarrow twf}$ acts perpendicularly to the radial vector from the center of the axle to the ground. In other words, everything is "squared up" with respect to the center of the axle and the point of contact with the ground which lies directly below it.

We will use the same $\hat{x}$-$\hat{y}$-$\hat{z}$ co-ordinate frame of reference that we used in the earlier paper. Its axes are shown to the lower right of the free body diagram. Note that I have defined rotation angle φ_{twf} in such a way that it increases as the wheels move forward. The sums of the forces in the $\hat{x}$ and $\hat{y}$-directions and the sum of the torques in the $\hat{z}$-direction, around the center of the front axle, are as follows:

$$
\begin{aligned}
\sum F_{x,twf} &= F_{h,c\rightarrow twf} - F_{h,g\rightarrow twf} && (1A)\\
\sum F_{y,twf} &= F_{v,g\rightarrow twf} - F_{v,c\rightarrow twf} - M_{twf}g && (1B)\\
\sum \tau_{z,twf} &= -F_{h,g\rightarrow twf}R_{twf} && (1C)
\end{aligned}
$$

In the earlier paper, we used the symbol φ for the angle by which an arbitrary wheel had rotated. For convenience, we will continue to use that symbol, with the appropriate subscript for the truck's front wheels. In the earlier paper, we used the symbol D for the horizontal distance an arbitrary wheel had travelled. We did not have a corresponding symbol for the vertical distance an arbitrary wheel had travelled, since we assumed that the rig remained in contact with the ground at all times. Let's use a new set of symbols for the distance travelled by the front wheels of the truck. Let x_{twf} and y_{twf} be the absolute co-ordinates of the front axle. We will measure y_{twf} upwards from the ground and x_{twf} from some spot related to the start of the run. Defined in this way, x_{twf} and y_{twf} cannot be said to be the "distances travelled" by the front wheels, but the distances travelled can easily be found from them by subtracting their starting values.

The horizontal acceleration of the wheels at any instant in time is the second derivative of the horizontal distance travelled. The vertical acceleration of the wheels at any instant in time is the second derivative of the vertical distance travelled. The rotational acceleration of the wheels at any instant in time is the second derivative of the angle by which the wheels have rotated. Note that, in Equation $(2C)$ following, the rotational acceleration around the $\hat{z}$-axis has a minus sign. This arises because I defined angle φ_{twf} in such a way that it increases as the wheel rotates around the negative $\hat{z}$-axis. We can state these relationships as:

$$
\begin{aligned}
LinearAcceleration_{x,twf} &= \frac{d^2 x_{twf}}{dt^2} && (2A)\\
LinearAcceleration_{y,twf} &= \frac{d^2 y_{twf}}{dt^2} && (2B)\\
RotationalAcceleration_{z,twf} &= -\frac{d^2 \varphi_{twf}}{dt^2} && (2C)
\end{aligned}
$$

To apply the linear variant of Newton's Law, the relevant physical parameter is the mass of the wheels, M_{twf}. The physical parameter needed to apply the rotational variant of Newton's Law is the moment of inertia. For the moment of inertia, we will use the symbol I_{twf}. I will explain presently how to calculate this quantity. One can think of the mass of the wheels as their resistance to moving in response to an applied force. In a like way, the moment of inertia of the wheels is their resistance to rotating in response to an applied torque. Newton's Law applied to the front wheels of the truck can be written as:

$$
\begin{aligned}
\sum F_{x,twf} &= M_{twf}\frac{d^2 x_{twf}}{dt^2} &&= F_{h,c\to twf} - F_{h,g\to twf} && (3A)\\
\sum F_{y,twf} &= M_{twf}\frac{d^2 y_{twf}}{dt^2} &&= F_{v,g\to twf} - F_{v,c\to twf} - M_{twf}g && (3B)\\
\sum \tau_{z,twf} &= -I_{twf}\frac{d^2 \varphi_{twf}}{dt^2} &&= -F_{h,g\to twf}R_{twf} && (3C)
\end{aligned}
$$

This is the set of three equations of motion which apply to the front wheels. We cannot solve them in isolation. Each of the five forces is associated with an equal and opposite force which is applied to some other body. The reaction to the force of gravity $M_{twf}g$, for example, is the upwards pull which is experienced by the center of the Earth. We don't really care how the Earth responds to the pull of the wheels. But we do care about the equal and opposite forces associated with two of these five forces, $F_{h,c\to twf}$ and $F_{v,c\to twf}$. The reaction forces are experienced by the truck's chassis, which is an integral part of the rig. There is said to be a "coupling" between the equations of motion for the front wheels and the equations of motion for the truck's chassis. Both equations of motion will need to be solved simultaneously. In fact, we are going to divide the rig up into five separate rigid bodies, each with its own three equations of motion. There are couplings between various pairs of these five bodies, with the result that all 15 equations of motion will need to be solved simultaneously

The three equations of motion involve seven unknowns: accelerations in the three variables x_{twf}, y_{twf} and φ_{twf}, a pair of ground forces and a pair of chassis forces. Even if we expect to get two more pieces of information, in the form of two more equations, when we solve the equations of motion for the chassis, we will still have an under-constrained set of equations. There will still be two more unknowns than equations. Resolving the shortfall is most easily done if we consider certain physical states, or cases, in which the front wheels can be in. There are three distinct states.

Case Front #1 -- The front wheels are in the air

If the front wheels do leave the ground, then the ground can no longer exert any vertical or horizontal forces on the wheel. I will call this condition "airborne". Mathematically, we can describe this state as:

$$\left.\begin{aligned} F_{h,g\to twf} &= 0 \\ F_{v,g\to twf} &= 0 \end{aligned}\right\} \text{ whenever } y_{twf} > R_{twf} \quad (4)$$

Note the inequality in $y_{twf} > R_{twf}$ which describes the front wheels being above the ground. Of course, when $y_{twf} = R_{twf}$, the wheels are on the ground. A less-than inequality is not physically possible since it would correspond to the wheels somehow sinking into the ground.

When the wheels are airborne, the equations of motion reduce to:

$$\begin{aligned} M_{twf}\frac{d^2x_{twf}}{dt^2} &= F_{h,c\to twf} && (5A) \\ M_{twf}\frac{d^2y_{twf}}{dt^2} &= -F_{v,c\to twf} - M_{twf}g && (5B) \\ \frac{d^2\varphi_{twf}}{dt^2} &= 0 && (5C) \end{aligned}$$

There are only five unknowns now and a solution should be possible. The third equation is a second-order differential equation in the single variable φ_{twf} which sets the rotational acceleration of the front wheels to zero. The rotational speed is the first integral of the rotational acceleration, so the rotational speed of the front wheels will be constant. In fact, this constant rotational speed will be the rotational speed at which the wheels were turning at the instant they left the ground. Mathematically, the wheels will keep spinning forever at this speed or, at least until the wheels come back into contact with the ground. In reality, the wheels will not keep spinning in the air forever. There is some friction in the axle that will absorb energy from the spin and slow the rotation. We are going to ignore this little bit of rotational drag.

Case Front #2 -- The front wheels are on the ground

When the front wheels are in contact with the ground, then the rims of the wheels do not slip with respect to the ground. As we saw in the earlier paper, this requires that the rate at which the circumference of the wheels comes into contact with the ground must be equal to the rate at which the wheels are moving forward. This relationship applies not just to the horizontal speed, but to the horizontal acceleration as well. Mathematically, we can express the accelerations' relationship in this state as:

$$\frac{d^2x_{twf}}{dt^2} = R_{twf}\frac{d^2\varphi_{twf}}{dt^2} \text{ whenever } y_{twf} = R_{twf} \quad (6)$$

We can notionally add this equation to the three which constitute the equations of motion. Equation (6) is a kinematic relationship, or constraint, which applies whenever the front wheels are on the ground.

Even adding this extra equation is not enough to give a unique solution. There is still one more unknown than there are equations. Before I describe how we can resolve this shortfall, let me show what happens when we substitute Equation (6) into the equations of motion. We can re-arrange Equation $(3A)$ to isolate the horizontal ground force $F_{h,g\to twf}$. After substituting for it, the first two equations of motion reduce to:

$$F_{h,c\to twf} = \left(M_{twf} + \frac{I_{twf}}{R_{twf}^2}\right)\frac{d^2x_{twf}}{dt^2} \quad (7A)$$

$$F_{v,g\to twf} - F_{v,c\to twf} = M_{twf}\left(\frac{d^2y_{twf}}{dt^2} + g\right) \quad (7B)$$

Notice that the first equation looks a lot like Newton's Law, $Force = Mass \times Acceleration$. In fact, it is exactly in the form of Newton's Law if we treat the factor $M_{twf} + I_{twf}/R_{twf}^2$ as the "effective mass" of the wheels. In other words, for the purposes of calculating their linear acceleration, the rotation of the front wheels can be treated like, and has the same effect as, an increase in their mass.

Notice that the second equation also looks a lot like Newton's Law. In fact, it is exactly in the form of Newton's Law if we interpret the factor $d^2y_{twf}/dt^2 + g$ as the "effective vertical acceleration" of the wheels. The term is the sum of the gravitational force and the reaction force to the acceleration of the wheels as they travel upwards.

Case Front #2A -- The front wheels are on the ground, and stay on the ground

It might be tempting to say: if the front wheels are on the ground, then the sum of the vertical forces must be zero. That is not necessarily so. It is true that, if the sum of the vertical forces is zero, then the front wheels will say of the ground. In fact, that is exactly what we would say if we were looking for a "static equilibrium" for the wheels. But, what if the front wheels are just on the verge of going airborne. It's best if we look at two sub-cases of the Front #2 case.

If the front wheels are on the ground, and are expected to stay on the ground, then the net vertical force must be zero. An equivalent description of this situation, in terms of acceleration, is that the vertical acceleration is zero. We can write down this situation mathematically as:

$$\left.\begin{array}{c} F_{v,g\to twf} - F_{v,c\to twf} - M_{twf}g = 0 \\ \text{or} \\ \dfrac{d^2y_{twf}}{dt^2} = 0 \end{array}\right\} \text{whenever } y_{twf} = R_{twf} \text{ and remains so} \quad (8)$$

One can look at this as either adding another equation or as eliminating one of the unknowns. Either way, the number of equations is brought into balance with the number of unknowns, making a solution possible.

Case Front #2B -- The front wheels are on the ground, but are on the verge of going airborne

Two things must happen at the same instant when the front wheels make a transition from being on the ground to being airborne. The vertical force of the ground on the wheels $F_{v,g\to twf}$ must be zero and the vertical acceleration d^2y_{twf}/dt^2 must be non-zero. Mathematically, we cannot say that one causes the other or vice versa. They happen together. Knowing that $F_{v,g\to twf} = 0$ has the effect of reducing the number of unknowns by one, once again making the set of equations soluble.

There is a certain symmetry between the Front #2A case and the Front #2B case. We set an acceleration equal to zero in the former; we set a force equal to zero in the latter.

It will not be very convenient if we have to set up and solve different sets of equations for different cases. As we will see, there are also different cases for the truck's rear wheels and for the sled's motion. It will be more convenient to have a single set of equations which can handle all of the various cases. One way to accomplish that is to use binary flags which cause certain terms or even equations to be included or excluded from the set. I propose to use two binary flags for the truck's front wheels. I will define them as follows:

State of truck's front wheels	B_{twfAir}	B_{twfRTO}
Case Front #1: Wheels are airborne	1	0
Case Front #2A: On ground; remain on ground	0	0
Case Front #2A: On ground; ready for take-off	0	1

The subscript of B_{twfAir} is intended to be descriptive. When the "truck's front wheels are airborne", it is equal to one. When they are not airborne, it is equal to zero. The flag B_{twfRTO} does not mean anything when the wheels are in the air but, when they are on the ground, it is equal to one when the "truck's front wheels are ready for take-off". Using these flags, we can combine everything we know about the state of the truck's front wheels into the following set of equations:

$$(3A)\qquad M_{twf}\frac{d^2x_{twf}}{dt^2}=F_{h,c\to twf}-F_{h,g\to twf}\qquad(7A)$$

$$(3B)\qquad M_{twf}\frac{d^2y_{twf}}{dt^2}=F_{v,g\to twf}-F_{v,c\to twf}-M_{twf}g\qquad(7B)$$

$$(3C)\qquad -I_{twf}\frac{d^2\varphi_{twf}}{dt^2}=-F_{h,g\to twf}R_{twf}\qquad(7C)$$

$$\text{Case Front \#1}\qquad B_{twfAir}F_{h,g\to twf}=0\qquad(7D)$$

$$\text{Case Front \#1}\qquad B_{twfAir}F_{v,g\to twf}=0\qquad(7E)$$

$$\text{Case Front \#2}\qquad (1-B_{twfAir})\left[\frac{d^2x_{twf}}{dt^2}-R_{twf}\frac{d^2\varphi_{twf}}{dt^2}\right]=0\qquad(7F)$$

$$\text{Case Front \#2A}\qquad (1-B_{twfAir})(1-B_{twfRTO})\frac{d^2y_{twf}}{dt^2}=0\qquad(7G)$$

$$\text{Case Front \#2B}\qquad (1-B_{twfAir})B_{twfRTO}F_{v,g\to twf}=0\qquad(7H)$$

It makes sense to combine Equations $(7E)$ and $(7H)$. The vertical force of the ground on the front wheels is identically equal to zero in two of the cases. We we combine them as follows:

$$\left[B_{twfAir}+(1-B_{twfAir})B_{twfTO}\right]F_{v,g\to twf}=0\qquad(8)$$

Having combined two of the equations leaves seven equations. They involve seven unknowns. But, they are not soluble. The reader can confirm that, for any combination of the binary flags, two of the equations be become trivial identities $0=0$. So, we will always have exactly five non-trivial equations. That's good, because the front wheels interact with the truck's chassis and the interaction will be the source of two more equations, which involve the two chassis forces.

Part II -- Dynamics of the sled's rear wheels, which are also non-driven wheels

Like the front wheels of the truck, the rear wheels of the sled are not driven. If we use the subscript swr, which stands for "sled wheels, rear", then we can take over all of the results from Part I. In fact, the rear wheels of the sled are even simpler. We do not expect them to rise off the ground, so there are not multiple cases. In other words, $y_{swr} = R_{swr}$ at all times. Not only that, but the net vertical force is always zero. These wheels are in a static vertical equilibrium. The equations of motion of the rear wheels of the sled can be written down by inspection from Equation (3) as follows:

$$\sum F_{x,swr} = M_{swr}\frac{d^2 x_{swr}}{dt^2} = F_{h,c\to swr} - F_{h,g\to swr} \qquad (9A)$$

$$\sum F_{y,swr} = 0 = F_{v,g\to swr} - F_{v,c\to swr} - M_{swr}g \qquad (9B)$$

$$\sum \tau_{z,swr} = -I_{swr}\frac{d^2 \varphi_{swr}}{dt^2} = -F_{h,g\to swr}R_{swr} \qquad (9C)$$

Since these wheels remain in contact with the ground, and do not slip, the following circumference versus distance relationship holds true as well:

$$\frac{d^2 x_{swr}}{dt^2} = R_{swr}\frac{d^2 \varphi_{swr}}{dt^2} \qquad (10)$$

We can combine Equations (9) and (10) into a single set of equations, as follows:

$$M_{swr}\frac{d^2 x_{swr}}{dt^2} = F_{h,c\to swr} - F_{h,g\to swr} \qquad (11A)$$

$$F_{v,g\to swr} - F_{v,c\to swr} - M_{swr}g = 0 \qquad (11B)$$

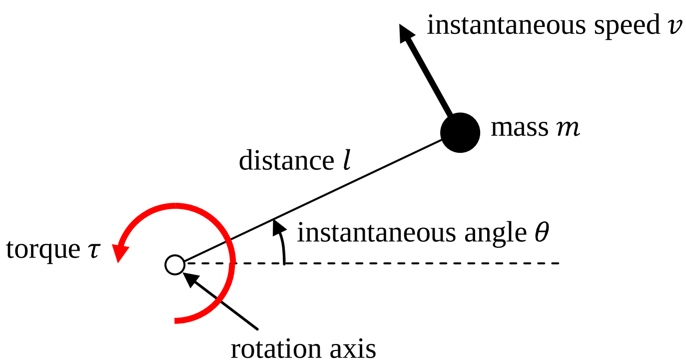
$$I_{swr}\frac{d^2 \varphi_{swr}}{dt^2} = F_{h,g\to swr}R_{swr} \qquad (11C)$$

$$\frac{d^2 x_{swr}}{dt^2} = R_{swr}\frac{d^2 \varphi_{swr}}{dt^2} \qquad (11D)$$

There are six unknowns: accelerations in the two variables x_{swr} and φ_{swr}, a pair of ground forces and a pair of chassis forces. A solution is possible. These four equations will be augmented by two more when the interaction between the sled's chassis and its rear wheels is taken into account.

Part III -- Calculating the moment of inertia of a wheel

As I have said, a moment of inertia is an object's resistance to a rotational force, which one calls torque. Let's consider the simplest case first, of a point mass m on the end of a massless stick with length l. We will try to rotate the mass around the other end of the stick by applying torque τ. (These are generic symbols and do not relate specifically to the truck, the sled or their wheels.) The following figure shows the setup.



Since the stick is fixed at one end, the mass revolves around the fixed end in a circle with radius l. At any instant in time, the mass is moving in a direction which is tangent to its orbit, which is to say, in the direction which is perpendicular to the stick. In the figure, I have shown an angle θ, which is the Greek letter "theta", which is measured from a starting position defined, arbitrarily, as the dotted horizontal line. As we have seen before, the distance travelled around the circumference of a circle is related to the angle subtended by the arc, and the constant of proportionality is the radius. So, the speed v of the mass is related to the rate of change of the subtended angle $d\theta/dt$ by:

$$v = l\frac{d\theta}{dt} \quad (12)$$

The direction of the speed changes as the mass revolves, but the instantaneous speed is always given by Equation (12). If the mass happens to be revolving around the rotation axis at speed v, and no torque is applied, then the mass will continue to orbit at this constant speed. It will nevertheless be accelerating, inwards, along the axis of the stick. This is the centripetal acceleration which counteracts the so-called centrifugal force. Circular motion is a special case where the acceleration is such that it changes the direction of travel of an object without changing its speed. When we apply a non-zero torque, we will cause changes in the speed of travel as well.

Let's consider a very short interval of time during which the object moves a very short distance along the circumference of its circular orbit. We can easily imagine such a short interval and such a short distance that the object travels, for all intents and purposes, along a straight line segment. If torque τ is being applied to the other end of the stick during this interval, then the point mass will experience a force along its direction of travel during the interval equal to the torque divided by the lever-arm l. If we let a be the mass's linear acceleration along the straight line during this interval, then Newton's Law for linear motion can be written as:

$$\frac{\tau}{l} = ma \quad (13)$$

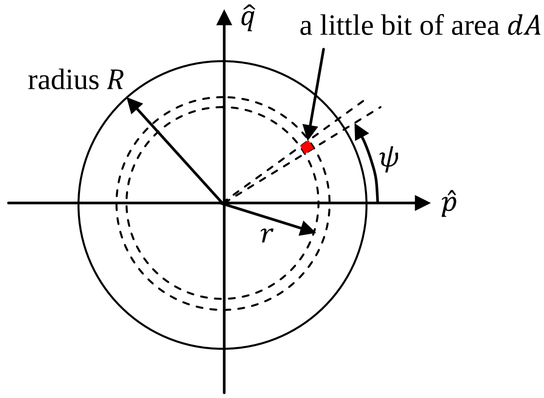
Since the acceleration a is the rate of change of the speed v, we can combine Equations (12) and (13) as:

$$\begin{aligned} \frac{\tau}{l} &= m\frac{dv}{dt} \\ &= ml\frac{d^2\theta}{dt^2} \\ \rightarrow \quad \tau &= ml^2\frac{d^2\theta}{dt^2} \quad (14) \end{aligned}$$

This has exactly the form of the variant of Newton's Law for rotational acceleration, $Torque = Moment\ of\ inertia \times Angular\ acceleration$, where the moment of inertia is ml^2. For a point mass, the moment of inertia is the mass multiplied by the square of the radius of the orbit. A point mass twice as heavy resists a torque by twice as much. A point mass twice as far away resists a torque by four times as much.

The units of a moment of inertia are mass multiplied by distance squared. In English units, we would say pound-inch-squared or pound-foot-squared. In S.I. units, we would say kilogram-meter-squared.

The concept and derivation of the moment of inertia of a point mass can be extended to any rigid object, including a three dimensional one. The trick is to divide the rigid object up into a large number of tiny bits, and to deal with each little bit as if it was a point mass. Consider the homogeneous disk shown in the following figure. The disk does not have to be thin; it can have thickness. The moment of inertia of the little bits depends only on their radial distance from the axis of rotation. It does not depend on their displacement along the axle, so we can think of a thick disk as being the sum of many thin disks glued together face-to-face.



If the disk has an outer radius R and a total mass M, then its mass per unit area of face is equal to $M/\pi R^2$. We will divide the face up into little bits in two steps. Firstly, we will divide the face up into annuli. An annulus is a circle with a circular hole cut out of the middle, just like a washer. One such annulus is shown in the figure. It has an inside radius r and an outside radius that is just a tiny bit dr bigger. The outside radius of the annulus is $r + dr$. Both the inside and outside circumferences of the annulus are shown as dotted circles in the figure. We will divide up the entire disk into a succession of such annuli, where the outer radius of each annulus is equal to the inner radius of the next bigger annulus, and so on until we reach the rim of the disk.

Secondly, each annulus is divided up by chopping the 360° around the circle into small angles. I have shown an angle ψ, which is the Greek letter "psi", which is defined by a radial dotted line. A nearby radial dotted line is positioned at angle $\psi + d\psi$, where $d\psi$ is a very small angle.

The intersection of the annulus and the small change in angle $d\psi$ is a small rectangular-shaped area. It is not a true rectangle. Two of its sides are arcs of circles and so are slightly curved. Its other two sides are radii of the same circle and so are not exactly parallel. However, as we imagine that dr and $d\psi$ become smaller and smaller, the small bit of area becomes more and more rectangular. In the limit as the two differentials become infinitely small, the area becomes exactly rectangular. Like any rectangle, its area is the product of its base times its height, $dA = rd\psi \times dr$. If we multiply this area by the mass of the disk per unit area, we get the mass of the disk enclosed by this rectangle: $dm = Mrdrd\psi/\pi R^2$. It should be apparant that I have used the symbol dm for the mass of this little rectangle.

We can now apply the formula for the moment of inertia of a point mass, $M.O.I. = mass \times radius^2$. We have just calculated the mass dm and, by definition, the radial distance of the little rectangle is r. If we let dI be the moment of inertia of the little rectangle, then:

$$dI = \frac{Mr^3 drd\psi}{\pi R^2} \quad (15)$$

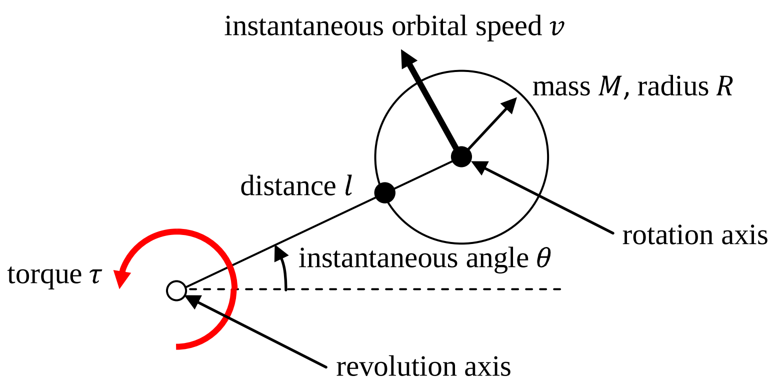
This formula applies to each little rectangle into which we can divide the disk. "Adding up" the dI contibutions to the total moment of inertia I of the disk as a whole is accomplished by integrating Equation (15) over the physical extemes of the disk. The integration is straight forward:

$$
\begin{aligned}
I &= \int_{r=0}^{r=R} \int_{\psi=0}^{\psi=2\pi} \frac{Mr^3}{\pi R^2} dr d\psi \\
&= \frac{M}{\pi R^2} \int_{r=0}^{r=R} r^3 \left(\int_{\psi=0}^{\psi=2\pi} d\psi \right) dr \\
&= \frac{2M}{R^2} \int_{r=0}^{r=R} r^3 dr \\
&= \frac{2M}{R^2} \times \tfrac{1}{4} R^4 \\
&= \tfrac{1}{2} M R^2 \qquad (16)
\end{aligned}
$$

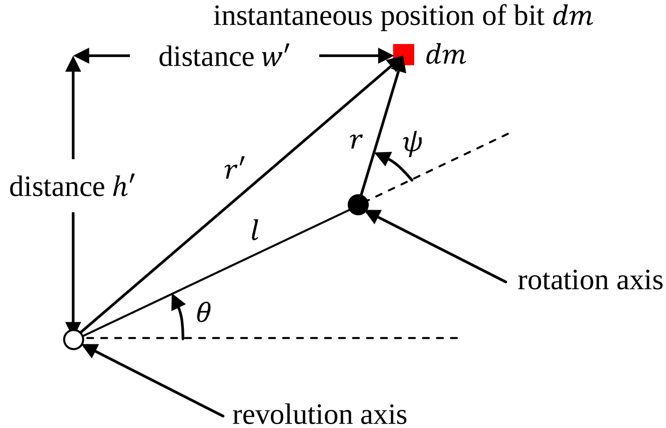
The moment of inertia of a uniform disk is quite similar to that of a point mass. The difference is the factor one-half. As expected, the units are the same, too, being mass times distance squared.

As a quick numerical example, suppose we have a uniform disk which has a mass of 100 pounds (45.4 kilograms) and a diameter of 30 inches (radius = 0.381 meters). Its moment of inertia is $45.4 \times 0.381^2 = 6.59$ kg-m^2. This is roughly equivalent to 17 pounds on the end of a yard-stick. A real wheel, of course, is not a uniform disk. It likely has a metal rim and rubber tires and, in any event, has various amounts of these materials at various radii from the center. Calculating the moment of inertia of a real wheel can be done by adding up the little bits, as we have done, but requires a complete specification of the mass as a function of radial distance.

Let's take the concept to the next stage. What if a rigid body is subjected to a combination of rotation and revolution? There is a difference between rotation and revolution. The Earth, for example, rotates once per day but revolves once per year. Rotation takes place around a body's center of mass while revolution is the rotation of the center of mass around some other point. To explore this concept, let's consider a uniform disk with mass M and radius R bolted to the end of a stick with length l. This is shown in the following figure.



The two black dots are the bolts that secure the disk to the stick. It must therefore be the case that the angle by which the disk rotates around the rotation axis is identical to the angle by which the disk revolves around the revolution axis. To calculate the moment of inertia of this body, we will follow the procedure we did above. We will divide the disk up into annuli and radial sectors. A typical small rectangle will again be located by radius r, measured from the rotation axis, and by angle ψ, measured counter-clockwise from the horizontal. But, to calculate the moment of inertia <u>around the revolution axis</u>, we need to convert the radius and angle by which the small rectangle is located into terms referenced to the revolution axis. We will use the trigonometry shown in the following figure to do the conversion.



The instantaneous location of the rectangle dm can be specified using the Cartesian co-ordinates h' and w', which are computed using the following trigonometry:

$$\left.\begin{aligned} h' &= r\sin(\psi+\theta) + l\sin\theta \\ w' &= r\cos(\psi+\theta) + l\cos\theta \end{aligned}\right\} \quad (17)$$

The radius r' from the revolution axis to the rectangle can be found by applying the Pythagorean Theorem:

$$\begin{aligned} r' &= \sqrt{h'^2 + w'^2} \\ &= \sqrt{[r\sin(\psi+\theta) + l\sin\theta]^2 + [r\cos(\psi+\theta) + l\cos\theta]^2} \\ &= \sqrt{\begin{matrix} r^2\sin^2(\psi+\theta) + 2rl\sin(\psi+\theta)\sin\theta + l^2\sin^2\theta + \cdots \\ \cdots + r^2\cos^2(\psi+\theta) + 2rl\cos(\psi+\theta)\cos\theta + l^2\cos^2\theta \end{matrix}} \\ &= \sqrt{\begin{matrix} r^2\sin^2(\psi+\theta) + r^2\cos^2(\psi+\theta) + \cdots \\ \cdots + 2rl\sin(\psi+\theta)\sin\theta + 2rl\cos(\psi+\theta)\cos\theta \\ \cdots + l^2\sin^2\theta + l^2\cos^2\theta \end{matrix}} \\ &= \sqrt{\begin{matrix} r^2 + \cdots \\ \cdots + 2rl[\sin(\psi+\theta)\sin\theta + \cos(\psi+\theta)\cos\theta] \\ \cdots + l^2 \end{matrix}} \\ &= \sqrt{r^2 + 2rl\cos(\psi+\theta-\theta) + l^2} \\ &= \sqrt{r^2 + 2rl\cos\psi + l^2} \end{aligned} \quad (18)$$

The mass of the small rectangle is the same as it was before, $dm = Mrdrd\psi/\pi R^2$. Now, however, the expression for the moment of inertia of the point mass dm must depend on its radius from the revolution axis, as follows:

$$\begin{aligned} dI &= dm \times {r'}^2 \\ &= \frac{Mrdrd\psi}{\pi R^2} \times (r^2 + 2rl\cos\psi + l^2) \end{aligned} \quad (19)$$

To find the total moment of inertia, we integrate over the extremes of the disk. The integration is a little bit more involved than it was before, since it involves a trigonometric term. One proceeds as follows, integrating over the angle ψ first:

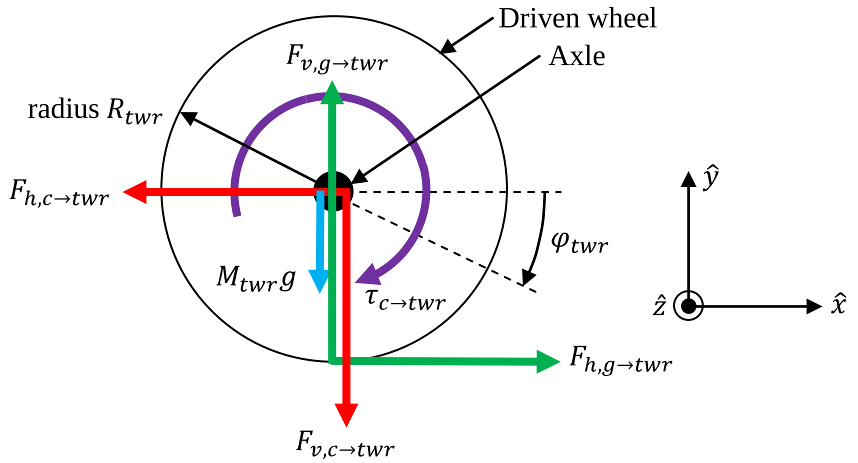
$$
\begin{aligned}
I &= \int_{r=0}^{r=R} \int_{\psi=0}^{\psi=2\pi} dI \\
&= \frac{M}{\pi R^2} \int_{r=0}^{r=R} \left[\int_{\psi=0}^{\psi=2\pi} r(r^2 + 2rl\cos\psi + l^2) d\psi \right] dr \\
&= \frac{M}{\pi R^2} \int_{r=0}^{r=R} \left\{ \left[r(r^2 + l^2) \int_{\psi=0}^{\psi=2\pi} d\psi \right] + \left[2r^2 l \int_{\psi=0}^{\psi=2\pi} \cos\psi \, d\psi \right] \right\} dr \\
&= \frac{M}{\pi R^2} \int_{r=0}^{r=R} [2\pi r(r^2 + l^2)] \, dr \\
&= \frac{2M}{R^2} \left[\left(\int_{r=0}^{r=R} r^3 \, dr \right) + \left(l^2 \int_{r=0}^{r=R} r \, dr \right) \right] \\
&= \frac{2M}{R^2} \left[\tfrac{1}{4} R^4 + l^2 \left(\tfrac{1}{2} R^2 \right) \right] \\
&= \tfrac{1}{2} MR^2 + Ml^2
\end{aligned}
\tag{20}
$$

This is a very important result. It says that the total moment of inertia is the sum of the M.O.I. of the disk around its central axis (the $\frac{1}{2}MR^2$ term) plus the M.O.I. of the disk's total mass treated as if it is a point mass revolving around the revolution axis (the Ml^2 term).

<u>Part IV -- Dynamics of the rear wheels of the truck, which are driven wheels</u>

The equations which describe the dynamics of a driven wheel are not much more complicated than those of a non-driven wheel. Driven wheels are more complex, or course, but the extra complexity is required to calculate the applied forces, not in the equations of motion themselves.

As a starting point, let's consider the rear wheels of the truck, which are a driven pair of wheels. The figure shown here is a free body diagram for these wheels. The subscript *twr* stands for "truck wheels, rear".



There are only three differences compared with a pair of non-driven wheels, two of which are merely changes in the directions in which the horizontal forces are expected to act. The differences are these:

- The chassis, through the powertrain, applies a clockwise torque $\tau_{c\to twr}$ to the rear wheels. Of course, the wheels exert an equal and opposite torque on the truck's chassis.
- The horizontal force of the chassis on the axle is a retarding force $F_{h,c\to twr}$ on the driven wheels. The wheels push the chassis ahead, and the reaction from the chassis is a force holding the wheels back.
- Similarly, the expected direction of the horizontal force of the ground on the wheels $F_{h,g\to twr}$ is reversed. The rims of the tires push backwards on the ground and the ground responds by pushing forward on the rims.

The sums of the forces in the $\hat{x}$ and $\hat{y}$-directions and the sum of the torques in the $\hat{z}$-direction, around the center of the axle, are as follows:

$$\begin{aligned}
\sum F_{x,twr} &= F_{h,g\to twr} - F_{h,c\to twr} && (21A)\\
\sum F_{y,twr} &= F_{v,g\to twr} - F_{v,c\to twr} - M_{twr}g && (21B)\\
\sum \tau_{z,twr} &= F_{h,g\to twr}R_{twr} - \tau_{c\to twr} && (21C)
\end{aligned}$$

The distances x_{twr} and y_{twr} and the angle φ_{twr} can be defined for this driven wheel in the same way as they were defined for the non-driven wheels above. Therefore, the expressions for the linear and rotational accelerations in Equation (2) apply to driven wheels, as well.

We will use M_{twr} for the mass of the truck's rear wheels and I_{twr} for their moment of inertia. Newton's Law applied to the wheels can then be written as:

$$\begin{aligned}
\sum F_{x,twr} &= M_{twr}\frac{d^2x_{twr}}{dt^2} &&= F_{h,g\to twr} - F_{h,c\to twr} && (22A)\\
\sum F_{y,twr} &= M_{twr}\frac{d^2y_{twr}}{dt^2} &&= F_{v,g\to twr} - F_{v,c\to twr} - M_{twr}g && (22B)\\
\sum \tau_{z,twr} &= -I_{twr}\frac{d^2\varphi_{twr}}{dt^2} &&= F_{h,g\to twr}R_{twr} - \tau_{c\to twr} && (22C)
\end{aligned}$$

Like we did for the non-driven wheels, we can specify some constraints. The vertical distance y_{twr} will always be equal to the wheels' radius R_{twr}. The drive wheels may slip, but they will never rise off the ground. Since $y_{twr} = R_{twr}$ at all times, both the vertical speed dy_{twr}/dt and the vertical acceleration d^2y_{twr}/dt^2 must be identically zero. So will the net vertical force. We can re-write Equation $(22B)$ as follows:

$$F_{v,g\to twr} - F_{v,c\to twr} - M_{twr}g = 0 \text{ at all times } t \qquad (23)$$

Although the truck's drive wheels are always in contact with the ground, they may or may not be slipping. The proportionality between the wheels rotational speed and translational speed cannot be taken for granted. In any event, it looks like we will have to consider two cases for the rear wheels as well. An important difference between the cases arises from the nature of the ground friction.

.

Part V -- The ground reaction forces on the rear wheels of the truck

I am going to refer to both coefficients of friction, the coefficient of static friction $C_{twr,s}$ which applies when the rear wheels do not slip and the coefficient of dynamic friction $C_{twr,d}$ which applies when they do. These coefficients have the same definitions as the coefficients of friction I described in the earlier paper.

Case Rear #1 -- The rear wheels are not slipping

Let's consider first the case when the rear wheels do not slip. When the wheels are not slipping, we will have the same circumferential-speed versus translational-speed relationship that obtains for non-driven wheels. We write that relationship for the truck's rear wheels as follows:

$$\frac{d^2 x_{twr}}{dt^2} = R_{twr} \frac{d^2 \varphi_{twr}}{dt^2} \qquad (24)$$

But, we know more than that. The static friction which the wheels and ground can exert on each other is subject to a maximum value. At a high enough applied torque, the ground will be unable to resist the shear force applied by the wheels. They will begin to slip. The maximum frictional drag is calculated in a way similar to the dynamic drag, but using the coefficient of static friction instead. I will write the condition as:

$$\left.\begin{array}{c} F_{h,g\to twr}\Big|_{max} = C_{twr,s} F_{v,g\to twr} \\ \text{or} \\ F_{h,g\to twr} \le C_{twr,s} F_{v,g\to twr} \end{array}\right\} \qquad (25)$$

Note that this is an inequality. It does not affect the calculation of the horizontal force $F_{h,g\to twr}$ or participate in the solution of the equations of motion. Instead, it is a condition that determines when the non-slipping state comes to an end and slipping starts. Because it is an inequality, we will have to deal with it differently than we deal with the other equations, all of which are equalities.

Case Rear #2 -- The rear wheels are slipping

Now, let's consider the case when the rear wheels are slipping. When there is slippage, the drag force is equal to the vertical force across the interface multiplied by the coefficient of dynamic friction. I will assume that the coefficient of dynamic friction already includes the effects of noise, heat and other inefficiencies and so will not define another efficiency parameter. So, when there is slippage:

$$F_{h,g\to twr} = C_{twr,d} F_{v,g\to twr} \qquad (26)$$

Equation (26) does not depend on the applied torque. Nor does it depend on the speed with which the wheels are spinning. This has important physical consequences. Once the rear wheels start to slip, the ground reaction force is a constant. Pressing down on the gas pedal does not increase the ground reaction force; it simply makes the wheels spin faster. The only way to stop the spin is to take one's foot off the gas pedal and wait for the frictional drag to slow the wheels back down to a point where they grip the ground once more. Usually, this means waiting for the wheels to come to a stop. Waiting for a complete stop is not a mathematical necessity, but the result of two practical matters: (i) with one's foot off the gas pedal, there is no tactile feedback about the status of the drag, and (ii) sitting in the driver's seat, there are no visual cues either.

At any given time during a run, we will have to figure out which of one of the cases to use. It is one thing to be able to calculate the horizontal ground reaction force $F_{h,g\to twr}$ for one case or the other, but it is a different thing to decide which case should be used. Nor is it as simple as comparing the tangential rim speed of the wheels to their horizontal speed. If the two speeds are equal and opposite, the wheels are, by definition, not slipping. But, they could be just at the point where slipping begins. The inequality in Equation (25) will be our test for the transition point.

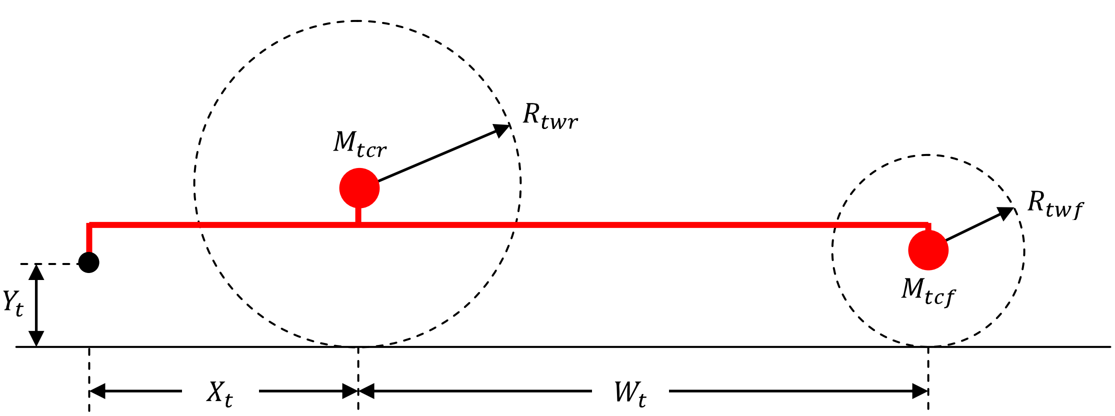
To handle these two with a single set of equations, I will introduce another binary flag. $B_{twrNoSlip}$ will have the value one when the wheels are not slipping and zero when they are. The subscript should help one remember that the flag is active when the wheels are not slipping. The equations which describe the dynamics of the truck's rear wheels can then be written as follows:

$(22A)$	$M_{twr}\dfrac{d^2x_{twr}}{dt^2} = F_{h,g\to twr} - F_{h,c\to twr}$	$(29A)$
(23)	$0 = F_{v,g\to twr} - F_{v,c\to twr} - M_{twr}g$	$(29B)$
$(22C)$	$-I_{twr}\dfrac{d^2\varphi_{twr}}{dt^2} = F_{h,g\to twr}R_{twr} - \tau_{c\to twr}$	$(29C)$
Case Rear #1	$B_{twrNoSlip}\left(\dfrac{d^2x_{twr}}{dt^2} - R_{twr}\dfrac{d^2\varphi_{twr}}{dt^2}\right) = 0$	(29D)
Case Rear #2	$(1 - B_{twrNoSlip})(F_{h,g\to twr} - C_{twr,d}F_{v,g\to twr}) = 0$	$(29E)$
Boundary test	$B_{twrNoSlip}(F_{h,g\to twr} \le C_{twr,s}F_{v,g\to twr})$	$(29F)$

Let's set the last equation -- the inequality -- aside for a moment. Although there are five other equations, only four will be non-trivial at any given time. Depending on the value of $B_{twrNoSlip}$, either Equation $(29D)$ or $(29E)$ will vanish. The set of equations involves six unknowns: accelerations in the two variables x_{twr} and φ_{twr}, a pair of ground forces and a pair of chassis forces. (Although the torque $\tau_{c\to twr}$ is a variable in the equations, it is not an unknown.) A solution should be possible. These four equations will be augmented by two more, which involve the two chassis forces, when the interaction between the truck's chassis and its rear wheels is taken into account.

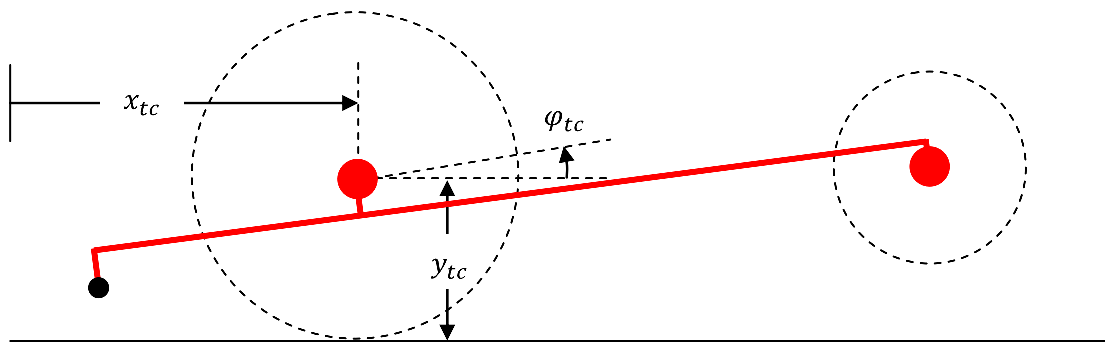
<u>Part VI -- Dynamics of the truck's chassis</u>

In the enhanced model, the front and rear wheels are treated as separate rigid bodies, so the chassis does not include them. The chassis does, however, feel their effects and the free body diagram will include the forces they exert. We are also going to add some hitch details to the enhanced model. The following figure shows the dimensions we will use for the truck.



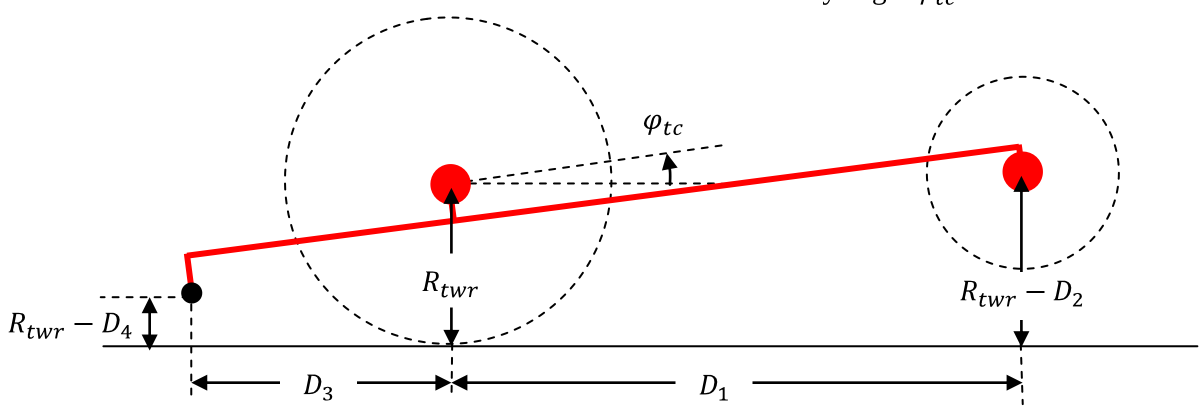
The chassis consists of the collection of red line segments, which should be thought of as having no mass. The effective masses of the chassis over the rear axle (M_{tcr}) and over the front axle $\left(M_{tcf}\right)$ are shown as the red dots. They are assumed to be concentrated at the centers of their respective axles. I have shown dotted circles which represent the front and rear wheels. They can have different diameters, and we will use the symbols R_{twf} and R_{twr} for their radii, respectively. Drawing the red line which represents the chassis horizontally at the vertical midpoint between the two axles is quite arbitrary. The hitch ball is shown by a small black dot. It is located a distance X_t behind the rear axle and a distance Y_t above the ground.

To keep track of the location of the chassis, we need to specify its horizontal and vertical co-ordinates. Because the front wheels might rise off the ground, we also need to keep track of the angle of the chassis's rotation. We will reference all three quantities to the center of the rear axle, as follows.



x_{tc} is the horizontal distance of the rear axle from its starting position. y_{tc} is the distance of the axle above the ground. Angle φ_{tc} is the angle by which the chassis has rotated in the counter-clockwise direction from the horizontal. The subscript tc stands for "truck chassis".

I am not going to draw the free body diagram just yet. Here's why. We intend to use the free body diagram as a reference to add up the forces and torques acting on the chassis. To calculate the torques, we will need to know the lever-arms through which the torquing forces act. However, the lengths of some of the lever-arms change if the chassis rotates. Therefore, I have shown in this next figure the distances from which we can calculate the lever-arms when the chassis is rotated by angle φ_{tc}.



where:

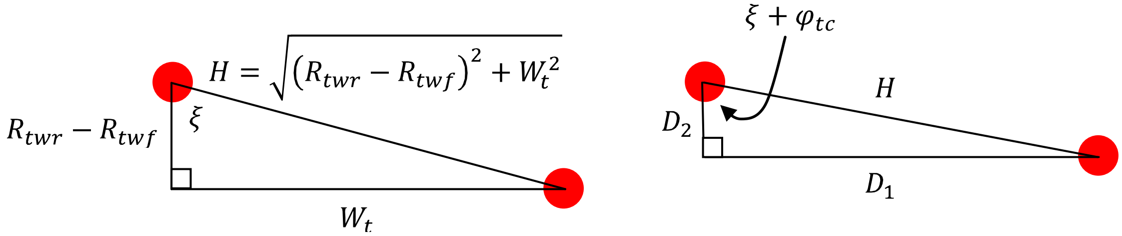
$$D_1 \quad = \sqrt{\left(R_{twr} - R_{twf}\right)^2 + W_t^2}\, \sin\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right] \quad (30A)$$

$$D_2 = \sqrt{\left(R_{twr} - R_{twf}\right)^2 + W_t^2}\cos\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right] \quad (30B)$$

$$D_3 = \sqrt{(R_{twr} - Y_t)^2 + X_t^2}\sin\left[\tan^{-1}\left(\frac{X_t}{R_{twr} - Y_t}\right) - \varphi_{tc}\right] \quad (30C)$$

$$D_4 = \sqrt{(R_{twr} - Y_t)^2 + X_t^2}\cos\left[\tan^{-1}\left(\frac{X_t}{R_{twr} - Y_t}\right) - \varphi_{tc}\right] \quad (30D)$$

Some trigonometry is required to determine distances D_1 through D_4. I will describe first how I determined the relationship between the front and rear axles, and distances D_1 and D_2. The following figure shows the relationship between the two axles (the red dots, as above) before any rotation (on the left) and after rotation by angle φ_{tc} (on the right).



Before the rotation, the vertical and horizontal separations of the two axles are the differences in wheel radii $\left(R_{twr} - R_{twf}\right)$ and the wheelbase (W_t), respectively. A right triangle can be set up with these two distances as its short sides. The hypotenuse H is the point-to-point distance between the centers of the axles and can be calculated using the Pythagorean Theorem. I have shown an angle ξ, which is the Greek letter "zi", as the angle between the vertical and the line segment connecting the axles.

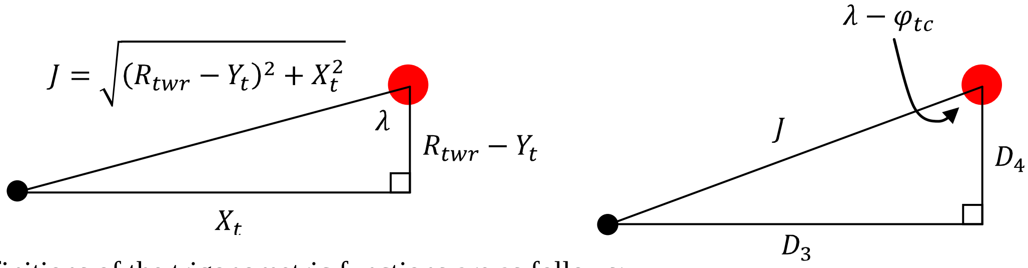
A similar right triangle can be set up after the rotation, as shown on the right. The vertical angle has increased by the amount of the rotation, and is now $\xi + \varphi_{tc}$. The hypotenuse stays the same, as H. Now, let's compare the sines, cosines and tangents of the vertical angles before and after.

$$\left.\begin{array}{llll} \multicolumn{2}{c}{\text{Before rotation}} & \multicolumn{2}{c}{\text{After rotation}} \\ \sin\xi & = \dfrac{W_t}{H} & \sin(\xi + \varphi_{tc}) & = \dfrac{D_1}{H} \\ \cos\xi & = \dfrac{R_{twr} - R_{twf}}{H} & \cos(\xi + \varphi_{tc}) & = \dfrac{D_2}{H} \\ \tan\xi & = \dfrac{W_t}{R_{twr} - R_{twf}} & \tan(\xi + \varphi_{tc}) & = \dfrac{D_1}{D_2} \end{array}\right\} \quad (31)$$

We can combine these equations to get:

$$\begin{aligned} D_1 &= H\sin(\xi + \varphi_{tc}) \\ &= H\sin\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right] \\ &= \sqrt{\left(R_{twr} - R_{twf}\right)^2 + W_t^2}\sin\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right] \quad (32A) \\ D_2 &= \sqrt{\left(R_{twr} - R_{twf}\right)^2 + W_t^2}\cos\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right] \quad (32B) \end{aligned}$$

The relationship between the hitch ball and the rear axle, from which distances D_3 and D_4 are calculated, is shown before and after the rotation in the following figure. Here, the unrotated vertical angle is shown as λ, the Greek letter "lambda". This vertical angle decreases when the chassis rotates.



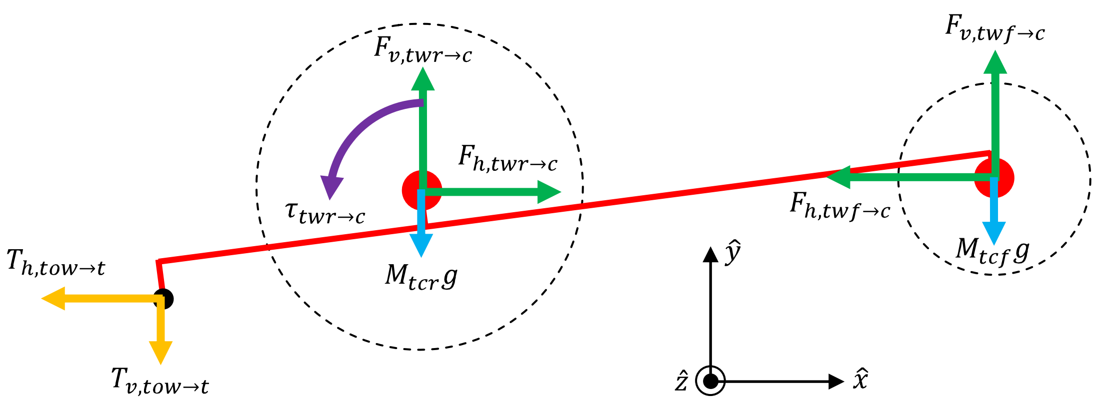
The definitions of the trigonometric functions are as follows:

$$\left.\begin{array}{llll}
\text{Before rotation} & & \text{After rotation} & \\
\sin\lambda & = \dfrac{X_t}{J} & \sin(\lambda - \varphi_{tc}) & = \dfrac{D_3}{J} \\
\cos\lambda & = \dfrac{R_{twr} - Y_t}{J} & \cos(\lambda - \varphi_{tc}) & = \dfrac{D_4}{J} \\
\tan\lambda & = \dfrac{X_t}{R_{twr} - Y_t} & \tan(\lambda - \varphi_{tc}) & = \dfrac{D_3}{D_4}
\end{array}\right\} \qquad (33)$$

which can be combined to give:

$$\begin{aligned}
D_3 &= J\sin(\lambda - \varphi_{tc}) \\
&= J\sin\left[\tan^{-1}\left(\frac{X_t}{R_{twr} - Y_t}\right) - \varphi_{tc}\right] \\
&= \sqrt{(R_{twr} - Y_t)^2 + X_t^2}\,\sin\left[\tan^{-1}\left(\frac{X_t}{R_{twr} - Y_t}\right) - \varphi_{tc}\right] \qquad (34A) \\
D_4 &= \sqrt{(R_{twr} - Y_t)^2 + X_t^2}\,\cos\left[\tan^{-1}\left(\frac{X_t}{R_{twr} - Y_t}\right) - \varphi_{tc}\right] \qquad (34B)
\end{aligned}$$

Now, we arrive at last at the free body diagram of the truck's chassis, which is shown in the following figure. Since we have just quantified the distances between important points on the chassis, I have not shown any dimensions in the free body diagram, just the forces and torques. To avoid any confusion, I have shown the chassis in a rotated position.



Three forces act at the point where the front mass is concentrated:

- $M_{tcf}g$ is the force of gravity acting on the concentrated front mass;
- $F_{v,twf\to c}$ is the vertical force which the front wheels exert on the chassis and
- $F_{h,twf\to c}$ is the horizontal force which the front wheels exert on the chassis.

As usual, I have picked a direction to use as the algebraically "positive" direction of each force. I have picked the direction in which we expect the forces to act when the truck is in normal operation. The front wheels, for example, must be pushed forwards, which results in their applying a retarding force on the chassis.

A similar set of three forces acts at the point where the rear mass is concentrated. There is also the reaction to the torque applied to the rear drive wheels. The relevant symbols are:

- $M_{tcr}g$ is the force of gravity acting on the concentrated rear mass;
- $F_{v,twr\to c}$ is the vertical force which the drive wheels exert on the chassis;
- $F_{h,twr\to c}$ is the horizontal force which the drive wheels exert on the chassis and
- $\tau_{twr\to c}$ is the reaction to the torque applied to the drive wheels.

The influence of the sled is felt through the tension exerted by the tow chain. The tow chain is no longer necessarily horizontal in the enhanced model, so we need to take its angle into account. The easiest way to do this is to resolve the tension into its vertical and horizontal components. This is done in the figure, where:

- $T_{h,tow\to t}$ is the horizontal component of the tension on the truck and
- $T_{v,tow\to t}$ is the vertical component of the tension on the truck.

Note that I have assumed that the tow chain is such that it will normally be pulling down on the truck's hitch. We now have enough information to write down the sum of the forces and torques acting on the chassis.

$$\sum F_{x,tc} = -F_{h,twf\to c} + F_{h,twr\to c} - T_{h,tow\to t} \qquad (35A)$$

$$\sum F_{y,tc} = F_{v,twf\to c} - M_{tcf}g + F_{v,twr\to c} - M_{tcr}g - T_{v,tow\to t} \qquad (35B)$$

$$\sum \tau_{z,tc} = \left\{ \begin{array}{cc} \tau_{twr\to c} + \cdots & \text{engine reaction torque} \\ \cdots + \left(F_{v,twf\to c} - M_{tcf}g\right)D_1 + \cdots & \text{vertical force on front mass} \\ \cdots - F_{h,twf\to c}D_2 + \cdots & \text{horizontal force on front mass} \\ \cdots + T_{v,tow\to t}D_3 + \cdots & \text{vertical force of tow chain} \\ \cdots - T_{h,tow\to t}D_4 & \text{horizontal force of tow chain} \end{array} \right\} \qquad (35C)$$

In the expression for the sum of the torques, I have identified the force which gives rise to each constituent torque. I have added or subtracted the torque from the total depending on whether it normally acts in the positive or negative $\hat{z}$-direction, respectively.

To apply Newton's Law, we need the total mass of the chassis, which is $M_{tcf} + M_{tcr}$, and the moment of inertia of the chassis, for which we will use the symbol I_{tc}. Since we expect that the truck will rotate around its rear axle if it rotates at all, we will want to use the moment of inertia which applies to rotations around that axis. The numerical value of the moment of inertia depends on the rotation axis chosen, in the same way that the numerical value of a torque depends what axis is chosen for measuring the lever-arm.

The accelerations of the chassis are slightly different from those for the wheels which we looked at above. The linear accelerations are the same but the rotational acceleration needs a positive sign rather than a negative sign. I defined angle φ_{tc}, which describes the rotation angle of the chassis, in such a way that φ_{tc} increases as the chassis rotates around the positive $\hat{z}$-axis. On the other hand, I defined the rotation angles φ for the wheels so that they increased by rotations around the negative $\hat{z}$-axis. For the chassis, the accelerations are given by:

$$
\begin{aligned}
&LinearAcceleration_{x,tc} &&= \frac{d^2 x_{tc}}{dt^2} \quad (36A)\\
&LinearAcceleration_{y,tc} &&= \frac{d^2 y_{tc}}{dt^2} \quad (36B)\\
&RotationalAcceleration_{z,tc} &&= \frac{d^2 \varphi_{tc}}{dt^2} \quad (36C)
\end{aligned}
$$

and the equations of motion can be written as:

$$
\begin{aligned}
\sum F_{x,tc} &= \left(M_{tcf} + M_{tcr}\right)\frac{d^2 x_{tc}}{dt^2} &&= -F_{h,twf\to c} + F_{h,twr\to c} - T_{h,tow\to t} && (37A)\\
\sum F_{y,tc} &= \left(M_{tcf} + M_{tcr}\right)\frac{d^2 y_{tc}}{dt^2} &&= F_{v,twf\to c} - \left(M_{tcf} + M_{tcr}\right)g + F_{v,twr\to c} - T_{v,tow\to t} && (37B)\\
\sum \tau_{z,tc} &= I_{tc}\frac{d^2 \varphi_{tc}}{dt^2} &&= \begin{bmatrix} \tau_{twr\to c} + \left(F_{v,twf\to c} - M_{tcf}g\right)D_1 + \cdots \\ \cdots - F_{h,twf\to c}D_2 + T_{v,tow\to t}D_3 - T_{h,tow\to t}D_4 \end{bmatrix} && (37C)
\end{aligned}
$$

We have only one constraint: that the net force in the $\hat{y}$-direction is zero. We know this because the chassis does not rise off the ground in the vertical direction. That is not to say that the front wheels cannot rise up off the ground. They can but, when they do, it is a result of the chassis's rotation, not a result of the chassis's upward translation. Enforcing the constraint allows us to write this set of equations as follows:

$$
\begin{aligned}
\left(M_{tcf} + M_{tcr}\right)\frac{d^2 x_{tc}}{dt^2} &= -F_{h,twf\to c} + F_{h,twr\to c} - T_{h,tow\to t} && (38A)\\
0 &= F_{v,twf\to c} - \left(M_{tcf} + M_{tcr}\right)g + F_{v,twr\to c} - T_{v,tow\to t} && (38B)\\
I_{tc}\frac{d^2 \varphi_{tc}}{dt^2} &= \begin{bmatrix} \tau_{twr\to c} + \left(F_{v,twf\to c} - M_{tcf}g\right)D_1 + \cdots \\ \cdots - F_{h,twf\to c}D_2 + T_{v,tow\to t}D_3 - T_{h,tow\to t}D_4 \end{bmatrix} && (38C)
\end{aligned}
$$

There are eight unknowns in these three equations: accelerations in the two variables x_{tc} and φ_{tc}, a pair of forces from the front wheels, a pair of forces from the rear wheels and a pair of forces from the tow chain. When we combine these equations with those of the front wheels, we will in effect be adding two more equations, which involve the two front wheel forces. When we combine these equations with those of the rear wheels, we will be adding two more equations, which involve the two rear wheel forces. Similarly, when we combine these equations with those of the sled, we will be adding another two, which involve the tow chain forces. In effect, we will have a total of $3 + 2 + 2 + 2 = 9$ equations.

At first blush, it looks like we have too many equations, one more than the number of unknowns. But, that's not true. We have over-counted both the number of unknowns and the number of equations. For example, the horizontal location of the chassis x_{tc} is not really a new unknown. We are measuring the horizontal location of the truck's chassis to the center of the rear axle, which is the same spot to which we are measuring the horizontal location of the truck's rear wheels x_{twr}. They will always have the same value, notwithstanding that we have used different symbols while setting up the equations of motion. The same applies to the rotation angle φ_{tc}, which can be derived from knowledge of the vertical displacement y_{twf} of the front wheels. There are other kinematic redundancies, which I will describe below. We are also over-counting the number of equations. Combining the chassis's equations of motion with those of the front wheels does not "add" another two equations to this set. It would be more accurate to say that it "removes" two equations from this set, since we already counted the information from two extra equations when we compared the numbers of unknowns and equations when analyzing the front wheels.

This is probably a good time to stop trying to predict solubility of the equations on a rigid body-by-rigid body basis. Despite my efforts to do so for the three sets of wheels, things are involved enough that we should defer a final check until all the equations have been written down and redundancies eliminated.

Before moving on, though, I want to digress for a moment to deal with one loose end: the moment of inertia of the truck's chassis. We have set up the dynamics so that the truck's rear axle is the axis of rotation, so we will want to know the moment of inertia of the chassis as it rotates around the rear axle. We modeled the chassis as two point masses. One of them is concentrated at the rear axle itself. Since its lever-arm is zero, it makes no contribution to the moment of inertia. The other point mass is concentrated at the front axle, whose distance away from the rear axle is approximately equal to the wheelbase W_t. More precisely, the separation between the axes is equal to the hypotenuse H shown in one of the figures above, whose numerical value is equal to $\sqrt{\left(R_{twr} - R_{twf}\right)^2 + W_t^2}$. Using the formula for the moment of inertia of a point mass, we get:

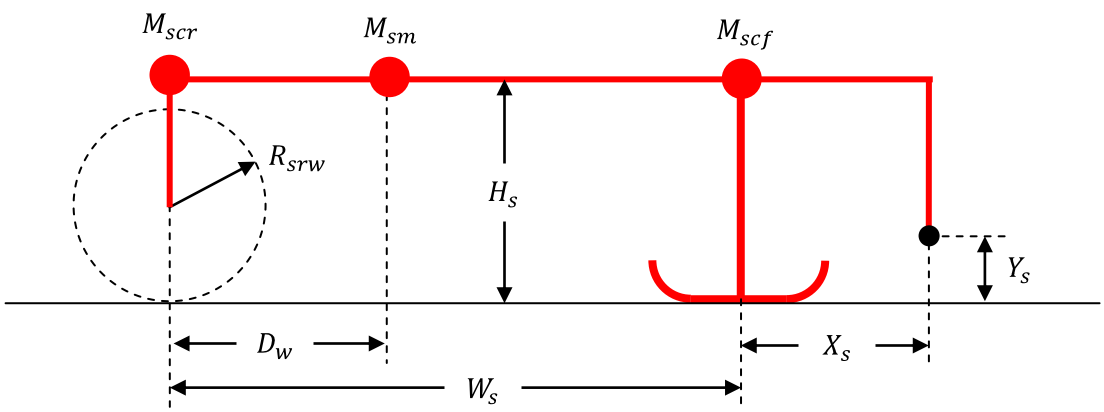
$$I_{tc} > M_{tcf}\left[\left(R_{twr} - R_{twf}\right)^2 + W_t^2\right] \quad (39)$$

The right-hand side of Equation (39) is the front mass M_{tcf} multiplied by the square of the lever-arm. So, why is there a "greater than" sign? Philosophically, the right-hand side of Equation (39) is the moment of inertia of the revolution of the front-end mass around the rear axle. What is missing from the right-hand side of Equation (39) is the additional moment of inertia which applies when bits of the chassis are rotated around each other. In order to do a more detailed calculation of the moment of inertia, we would need a lot more information about the location and weight of the various components which make up the truck. If we use the estimate of I_{tc} set out in Equation (39), we will under-estimate the moment of inertia of the truck's chassis. We should ask the question: is this under-estimation a problem? One can argue that it is not.

Our objective is to win truck pulls. If the front wheels of the truck leave the ground, it is a sign that the truck's ability to pull the sled further has pretty much come to an end. Under-estimating the moment of inertia will result in over-estimating the rotational acceleration of the chassis. The chassis will rotate more quickly than the use of the correct, and larger, moment of inertia would predict. Rotation of the chassis is an interesting phenomenon to see but, once it starts, there will not be much, if any, further progress down the course. In any event, if you can find or derive a better value for I_{tc} than the estimate in Equation (39), use it.

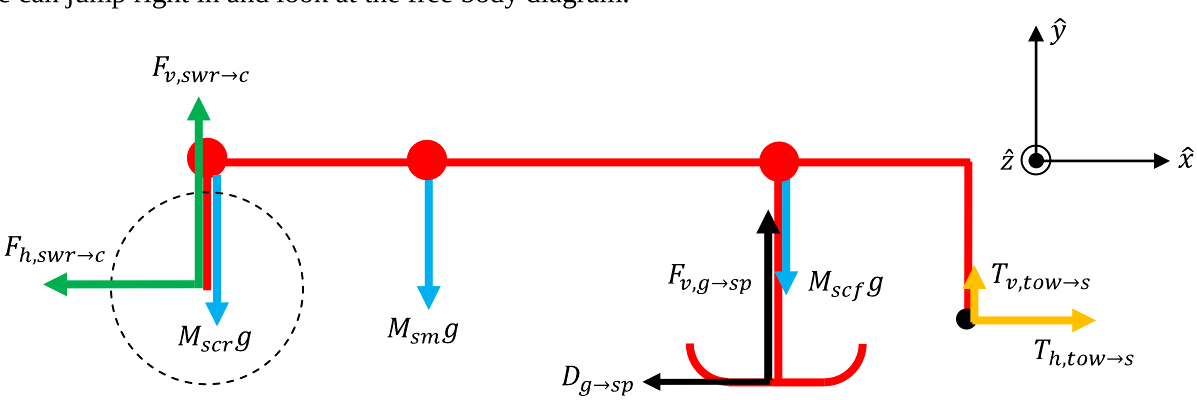
Part VII -- Dynamics of the sled's chassis

In this section, we will look at the sled or, more precisely, the chassis of the sled. The rear wheels of the sled have been dealt with as a separate rigid body. The following figure shows the dimensions of the sled we will consider in the enhanced model.



The sled has not changed too much from the simplified model. It still has the masses concentrated over the rear wheels (M_{scr}) and over the center-line of the pan $\left(M_{scf}\right)$. Although the sled has not changed physically, the numerical values of one or both of these masses will need to be changed because the rear wheels are no longer included as part of the chassis. The wheelbase of the sled (W_s) is defined as it was before. The moveable weight has the same mass (M_{sm}) and instantaneous relative location (D_w) as it had before.

I have changed the definition of the deck height H_s. Here, it is equal to the distance above the ground of the center of mass of the moveable weight. In practice, this height will be somewhat above the top of the rails which carry the moveable weight. Also new is the hitch, whose ball is located a distance X_s ahead of the center-line of the pan and a distance Y_t up from the ground. Since the sled is not expected to rotate, we can jump right in and look at the free body diagram.



There are three forces which act in the neighbourhood of the rear of the sled.

- $M_{scr}g$ is the force of gravity acting on the concentrated rear mass;
- $F_{v,swr\to c}$ is the vertical force which the rear wheels exert on the chassis and
- $F_{h,swr\to c}$ is the horizontal force which the rear wheels exert on the chassis.

I have assumed that the first force, the gravitational one, acts at a point at deck height. The latter two forces, which are equal and opposite to the forces which the sled's chassis exerts on the rear wheels, are assumed to act at the center of the rear axle.

The force of gravity on the moveable weight is unchanged from the simplified model. It is $M_{sm}g$ and is assumed to act at the center of mass of the weight.

There are three forces which act in the neighbourhood of the pan.

- $M_{scf}g$ is the force of gravity acting on the concentrated front mass;
- $F_{v,g\to sp}$ is the vertical force with which the ground pushes up on the pan and
- $D_{g\to sp}$ is the horizontal drag of the ground on the bottom of the pan. (Consistency might suggest that one use the symbol $F_{h,g\to sp}$ for this important force. I will defy consistency and continue to use D for drag.)

The force of the tow chain on the hitch completes the picture. As with the truck, it is easiest if we resolve the tension in the tow chain into its horizontal $\left(T_{h,tow\to s}\right)$ and vertical $\left(T_{v,tow\to s}\right)$ components.

We can now write down the sums of the forces. We get:

$$\sum F_{x,sc} = T_{h,tow\to s} - D_{g\to sp} - F_{h,swr\to c} \qquad (40A)$$
$$\sum F_{y,sc} = T_{v,tow\to s} + F_{v,g\to sp} - M_{scf}g - M_{sm}g + F_{v,swr\to c} - M_{scr}g \qquad (40B)$$

To write down the sum of the torques, we need to pick an axis of rotation. For convenience, we will use the center of the rear axle. Bear in mind that the radius of the rear axle is not the same as the height of the deck. The sum of the torques is:

$$\sum \tau_{z,sc} = \left\{\begin{array}{cc} T_{v,tow\to s}(W_s + X_s) + \cdots & \text{vertical force of tow chain} \\ \cdots + T_{h,tow\to s}(R_{swr} - Y_s) + \cdots & \text{horizontal force of tow chain} \\ \cdots + \left(F_{v,g\to sp} - M_{scf}g\right)W_s + \cdots & \text{vertical forces at pan} \\ \cdots - D_{g\to sp}R_{swr} + \cdots & \text{drag force on pan} \\ \cdots - M_{sm}gD_w & \text{gravity on moveable weight} \end{array}\right\} \qquad (40C)$$

To write down the equations of motion, we need to know the total mass of the sled's chassis, which is equal to $M_{scf} + M_{sm} + M_{scr}$. If we were purists, we might also want to know the moment of inertia of the sled's chassis. But it is not something we need to know. Since the sled will never rotate, we can set the sum of the torques to zero. We can do that merely by setting the right-hand side of Equation $(40C)$ to zero. In addition, since the sled will remain on the ground, we can set the sum of the vertical forces to zero. The only motion the sled makes is a horizontal translation. We will use the symbol x_{sc} for the instantaneous location of the rear axle with respect to its location when the rig is at the starting line. Then, the equations of motion are:

$$\left(M_{scf} + M_{sm} + M_{scr}\right)\frac{d^2x_{sc}}{dt^2} = T_{h,tow\to s} - D_{g\to sp} - F_{h,swr\to c} \qquad (41A)$$
$$0 = T_{v,tow\to s} + F_{v,g\to sp} + F_{v,swr\to c} - \left(M_{scf} + M_{sm} + M_{scr}\right)g \qquad (41B)$$
$$0 = \left[\begin{array}{c} T_{v,tow\to s}(W_s + X_s) + T_{h,tow\to s}(R_{swr} - Y_s) + \cdots \\ \cdots + \left(F_{v,g\to sp} - M_{scf}g\right)W_s - D_{g\to sp}R_{swr} - M_{sm}gD_w \end{array}\right] \qquad (41C)$$

Part VIII - The ground reaction forces on the pan

When the sled is moving, the horizontal drag on the pan $D_{g\to sp}$ is determined by the coefficient of dynamic friction C_{pd}, as follows:

$$D_{g \to sp} = C_{pd} F_{v,g \to sp} \quad (42)$$

where $F_{v,g \to sp}$ is the vertical force acting across the boundary between the bottom of the pan and the ground.

When the sled is stopped, a slightly different relationship holds:

$$\left.\begin{array}{c} D_{g \to sp}\big|_{max} = C_{ps} F_{v,g \to sp} \\ or \\ D_{g \to sp} \leq C_{ps} F_{v,g \to sp} \end{array}\right\} \quad (43)$$

where the maximum frictional drag the ground can sustain is equal to the coefficient of <u>static</u> friction multiplied by the vertical force across the boundary. This is an inequality, so will not help in finding the value of $D_{g \to sp}$. Instead, Equation (43) is a condition that determines when the pan's non-slipping state comes to an end and the sled starts moving. However, when the sled is stopped and expected to remain stopped, we know that the horizontal acceleration is equal to zero:

$$\frac{d^2 x_{sc}}{dt^2} = 0 \quad (44)$$

It is apparent that there are two cases for the drag on the pan, one when the sled is moving and the other when it is stopped. The procedure to calculate the drag $D_{g \to sp}$ is going to differ between the two cases, but the same equations of motion, Equation (41), apply equally to both cases. Like we have done before, we can combine the cases into a single set using another binary flag. For this purpose, we will use the flag $B_{SledMove}$, where the subscript refers to the "<u>s</u>led <u>m</u>oving" and the value is set to one when the sled is moving. When the sled is stopped, $B_{SledMove} = 0$. Then, the equations which describe the dynamics of the sled are as follows:

$$\begin{array}{lll}
(41A) & \left(M_{scf} + M_{sm} + M_{scr}\right) \dfrac{d^2 x_{sc}}{dt^2} = T_{h,tow \to s} - D_{g \to sp} - F_{h,swr \to c} & (45A) \\
(41B) & 0 = T_{v,tow \to s} + F_{v,g \to sp} + F_{v,swr \to c} - \left(M_{scf} + M_{sm} + M_{scr}\right) g & (45B) \\
(41C) & 0 = \begin{bmatrix} T_{v,tow \to s}(W_s + X_s) + T_{h,tow \to s}(R_{swr} - Y_s) + \cdots \\ \cdots + \left(F_{v,g \to sp} - M_{scf} g\right) W_s - D_{g \to sp} R_{swr} - M_{sm} g D_w \end{bmatrix} & (45C) \\
(42) & B_{SledMove}\left(D_{g \to sp} - C_{pd} F_{v,g \to sp}\right) = 0 & (45D) \\
(44) & (1 - B_{SledMove}) \dfrac{d^2 x_{sc}}{dt^2} = 0 & (45E) \\
\text{Boundary test} & (1 - B_{SledMove})\left(D_{g \to sp} \leq C_{ps} F_{v,g \to sp}\right) & (45F)
\end{array}$$

The alert reader may compare what we have done here with what we did with the truck's front wheels. For the front wheels, we divided things into three cases: (i) the front wheels airborne, (ii) the front wheels on the ground and expected to remain on the ground and (iii) the front wheels on the ground but also on the verge of going airborne. Why do we not have three cases here: (i) the sled moving, (ii) the sled stopped and expected to stay stopped and (iii) the sled stopped but on the verge of starting to move? It is not necessary here since the difference between the two cases only involves two variables: a horizontal force and a horizontal acceleration. The differences between the cases for the front wheels involved three variables: a relative tangential speed, a vertical force and a vertical acceleration. If took more work, and two binary flags, to sort out the possibilities there.

Part IX -- Collating the equations of motion

Let me bring together in this section, for future reference, all of the equations of motion and the constraints which we identified above. I will make the following changes before transcribing them.

There are certain pairs of forces (and a torque) between some of the rig's constituent parts which are at all times equal and opposite. The free body diagrams were drawn in such a way that these equal and opposite forces (and the torque) were drawn in opposing directions. No minus signs are needed. We can set equal the magnitudes of the following equal and opposite forces (and the torque):

$$\left.\begin{aligned} F_{h,c\to twf} &= F_{h,twf\to c} \\ F_{v,c\to twf} &= F_{v,twr\to c} \\ F_{h,c\to swr} &= F_{h,swr\to c} \\ F_{v,c\to swr} &= F_{v,swr\to c} \\ F_{h,c\to twr} &= F_{h,twr\to c} \\ F_{v,c\to twr} &= F_{v,twr\to c} \\ \tau_{c\to twr} &= \tau_{twr\to c} \end{aligned}\right\} \qquad (46)$$

In addition, the forces at the end of the tow chain must always be equal and opposite:

$$\left.\begin{aligned} T_{h,tow\to t} &= T_{h,tow\to s} \\ T_{v,tow\to t} &= T_{v,tow\to s} \end{aligned}\right\} \qquad (47)$$

We will substitute the variables on the left-hand sides of Equations (46) and (47) for the variables on the right-hand sides wherever they occur. The complete equations of dynamics can then be written as:

$$h, twf \qquad M_{twf}\frac{d^2x_{twf}}{dt^2} = F_{h,c\to twf} - F_{h,g\to twf} \qquad (48A)$$

$$v, twf \qquad M_{twf}\frac{d^2y_{twf}}{dt^2} = F_{v,g\to twf} - F_{v,c\to twf} - M_{twf}g \qquad (48B)$$

$$z, twf \qquad I_{twf}\frac{d^2\varphi_{twf}}{dt^2} = F_{h,g\to twf}R_{twf} \qquad (48C)$$

$$\text{Case Front \#1} \qquad B_{twfAir}F_{h,g\to twf} = 0 \qquad (48D)$$

$$\text{Cases Front \#1, 2B} \qquad \left[B_{twfAir} + (1 - B_{twfAir})B_{twfRTO}\right]F_{v,g\to twf} = 0 \qquad (48E)$$

$$\text{Case Front \#2} \qquad (1 - B_{twfAir})\left(\frac{d^2x_{twf}}{dt^2} - R_{twf}\frac{d^2\varphi_{twf}}{dt^2}\right) = 0 \qquad (48F)$$

$$\text{Case Front \#2A} \qquad (1 - B_{twfAir})(1 - B_{twfRTO})\frac{d^2y_{twf}}{dt^2} = 0 \qquad (48G)$$

$$h, swr \qquad M_{swr}\frac{d^2x_{swr}}{dt^2} = F_{h,c\to swr} - F_{h,g\to swr} \qquad (48H)$$

$$v, swr \qquad F_{v,g\to swr} - F_{v,c\to swr} - M_{swr}g = 0 \qquad (48I)$$

$$z, swr \qquad I_{swr}\frac{d^2\varphi_{swr}}{dt^2} = F_{h,g\to swr}R_{swr} \qquad (48J)$$

$$(11D) \qquad \frac{d^2x_{swr}}{dt^2} = R_{swr}\frac{d^2\varphi_{swr}}{dt^2} \qquad (48K)$$

and

h, twr	$M_{twr}\dfrac{d^2x_{twr}}{dt^2} = F_{h,g\to twr} - F_{h,c\to twr}$	$(48L)$
v, twr	$0 = F_{v,g\to twr} - F_{v,c\to twr} - M_{twr}g$	$(48M)$
z, twr	$I_{twr}\dfrac{d^2\varphi_{twr}}{dt^2} = \tau_{c\to twr} - F_{h,g\to twr}R_{twr}$	$(48N)$
Case Rear #1	$B_{twrNoSlip}\left(\dfrac{d^2x_{twr}}{dt^2} - R_{twr}\dfrac{d^2\varphi_{twr}}{dt^2}\right) = 0$	$(48O)$
Case Rear #2	$\left(1 - B_{twrNoSlip}\right)\left(F_{h,g\to twr} - C_{twr,d}F_{v,g\to twr}\right) = 0$	$(48P)$
Rear boundary test	$B_{twrNoSlip}\left(F_{h,g\to twr} \le C_{twr,s}F_{v,g\to twr}\right)$	$(48Q)$
h, tc	$\left(M_{tcf} + M_{tcr}\right)\dfrac{d^2x_{tc}}{dt^2} = -F_{h,c\to twf} + F_{h,c\to twr} - T_{h,tow\to t}$	$(48R)$
v, tc	$0 = F_{v,c\to twf} - \left(M_{tcf} + M_{tcr}\right)g + F_{v,c\to twr} - T_{v,tow\to t}$	$(48S)$
z, tc	$I_{tc}\dfrac{d^2\varphi_{tc}}{dt^2} = \begin{bmatrix} \tau_{c\to twr} + \left(F_{v,c\to twf} - M_{tcf}g\right)D_1 + \cdots \\ \cdots - F_{h,c\to twf}D_2 + T_{v,tow\to t}D_3 - T_{h,tow\to t}D_4 \end{bmatrix}$	$(48T)$
h, wc	$\left(M_{scf} + M_{sm} + M_{scr}\right)\dfrac{d^2x_{sc}}{dt^2} = T_{h,tow\to t} - D_{g\to sp} - F_{h,c\to swr}$	$(48U)$
v, sc	$0 = T_{v,tow\to t} + F_{v,g\to sp} + F_{v,c\to swr} - \left(M_{scf} + M_{sm} + M_{scr}\right)g$	$(48V)$
z, sc	$0 = \begin{bmatrix} T_{v,tow\to t}(W_s + X_s) + T_{h,tow\to t}(R_{swr} - Y_s) + \cdots \\ \cdots + \left(F_{v,g\to sp} - M_{scf}g\right)W_s - D_{g\to sp}R_{swr} - M_{sm}gD_w \end{bmatrix}$	(48W)
Case Sled #1	$B_{SledMove}\left(D_{g\to sp} - C_{pd}F_{v,g\to sp}\right) = 0$	$(48X)$
Case Sled #2	$(1 - B_{SledMove})\dfrac{d^2x_{sc}}{dt^2} = 0$	$(48Y)$
Sled boundary test	$(1 - B_{SledMove})\left(D_{g\to sp} \le C_{ps}F_{v,g\to sp}\right)$	$(48Z)$

I will eliminate the internal variables which represent the forces between the wheels and their respective chasses. Although the tensions in the tow chain are internal forces from the point-of-view of the rig, I will not eliminate them. I will use the following equations as the sources expressions for the internal variables to be eliminated.

Variable	**Equation**
$F_{h,c\to twf}$	$(48A)$
$F_{v,c\to twf}$	$(48B)$
$F_{h,c\to swr}$	$(48H)$
$F_{v,c\to swr}$	$(48I)$
$F_{h,c\to twr}$	$(48L)$
$F_{h,c\to twr}$	$(48M)$

The set of equations which remains after the substitutions is as follows. For convenience in tracing, I have not changed the letter suffixes in the equation identification numbers.

z, twf	$I_{twf}\dfrac{d^2\varphi_{twf}}{dt^2} = F_{h,g\to twf}R_{twf}$	$(49C)$
Case Front #1	$B_{twfAir}F_{h,g\to twf} = 0$	$(49D)$

Cases Front #1, 2B
$$\left[B_{twfAir} + (1 - B_{twfAir})B_{twfRTO}\right]F_{v,g\to twf} = 0 \tag{49E}$$

Case Front #2
$$(1 - B_{twfAir})\left(\frac{d^2x_{twf}}{dt^2} - R_{twf}\frac{d^2\varphi_{twf}}{dt^2}\right) = 0 \tag{49F}$$

Case Front #2A
$$(1 - B_{twfAir})(1 - B_{twfRTO})\frac{d^2y_{twf}}{dt^2} = 0 \tag{49G}$$

z, swr
$$I_{swr}\frac{d^2\varphi_{swr}}{dt^2} = F_{h,g\to swr}R_{swr} \tag{49J}$$

$(11D)$
$$\frac{d^2x_{swr}}{dt^2} = R_{swr}\frac{d^2\varphi_{swr}}{dt^2} \tag{49K}$$

z, twr
$$I_{twr}\frac{d^2\varphi_{twr}}{dt^2} = \tau_{c\to twr} - F_{h,g\to twr}R_{twr} \tag{49N}$$

Case Rear #1
$$B_{twrNoSlip}\left(\frac{d^2x_{twr}}{dt^2} - R_{twr}\frac{d^2\varphi_{twr}}{dt^2}\right) = 0 \tag{49O}$$

Case Rear #2
$$\left(1 - B_{twrNoSlip}\right)\left(F_{h,g\to twr} - C_{twr,d}F_{v,g\to twr}\right) = 0 \tag{49P}$$

h, tc
$$\begin{bmatrix} \left(M_{tcf} + M_{tcr}\right)\dfrac{d^2x_{tc}}{dt^2} + \cdots \\ \cdots + M_{twf}\dfrac{d^2x_{twf}}{dt^2} + M_{twr}\dfrac{d^2x_{twr}}{dt^2} \end{bmatrix} = -F_{h,g\to twf} + F_{h,g\to twr} - T_{h,tow\to t} \tag{49R}$$

v, tc
$$M_{twf}\frac{d^2y_{twf}}{dt^2} = F_{v,g\to twf} + F_{v,g\to twr} - T_{v,tow\to t} - M_{truck}g \tag{49S}$$

z, tc
$$\begin{bmatrix} M_{twf}D_2\dfrac{d^2x_{twf}}{dt^2} + \cdots \\ \cdots + M_{twf}D_1\dfrac{d^2y_{twf}}{dt^2} + \cdots \\ \cdots + I_{tc}\dfrac{d^2\varphi_{tc}}{dt^2} \end{bmatrix} = \begin{bmatrix} \tau_{c\to twr} + \cdots \\ F_{v,g\to twf}D_1 - F_{h,g\to twf}D_2 + \cdots \\ \cdots + T_{v,tow\to t}D_3 - T_{h,tow\to t}D_4 + \cdots \\ \cdots - \left(M_{twf} + M_{tcf}\right)D_1 g \end{bmatrix} \tag{49T}$$

h, wc
$$\begin{bmatrix} \left(M_{scf} + M_{sm} + M_{scr}\right)\dfrac{d^2x_{sc}}{dt^2} + \cdots \\ \cdots + M_{swr}\dfrac{d^2x_{swr}}{dt^2} \end{bmatrix} = T_{h,tow\to t} - D_{g\to sp} - F_{h,g\to swr} \tag{49U}$$

v, sc
$$0 = T_{v,tow\to t} + F_{v,g\to sp} + F_{v,g\to swr} - M_{sled}g \tag{49V}$$

z, sc
$$0 = \begin{bmatrix} T_{v,tow\to t}(W_s + X_s) + T_{h,tow\to t}(R_{swr} - Y_s) + \cdots \\ \cdots + \left(F_{v,g\to sp} - M_{scf}g\right)W_s - D_{g\to sp}R_{swr} - M_{sm}gD_w \end{bmatrix} \tag{49W}$$

Case Sled #1
$$B_{SledMove}\left(D_{g\to sp} - C_{pd}F_{v,g\to sp}\right) = 0 \tag{49X}$$

Case Sled #2
$$(1 - B_{SledMove})\frac{d^2x_{sc}}{dt^2} = 0 \tag{49Y}$$

Rear boundary test
$$B_{twrNoSlip}\left(F_{h,g\to twr} \le C_{twr,s}F_{v,g\to twr}\right) \tag{49Q}$$

Sled boundary test
$$(1 - B_{SledMove})\left(D_{g\to sp} \le C_{ps}F_{v,g\to sp}\right) \tag{49Z}$$

How many unknowns are there? There are six forces acting between separate rigid bodies in the rig; they are listed in Equation (46). The torque is not an unknown; the assumptions we will make about the powertrain will quantify the torque. There are the two forces acting on the tow chain, as listed in Equation (47). There are four vertical ground-reaction forces -- $F_{v,g\to twf}$, $F_{v,g\to swr}$, $F_{v,g\to twr}$ and $F_{v,g\to sp}$ -- one for each point of contact between the rig and the ground. There are four corresponding horizontal ground-reaction forces -- $F_{h,g\to twf}$, $F_{h,g\to swr}$, $F_{h,g\to twr}$ and $D_{g\to sp}$. There are five horizontal distances -- x_{twf}, x_{swr}, x_{twr}, x_{tc} and x_{sc} -- one for each of the constituent rigid bodies. There is only one vertical distance -- y_{twf} -- since the truck's front wheels are the only ones which can rise off the ground. And, there are four rotation angles -- φ_{twf}, φ_{swr}, φ_{twr} and φ_{tc} -- one for each pair of wheels and one for the truck's chassis. The parameters D_1 through D_4 are not independent variables; they can all expressed in terms of the angle φ_{tc}, which we have already noted as an independent variable.

This is a total of $6+2+4+4+5+1+4=26$ variables. There are 20 equations, not all of which will be non-trivial at any time.

We will not be able to get a unique solution unless the number of non-trivial equations is equal to the number of variables. We are going to need more pieces of information. Here is how I think we can get at least some of them.

1. We do not need five different horizontal distances. Let's pick the location of the rear axle of the sled x_{sc} as the one which is truly independent. We can express the other four distances in terms of distance x_{sc} and angle φ_{tc}.
2. We can also express the vertical distance y_{twf} in terms of the truck's rotation angle φ_{tc}.
3. The horizontal and vertical components of the tension in the tow chain are not independent. They can both be expressed in terms of a single variable, the total tension force in the tow chain, for example, and the truck's rotation angle φ_{tc}.
4. The binary flags vary among the cases. Some settings of these flags, either to one or to zero, cause some equations to become trivial or redundant, but also set the values of some of the variables. I am hopeful that the dependencies will work themselves out.

The additional information described in paragraphs 1 through 3 are related to the tow chain. I will describe this matter in the next section.

Part X -- The tow chain

The tow chain is the only connection between sled and the truck. Its length and angle determine their their horizontal separation. If the truck does not rotate, then the tow chain is static and the horizontal separation remains constant. Things become interesting when the truck rotates. Not only does the horizontal separation of the vehicles change, which requires that their horizontal accelerations be different, but the ratio of the horizontal and vertical forces which are transmitted between them changes as well.

I set up the variables so that the horizontal locations of the sled's chassis x_{sc} and the sled's rear wheels x_{swr} are measured to the same spot, being the center of the sled's rear axle. Similarly, the horizontal locations of the truck's chassis x_{tc} and rear wheels x_{twr} are measured to the same spot, being the center of the truck's rear axle. Therefore:

$$x_{swr}=x_{sc}\quad\to\frac{dx_{swr}}{dt}=\frac{dx_{sc}}{dt}\quad\to\frac{d^2x_{swr}}{dt^2}=\frac{d^2x_{sc}}{dt^2}\quad(50A)$$
$$x_{twr}=x_{tc}\quad\to\frac{dx_{twr}}{dt}=\frac{dx_{tc}}{dt}\quad\to\frac{d^2x_{twr}}{dt^2}=\frac{d^2x_{tc}}{dt^2}\quad(50B)$$

Let's look at the front wheels of the truck, whose horizontal and vertical locations are given by x_{twf} and y_{twf}, respectively. When the truck is on the ground, the horizontal separation between its axles is equal to the wheelbase W_t. If the truck rotates, the horizontal separation is the distance D_1 which we illustrated and quantified in an earlier section. When the truck is on the ground, the front axle is a distance R_{twf} above the ground. If the truck rotates, this distance increases to $R_{twr} - D_2$, where distance D_2 was also illustrated and quantified in the earlier section. Recall that D_1 and D_2 depend on the rotation angle of the chassis φ_{tc}. In any event, we can express the absolute co-ordinates of the front wheels in terms of chassis variables as follows:

$$\begin{aligned} x_{twf} &= x_{tc} + D_1 \qquad (51A) \\ y_{twf} &= R_{twr} - D_2 \quad (51B) \end{aligned}$$

To calculate the horizontal and vertical speeds of the front wheels, we need to take the derivatives of x_{twf} and y_{twf}. We proceed as follows, using the expressions for D_1 and D_2 from above:

$$\begin{aligned} \frac{dx_{twf}}{dt} &= \frac{dx_{tc}}{dt} + \frac{d}{dt}\left\{\sqrt{\left(R_{twr} - R_{twf}\right)^2 + W_t^2}\sin\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right]\right\} \\ &= \frac{dx_{tc}}{dt} + \sqrt{\left(R_{twr} - R_{twf}\right)^2 + W_t^2}\cos\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right]\frac{d\varphi_{tc}}{dt} \quad (52A) \end{aligned}$$

$$\begin{aligned} \frac{dy_{twf}}{dt} &= -\frac{d}{dt}\left\{\sqrt{\left(R_{twr} - R_{twf}\right)^2 + W_t^2}\cos\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right]\right\} \\ &= \sqrt{\left(R_{twr} - R_{twf}\right)^2 + W_t^2}\sin\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right]\frac{d\varphi_{tc}}{dt} \qquad (52B) \end{aligned}$$

Take a close look at the arguments of the sine and cosine terms, and their coefficients. After comparison with the definitions of D_1 and D_2, it is apparent that:

$$\begin{aligned} \frac{dx_{twf}}{dt} &= \frac{dx_{tc}}{dt} + D_2\frac{d\varphi_{tc}}{dt} \quad (53A) \\ \frac{dy_{twf}}{dt} &= -D_1\frac{d\varphi_{tc}}{dt} \qquad (53B) \end{aligned}$$

Then, for the accelerations, we derive a second time, as follows:

$$\begin{aligned} \frac{d^2x_{twf}}{dt^2} &= \frac{d^2x_{tc}}{dt^2} + \frac{d}{dt}\left\{\sqrt{\left(R_{twr} - R_{twf}\right)^2 + W_t^2}\cos\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right]\frac{d\varphi_{tc}}{dt}\right\} \\ &= \frac{d^2x_{tc}}{dt^2} + \sqrt{\left(R_{twr} - R_{twf}\right)^2 + W_t^2}\left\{\begin{array}{c} -\sin\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right]\left(\frac{d\varphi_{tc}}{dt}\right)^2 + \cdots \\ \cdots + \cos\left[\tan^{-1}\left(\frac{W_t}{R_{twr} - R_{twf}}\right) + \varphi_{tc}\right]\frac{d^2\varphi_{tc}}{dt^2} \end{array}\right\} \quad (54A) \end{aligned}$$

and

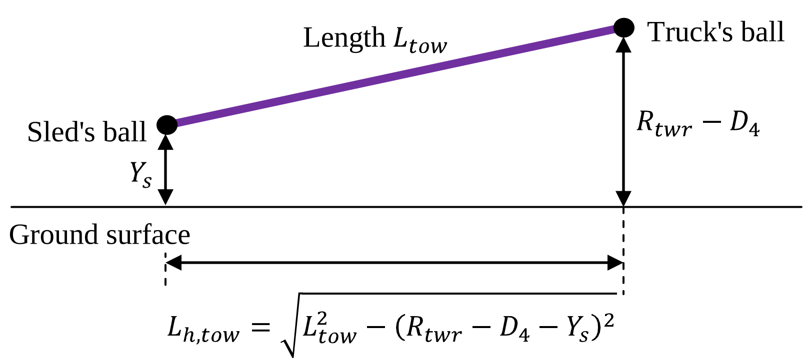
$$
\begin{aligned}
\frac{d^2 y_{twf}}{dt^2} &= \frac{d}{dt}\left\{\sqrt{\left(R_{twr}-R_{twf}\right)^2+W_t^2}\sin\left[\tan^{-1}\left(\frac{W_t}{R_{twr}-R_{twf}}\right)+\varphi_{tc}\right]\frac{d\varphi_{tc}}{dt}\right\} \\
&= \sqrt{\left(R_{twr}-R_{twf}\right)^2+W_t^2}\left\{\begin{aligned}&\cos\left[\tan^{-1}\left(\frac{W_t}{R_{twr}-R_{twf}}\right)+\varphi_{tc}\right]\left(\frac{d\varphi_{tc}}{dt}\right)^2+\cdots \\ &\cdots+\sin\left[\tan^{-1}\left(\frac{W_t}{R_{twr}-R_{twf}}\right)+\varphi_{tc}\right]\frac{d^2\varphi_{tc}}{dt^2}\end{aligned}\right\} \qquad (54B)
\end{aligned}
$$

Again comparing certain factors with the definitions of D_1 and D_2, it is apparent that:

$$
\begin{aligned}
\frac{d^2 x_{twf}}{dt^2} &= \frac{d^2 x_{tc}}{dt^2} - D_1\left(\frac{d\varphi_{tc}}{dt}\right)^2 + D_2\frac{d^2\varphi_{tc}}{dt^2} \qquad (55A) \\
\frac{d^2 y_{twf}}{dt^2} &= D_2\left(\frac{d\varphi_{tc}}{dt}\right)^2 + D_1\frac{d^2\varphi_{tc}}{dt^2} \qquad (55B)
\end{aligned}
$$

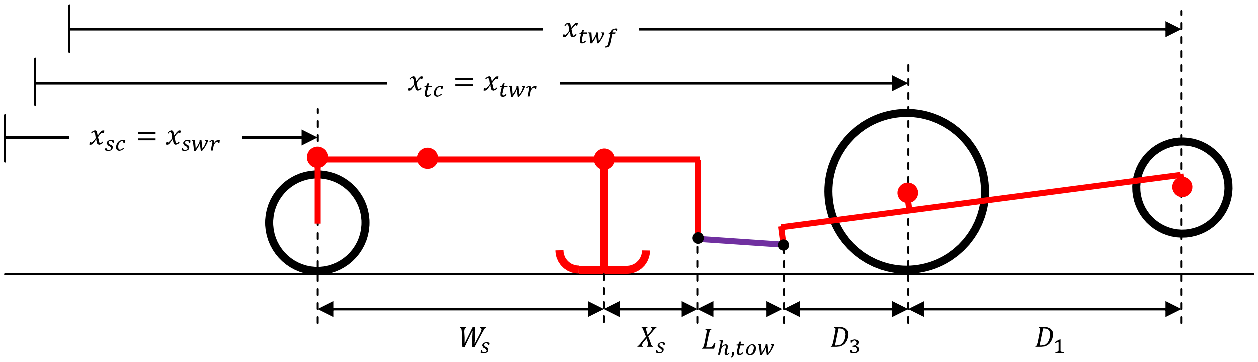
Note the symmetry between the two equations. When the truck is firmly on the ground, and the truck's rotation angle remains zero, the first expression says that the horizontal accelerations of the truck's front wheels and the truck's chassis are the same, and the second expression says that the vertical acceleration of the front wheels is zero. Both of these are as we expect.

Next, we need to relate the horizontal locations of the truck x_{tc} and the sled x_{sc}. They are connected by the tow chain. We will continue to assume, even in the enhanced model, that the tow chain has zero mass or, alternatively, that its mass is included in the masses used for the vehicles. Even so, the tow chain imposes another constraint on the rigid bodies in the rig. We will use the symbol L_{tow} for the nominal length of the tow chain. During normal operation, when the tow chain is taut, the relevant dimensions are the following:



Note that Y_t was defined above as the nominal height above ground of the truck's hitch ball. If the truck's chassis rotates, the height of the ball will decrease. Its instantaneous height is the quantity $R_{twr} - D_4$, where the distance D_4 was described above. Using the distance $R_{twr} - D_4$, the horizontal extent of the tow chain can be found using the Pythagorean Theorem, and is the radical shown in the figure. Of course, if the truck is not rotated, then the truck's ball is a height Y_t above ground and the radical reduces to $\sqrt{L_{tow}^2 - (Y_t - Y_s)^2}$.

We are interested in the horizontal component of the tow chain's length because it is the only dimensional variable which links the horizontal distances, speeds and accelerations of the truck and the sled. The following figure shows the five rigid bodies and the definitions of their instantaneous horizontal locations. In the figure, $L_{h,tow}$ is the horizontal component of the tow chain's length. The truck's chassis is shown with a bit of rotation, by angle φ_{tc}, which has actually caused the tow chain to angle downwards.



As we saw above, the distance variable for the sled's chassis is measured to the center of its rear axle, as is the distance variable for the sled's rear wheels, so $x_{sc} = x_{swr}$. Similarly, the distance variable for the truck is measured to the center of its rear axle, as is the distance variable for the truck's rear wheels, so $x_{tc} = x_{twr}$. The distance variable for the truck's front wheels x_{twf} is measured to the center of the front axle.

Each of the three distinct distance variables is measured from its location when the rig was back at the starting line. Just to be clear about this matter, it means that the distance travelled by each rigid body must be calculated by subtracting the starting distance from the current value.

Consider the location of the truck's chassis x_{tc}. It can be written in terms of the location of the sled's chassis x_{sc} as follows:

$$x_{tc} = x_{sc} + W_s + X_s + L_{h,tow} + D_3 \quad (56)$$

The horizontal speed of the truck's chassis is the first derivative of this expression with respect to time. Since W_s and X_s are constants, their derivatives are identically equal to zero and we are left with:

$$\frac{dx_{tc}}{dt} = \frac{dx_{sc}}{dt} + \frac{dL_{h,tow}}{dt} + \frac{dD_3}{dt} \quad (57)$$

Let's tackle the derivative of $L_{h,tow}$ first.

$$\begin{aligned}
L_{h,tow} &= \sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2} \\
\rightarrow \quad \frac{dL_{h,tow}}{dt} &= \frac{1}{2\sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2}} \frac{d}{dt}[L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2] \\
&= \frac{-2(R_{twr} - D_4 - Y_s)}{2\sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2}} \frac{d}{dt}(R_{twr} - D_4 - Y_s) \\
&= \frac{(R_{twr} - D_4 - Y_s)}{\sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2}} \frac{dD_4}{dt} \qquad (58)
\end{aligned}$$

It looks like we are going to need to find the derivative dD_4/dt and well as dD_3/dt. Proceeding first with D_3, we get:

$$
\begin{aligned}
D_3 &= \sqrt{(R_{twr}-Y_t)^2+X_t^2}\,\sin\left[\tan^{-1}\left(\frac{X_t}{R_{twr}-Y_t}\right)-\varphi_{tc}\right] \\
\rightarrow\quad \frac{dD_3}{dt} &= \sqrt{(R_{twr}-Y_t)^2+X_t^2}\,\frac{d}{dt}\sin\left[\tan^{-1}\left(\frac{X_t}{R_{twr}-Y_t}\right)-\varphi_{tc}\right] \\
&= -\sqrt{(R_{twr}-Y_t)^2+X_t^2}\,\cos\left[\tan^{-1}\left(\frac{X_t}{R_{twr}-Y_t}\right)-\varphi_{tc}\right]\frac{d\varphi_{tc}}{dt} \\
&= -D_4\frac{d\varphi_{tc}}{dt} \qquad (59)
\end{aligned}
$$

Similarly, dD_4/dt is calculated as:

$$
\begin{aligned}
D_4 &= \sqrt{(R_{twr}-Y_t)^2+X_t^2}\,\cos\left[\tan^{-1}\left(\frac{X_t}{R_{twr}-Y_t}\right)-\varphi_{tc}\right] \\
\rightarrow\quad \frac{dD_4}{dt} &= \sqrt{(R_{twr}-Y_t)^2+X_t^2}\,\sin\left[\tan^{-1}\left(\frac{X_t}{R_{twr}-Y_t}\right)-\varphi_{tc}\right]\frac{d\varphi_{tc}}{dt} \\
&= D_3\frac{d\varphi_{tc}}{dt} \qquad (60)
\end{aligned}
$$

Substituting Equations (58) through (60) into Equation (57) gives:

$$
\frac{dx_{tc}}{dt} = \frac{dx_{sc}}{dt}+\left[\frac{(R_{twr}-D_4-Y_s)D_3}{\sqrt{L_{tow}^2-(R_{twr}-D_4-Y_s)^2}}-D_4\right]\frac{d\varphi_{tc}}{dt} \qquad (61)
$$

The horizontal acceleration of the truck's chassis is the derivative of Equation (61) with respect to time.

$$
\begin{aligned}
\frac{d^2x_{tc}}{dt^2} &= \left\{
\begin{array}{c}
\dfrac{d^2x_{sc}}{dt^2}+\cdots \\
\cdots+\dfrac{d}{dt}\left[\dfrac{(R_{twr}-D_4-Y_s)D_3}{\sqrt{L_{tow}^2-(R_{twr}-D_4-Y_s)^2}}-D_4\right]\dfrac{d\varphi_{tc}}{dt}+\cdots \\
\cdots+\left[\dfrac{(R_{twr}-D_4-Y_s)D_3}{\sqrt{L_{tow}^2-(R_{twr}-D_4-Y_s)^2}}-D_4\right]\dfrac{d^2\varphi_{tc}}{dt^2}
\end{array}
\right\} \\
&= \left\{
\begin{array}{c}
\dfrac{d^2x_{sc}}{dt^2}+\cdots \\
\cdots+\left[
\begin{array}{c}
-\dfrac{D_3}{\sqrt{L_{tow}^2-(R_{twr}-D_4-Y_s)^2}}\dfrac{dD_4}{dt}-\dfrac{dD_4}{dt}+\cdots \\
\cdots+\dfrac{(R_{twr}-D_4-Y_s)}{\sqrt{L_{tow}^2-(R_{twr}-D_4-Y_s)^2}}\dfrac{dD_3}{dt}+\cdots \\
\cdots+\dfrac{-\frac{1}{2}(R_{twr}-D_4-Y_s)D_3}{\left[L_{tow}^2-(R_{twr}-D_4-Y_s)^2\right]^{3/2}}(-2)\left(-\dfrac{dD_4}{dt}\right)
\end{array}
\right]\dfrac{d\varphi_{tc}}{dt}+\cdots \\
\cdots+\left[\dfrac{(R_{twr}-D_4-Y_s)D_3}{\sqrt{L_{tow}^2-(R_{twr}-D_4-Y_s)^2}}-D_4\right]\dfrac{d^2\varphi_{tc}}{dt^2}
\end{array}
\right\}
\end{aligned}
$$

and, continuing:

$$\frac{d^2 x_{tc}}{dt^2} = \left\{ \begin{array}{c} \frac{d^2 x_{sc}}{dt^2} + \cdots \\ \cdots + \left[\begin{array}{l} -\frac{D_3}{\sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2}} \left(D_3 \frac{d\varphi_{tc}}{dt} \right) - \left(D_3 \frac{d\varphi_{tc}}{dt} \right) + \cdots \\ \cdots + \frac{(R_{twr} - D_4 - Y_s)}{\sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2}} \left(-D_4 \frac{d\varphi_{tc}}{dt} \right) + \cdots \\ \cdots + \frac{-\frac{1}{2}(R_{twr} - D_4 - Y_s) D_3}{[L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2]^{3/2}} (-2) \left(-D_3 \frac{d\varphi_{tc}}{dt} \right) \end{array} \right] \frac{d\varphi_{tc}}{dt} + \cdots \\ \cdots + \left[\frac{(R_{twr} - D_4 - Y_s) D_3}{\sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2}} - D_4 \right] \frac{d^2 \varphi_{tc}}{dt^2} \end{array} \right\}$$

$$= \left\{ \begin{array}{c} \frac{d^2 x_{sc}}{dt^2} + \cdots \\ \cdots - \left[\begin{array}{l} 1 + \frac{D_3^2 + (R_{twr} - D_4 - Y_s) D_4}{\sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2}} + \cdots \\ \cdots + \frac{(R_{twr} - D_4 - Y_s) D_3^2}{[L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2]^{3/2}} \end{array} \right] \left(\frac{d\varphi_{tc}}{dt} \right)^2 + \cdots \\ \cdots + \left[\frac{(R_{twr} - D_4 - Y_s) D_3}{\sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2}} - D_4 \right] \frac{d^2 \varphi_{tc}}{dt^2} \end{array} \right\} \quad (62)$$

I will not try to simplify this expression further since we are going to be computerizing the calculations anyway. Note that this expression has the same functional form as the acceleration of the front wheels. The horizontal acceleration of the truck's chassis is equal to the horizontal acceleration of the sled's chassis plus two types of terms. One type is proportional to the square of the first derivative of the truck's rotation angle. The other is proportional to the second derivative of the truck's rotation angle. This is a pretty common sight in dynamics and, for that matter, in second derivatives generally. We will write Equation (62) as follows:

$$\frac{d^2 x_{tc}}{dt^2} = \frac{d^2 x_{sc}}{dt^2} + f_1 \left(\frac{d\varphi_{tc}}{dt} \right)^2 + f_2 \frac{d^2 \varphi_{tc}}{dt^2} \quad (63)$$

where the functions $f_1(\varphi_{tc})$ and $f_2(\varphi_{tc})$ are functions only of the rotation angle φ_{tc} and are defined by their position in Equation (62).

Fortunately, this is as far as we need to take the relationships among the distances. We have the ones we need. We will substitute for all kinematic variables other than x_{sc} and φ_{tc}, leaving them as the only two independent variables needed to describe the rig's location.

We do, however, want an expression relating the horizontal and vertical components of the tension force in the tow chain. Since the tow chain is a chain, it can resist a force which tends to pull it apart, which one calls a "tension" force, but it cannot resist a compressive force. The tension force must act along the axis of the chain. Therefore, the angle of the forces on its two ends will be the same as the physical slope of the chain itself. From the diagram of the chain above, we can write the slope of the chain as follows:

$$\text{Slope of chain} = \frac{\text{Rise}}{\text{Run}} = \frac{R_{twr} - D_4 - Y_s}{\sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2}} \quad (64)$$

The ratio of the horizontal component of the tension to the vertical component must be the same.

$$\frac{T_{v,tow\to t}}{T_{h,tow\to t}} = \text{Slope of chain} \quad (65)$$

Combining the two expressions gives:

$$T_{v,tow\to t} = \frac{R_{twr} - D_4 - Y_s}{\sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2}} T_{h,tow\to t} \quad (66)$$

To simply the algebra, I propose to introduce the symbol Ю, which is the Cyrillic letter "yoo", as the factor of proportionality between the vertical and horizontal forces in the tow chain, thus:

$$T_{v,tow\to t} = \text{Ю} T_{h,tow\to t} \quad \text{where Ю} = \frac{R_{twr} - D_4 - Y_s}{\sqrt{L_{tow}^2 - (R_{twr} - D_4 - Y_s)^2}} \quad (67)$$

Although this is not an equation of motion *per se*, we will add Equation (67) to our set of equations.

Part XI -- The equations of motion revisited

We will start with the equations of motion in Equation (49). We will make the appropriate substitutions from the preceding section to remove references in all of the equations to the second derivatives d^2x_{twf}/dt^2, d^2y_{twf}/dt^2, d^2x_{twr}/dt^2, d^2x_{tc}/dt^2 and d^2x_{swr}/dt^2. In a couple of equations, two successive substitutions are needed. In addition, we will also add Equation (67) to the set. After all is said and done, we have the following 21 equations.

$$I_{twf} \frac{d^2\varphi_{twf}}{dt^2} = F_{h,g\to twf} R_{twf} \qquad (68C)$$

$$B_{twfAir} F_{h,g\to twf} = 0 \qquad (68D)$$

$$\left[B_{twfAir} + (1 - B_{twfAir}) B_{twfRTO}\right] F_{v,g\to twf} = 0 \qquad (68E)$$

$$(1 - B_{twfAir}) \begin{bmatrix} \dfrac{d^2x_{sc}}{dt^2} + (f_1 - D_1)\left(\dfrac{d\varphi_{tc}}{dt}\right)^2 + \cdots \\ \cdots + (f_2 + D_2)\dfrac{d^2\varphi_{tc}}{dt^2} - R_{twf}\dfrac{d^2\varphi_{twf}}{dt^2} \end{bmatrix} = 0 \qquad (68F)$$

$$(1 - B_{twfAir})(1 - B_{twfRTO}) \left[D_2 \left(\frac{d\varphi_{tc}}{dt}\right)^2 + D_1 \frac{d^2\varphi_{tc}}{dt^2}\right] = 0 \qquad (68G)$$

$$I_{swr} \frac{d^2\varphi_{swr}}{dt^2} = F_{h,g\to swr} R_{swr} \qquad (68J)$$

$$\frac{d^2x_{sc}}{dt^2} = R_{swr} \frac{d^2\varphi_{swr}}{dt^2} \qquad (68K)$$

$$I_{twr} \frac{d^2\varphi_{twr}}{dt^2} = \tau_{c\to twr} - F_{h,g\to twr} R_{twr} \qquad (68N)$$

$$B_{twrNoSlip} \left[\frac{d^2x_{sc}}{dt^2} + f_1 \left(\frac{d\varphi_{tc}}{dt}\right)^2 + f_2 \frac{d^2\varphi_{tc}}{dt^2} - R_{twr} \frac{d^2\varphi_{twr}}{dt^2}\right] = 0 \quad (68O)$$

$$\left(1 - B_{twrNoSlip}\right)\left(F_{h,g\to twr} - C_{twr,d} F_{v,g\to twr}\right) = 0 \qquad (68P)$$

$$\begin{bmatrix} M_{truck}\dfrac{d^2x_{sc}}{dt^2}+\cdots \\ \cdots+\left(M_{truck}f_1-M_{twf}D_1\right)\left(\dfrac{d\varphi_{tc}}{dt}\right)^2+\cdots \\ \cdots+\left(M_{truck}f_2+M_{twf}D_2\right)\dfrac{d^2\varphi_{tc}}{dt^2} \end{bmatrix}=-F_{h,g\to twf}+F_{h,g\to twr}-T_{h,tow\to t} \tag{68R}$$

$$M_{twf}\left[D_2\left(\frac{d\varphi_{tc}}{dt}\right)^2+D_1\frac{d^2\varphi_{tc}}{dt^2}\right]=F_{v,g\to twf}+F_{v,g\to twr}-T_{v,tow\to t}-M_{truck}g \tag{68S}$$

$$\begin{Bmatrix} M_{twf}D_2\dfrac{d^2x_{sc}}{dt^2}+\cdots \\ \cdots+M_{twf}f_1D_2\left(\dfrac{d\varphi_{tc}}{dt}\right)^2+\cdots \\ \cdots+\left[I_{tc}+M_{twf}(D_1^2+f_2D_2+D_2^2)\right]\dfrac{d^2\varphi_{tc}}{dt^2} \end{Bmatrix}=\begin{bmatrix} \tau_{c\to twr}+\cdots \\ F_{v,g\to twf}D_1-F_{h,g\to twf}D_2+\cdots \\ \cdots+T_{v,tow\to t}D_3-T_{h,tow\to t}D_4+\cdots \\ \cdots-\left(M_{twf}+M_{tcf}\right)D_1g \end{bmatrix} \tag{68T}$$

$$M_{sled}\frac{d^2x_{sc}}{dt^2}=T_{h,tow\to t}-D_{g\to sp}-F_{h,g\to swr} \tag{68U}$$

$$0=T_{v,tow\to t}+F_{v,g\to sp}+F_{v,g\to swr}-M_{sled}g \tag{68V}$$

$$0=\begin{bmatrix} T_{v,tow\to t}(W_s+X_s)+T_{h,tow\to t}(R_{swr}-Y_s)+\cdots \\ \cdots+\left(F_{v,g\to sp}-M_{scf}g\right)W_s-D_{g\to sp}R_{swr}-M_{sm}gD_w \end{bmatrix} \tag{68W}$$

$$B_{SledMove}\left(D_{g\to sp}-C_{pd}F_{v,g\to sp}\right)=0 \tag{68X}$$

$$(1-B_{SledMove})\frac{d^2x_{sc}}{dt^2}=0 \tag{68Y}$$

$$T_{v,tow\to t}=\text{Ю}T_{h,tow\to t} \tag{68AA}$$

$$B_{twrNoSlip}\left(F_{h,g\to twr}\le C_{twr,s}F_{v,g\to twr}\right) \tag{68Q}$$

$$(1-B_{SledMove})\left(D_{g\to sp}\le C_{ps}F_{v,g\to sp}\right) \tag{68Z}$$

I am going to re-order the equations in anticipation of writing them in matrix format as a prelude to inversion. At the same time, I am going to make an ordered list of the unknowns. My intention is to populate the main diagonal of the matrix with as many non-zero entries as possible. Planning the first dozen or so entries in the lists is pretty straight forward.

Matrix row #	Equation	Unknown	Value of A(i, i)
1	$(68R)$	$\frac{d^2x_{sc}}{dt^2}$	M_{truck}
2	$(68T)$	$\frac{d^2\varphi_{tc}}{dt^2}$	$I_{tc}+M_{twf}(D_1^2+f_2D_2+D_2^2)$
3	$(68C)$	$\frac{d^2\varphi_{twf}}{dt^2}$	I_{twf}
4	$(68K)$	$\frac{d^2\varphi_{swr}}{dt^2}$	R_{swr}
5	$(68N)$	$\frac{d^2\varphi_{twr}}{dt^2}$	I_{twr}
6	$(68W)$	$F_{v,g\to sp}$	W_s

7	$(68AA)$	$T_{v,tow\to t}$	1
8	$(68U)$	$T_{h,tow\to t}$	1
9	$(68S)$	$F_{v,g\to twr}$	1
10	$(68V)$	$F_{v,g\to swr}$	R_{swr}
11	$(68J)$	$F_{h,g\to swr}$	R_{swr}

After this, things get less deterministic. Most of the remaining equations involve one or more binary flags so there is less assurance that some or all of the coefficients will be non-zero when it comes time to invert the matrix. I have chosen to complete the ordering of equations and unknowns as follows.

Matrix row #	Equation	Unknown	Value of A(i, i)
12	$(68X)$	$D_{g\to sp}$	$B_{SledMove}$
13	$(68P)$	$F_{h,g\to twr}$	$1 - B_{twrNoSlip}$
14	$(68D)$	$F_{h,g\to twf}$	B_{twfAir}
15	$(68E)$	$F_{v,g\to twf}$	$B_{twfAir} + (1 - B_{twfAir})B_{twfRTO}$
16	$(68F)$		n/a
17	$(68G)$		n/a
18	$(68O)$		n/a
19	$(68Y)$		n/a

The 19 rows in the table do not include the two inequalities. We will test the inequalities only after we have found a solution. We can write the 19 equations in the form of the following matrix equation:

$$\begin{gathered}\tilde{A} \cdot \tilde{U} = \tilde{B}, \\ \text{which has the dimensions} \\ [19 \text{ rows} \times 15 \text{ columns}][15 \text{ unknowns}] = [19 \text{ "constants"}]\end{gathered} \tag{69}$$

Vector $\tilde{U}$ is the 15×1 list of the unknwons in the order set out in the third column of the table. The "constant" vector $\tilde{B}$ is the 19×1 list of all the terms in the equations which are not coefficients of the unknowns. Any time we solve the equation, all of the terms in vector $\tilde{B}$ will be known. Each row in the main matrix $\tilde{A}$ corresponds to one of the equations; each column corresponds to one of the unknowns. I ordered the equations and variables in such a way that we are guaranteed that the elements on the principal diagonal are non-zero down to the eleventh row. That is, $\tilde{A}_{ii}$ is non-zero for $i = 1, 2 \cdots 11$.

A unique solution can be obtained only if the number of non-zero rows and columns in matrix $\tilde{A}$ are the same. To ensure that this will always be the case, we need to look at rows 12 through 19 of matrix $\tilde{A}$. The following figure is a block diagram of those rows. The elements of the matrix are not given in full. Only their dependence on the binary flags is shown. To keep the size of the diagram reasonable, I have used the symbol $\bar{B}_x$ for the complement of binary flag B_x. In addition, I have excluded those unknowns (columns) which are not referenced by the last eight equations.

	$\frac{d^2x_{sc}}{dt^2}$	$\frac{d^2\varphi_{tc}}{dt^2}$	$\frac{d^2\varphi_{twf}}{dt^2}$	$\frac{d^2\varphi_{twr}}{dt^2}$	$F_{v,g\to sp}$	$F_{v,g\to twr}$	$D_{g\to sp}$	$F_{h,g\to twr}$	$F_{h,g\to twf}$	$F_{v,g\to twf}$
$(68X)$	0	0	0	0	$B_{SledMove}$	0	$B_{SledMove}$	0	0	0
$(68P)$	0	0	0	0	0	$\bar{B}_{twrNoSlip}$	0	$\bar{B}_{twrNoSlip}$	0	0
$(68D)$	0	0	0	0	0	0	0	0	B_{twfAir}	0
$(68E)$	0	0	0	0	0	0	0	0	0	$\left(\begin{array}{c} B_{twfAir} + \\ \bar{B}_{twfAir}B_{twfRTO} \end{array}\right)$
$(68F)$	$\bar{B}_{twfAir}$	$\bar{B}_{twfAir}$	$\bar{B}_{twfAir}$	0	0	0	0	0	0	0
$(68G)$	0	$\bar{B}_{twfAir}\bar{B}_{twfRTO}$	0	0	0	0	0	0	0	0
$(68O)$	$B_{twrNoSlip}$	$B_{twrNoSlip}$	0	$B_{twrNoSlip}$	0	0	0	0	0	0
$(68Y)$	$\bar{B}_{SledMove}$	0	0	0	0	0	0	0	0	0

The first and last rows, corresponding to Equations $(68X)$ and $(68Y)$, respectively, are complemenatry. Depending on the setting of the flag $B_{SledMove}$, exactly one of these two equations will vanish.

Similarly, the second and second-to-last rows, corresponding to Equations $(68P)$ and $(68O)$, respectively, are also complementary. Depending on the setting of $B_{twrNoSlip}$, exactly one of these two equations will vanish.

The third and fourth-from-last equations, corresponding to Equations $(68D)$ and $(68F)$, respectively, are complementary. Depending on the setting of B_{twfAir}, exactly one of these two equations will vanish.

That leaves Equations $(68E)$ and $(68G)$, which are related to the front wheels (and not the other wheels) and also on the whether or not the front wheels are on the verge of going airborne. These two equations do not look complementary, but they are. See the following truth table.

$\boldsymbol{B_{twfAir}}$	$\boldsymbol{B_{twfTO}}$	**Coefficient in $(\boldsymbol{68E})$**	**Coefficient in $(\boldsymbol{68G})$**
1	1	$0 \cdot 0 = 0$	$1 + 0 \cdot 1 = 1$
1	0	$0 \cdot 1 = 0$	$1 + 0 \cdot 0 = 1$
0	1	$1 \cdot 0 = 0$	$0 + 1 \cdot 1 = 1$
0	0	$1 \cdot 1 = 1$	$0 + 1 \cdot 0 = 0$

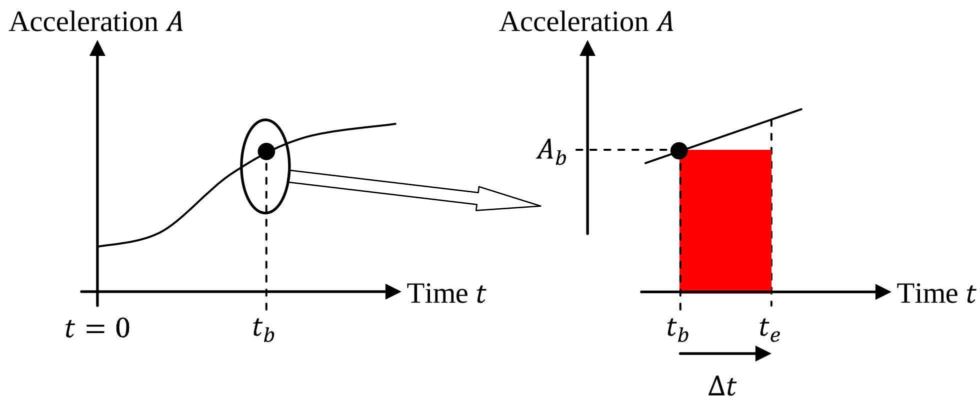
It turns out, then, that the last eight rows of matrix $\tilde{A}$ will always reduce to exactly four non-trivial equations. Along with the first 11 rows, there will always be exactly 15 equations. There are 15 unknowns, so we shoudl always be able to get a solution.

Ordering the equations and unknowns so that the principal diagonal of matrix $\tilde{A}$ has as few zero elements as possible is not a mathematical necessity. But I did have a reason to do so. Inverting a 15×15 matrix which represents non-trivial and linearly independent equations is always possible. That is not to say that it is easy or that, if done by computer, can be done quickly. We are going to be solving the equations of motion several hundred thousand times per real-world second, so we do not want to waste processing time.

Our matrix $\tilde{A}$ has a characteristic that I would like to take advantage of. It is very sparse. Although we have a lot of equations and a lot of unknowns, each equation only refers to a couple of the unknowns. Most of the elements in matrix $\tilde{A}$ are zero. There are numerical methods to invert matrices which have been optimized for sparse matrices, but I do not have such a routine at hand. I will use Eulerian elimination instead. When a matrix is sparse, and has a healthy diagonal, Eulerian substitution can run pretty quickly. It can be particularly good/fast if the ALU does a pre-check for multiplication by zero.

Part XII -- Numerical integration

As I explained in the earlier paper, adding up the area below the curve in an acceleration-versus-time graph gives the speed. Adding up the area below the curve in a speed-versus-time graph gives the distance travelled. When the Calculus is used, adding up the area below a curve is called "integrating". But that is not the only way areas can be added up. Another way, not much used in recent times, is to draw the curve on heavy paper, use scissors to cut out the areas corresponding to different times, and then weigh the pieces on a scale. A third way, the mainstay of computerized calculations, is to divide the area up into little rectangles, each of whose area can be found using the $base \times height$ formula, and to add up the little bits. The following figure illustrates how this can be done.



The curve on the left is a typical plot of acceleration A versus time t. The area under this curve from time $t = 0$ to some arbitrary time t_b is the change in the object's speed since time $t = 0$. A very small section of the curve is shown in detail on the right. We are going to consider an extremely short period of time, with duration Δt. Times t_b and t_e are the times at which the interval begins and ends. If we know the object's speed S_b and its acceleration A_b at the beginning of the interval, then we can approximate its speed at the end of the interval as:

$$S_e \cong S_b + A_b \Delta t \quad (70)$$

The product $A_b \Delta t$ is the area of the red rectangle shown on the right. The area of the red rectangle differs from the true area under the curve by the small right triangle above the rectangle. There are lots of methods which are sometimes used to increase the accuracy of the approximation, such as adding a term based on the assumption that the change in the acceleration over the preceding small increment of time continues during this one. It should be noted, though, that the missing area can be made arbitrarily small by shrinking the length of the time-interval. Shrinking the length of the time-intervals, or "time steps", does increase the number of such steps and the amount of labour needed to do the calculations.

As described in the earlier paper, the relationship between the object's speed and the distance it travels are related in exactly the same way. The area under a speed-versus-time curve from time $t = 0$ to some arbitrary time t_b is the change in the object's location since time $t = 0$. We can update the estimate of the object's position D during a time step using the following equation:

$$D_e \cong D_b + S_b \Delta t + \tfrac{1}{2} A_b \Delta t^2 \quad (71)$$

Like the estimate for speed, this expression becomes exact in the limit as we make the duration of the time steps vanishingly small.

Here is how we will use these expressions.

1. Let's pick a time step which is 100 microseconds, or 0.0001 seconds, long. A truck pull which lasts 20 seconds will require $20/0.0001 = 200{,}000$ time steps. (A quick and dirty method to determine if the time steps are short enough is to re-run one or more cases using a different time step and compare the results.)

2. We will make a note of the location and speed of each of the five rigid bodies at the start of the run, which we will say happens at time $t = 0$. Note that "location and speed" actually refers to six quantities for each rigid body. Each body has two directions of motion and a direction of rotation, plus corresponding changes, or speeds, for those three directions.

3. As we move along from one time step to the next, we will keep track of the updated values for the three location variables and three corresponding speeds of each rigid body. It follows that, at the start of any particular time step, we will know the five bodies' current locations and speeds.

4. With the locations and speeds at time t_b known, we will solve the equations of motion to calculate the current accelerations. There will be three accelerations for each rigid body, one for each direction of travel and one for rotation.

5. With the accelerations at time t_b now at hand, we will use Equations (70) and (71) to calculate the locations and speeds at the end of the time step, at time t_e. The relationships for speed and distance in these two equations are not restricted to systems which involve a single variable. They apply independently to each to each kinematic variable in any physical system.

6. We are now set up to begin processing the next time step, whose beginning time will be equal to the ending time t_e of the previous time step.

I want to expand the descriptions I gave in steps 4 and 5 to be more particular about the implementation of these steps in the attached computer code. The matrix equations we developed above do not include accelerations in all of the kinematic variables. Certain geometric relationships allowed us to express d^2x_{twf}/dt^2, d^2y_{twf}/dt^2, d^2x_{twr}/dt^2, d^2x_{tc}/dt^2 and d^2x_{swr}/dt^2 in terms of d^2x_{sc}/dt^2 and $d^2\varphi_{tc}/dt^2$. It follows that, when we solve the equations of motion by inverting matrix $\tilde{A}$, the only accelerations whose values we will have are those for the sled's horizontal location x_{sc}, the truck's rotation angle φ_{tc} and the rotational acceleration of the three sets of wheels. Philosophically, there are two ways to proceed.

One way would be to use the geometric relationships to calculate the accelerations for all ten kinematic variables. Each acceleration could be integrated separately using Equations (70) and (71) to find the corresponding speed and position.

The other way is to limit the integration to the five kinematic variables whose accelerations are produced by the matrix inversion. The speeds and locations of the other kinematic variables would be inferred from the changes in the basic five. For example, we know that the change in the location of the sled's rear wheels x_{swr} during any time step must be exactly equal to the change in the location of sled's chassis x_{sc}.

Mathematically, the two ways give identical answers. When the equations are discretized for computerization solution, small differences arise, which grow as the integration continues. After a large number of time steps, for example, one could find that the rear wheels of the sled have travelled a slightly different distance than the sled. These inconsistencies can be eliminated by ensuring that the kinematic variables are always updated in a consistent way.

At the end of every time step, the updated speeds and locations need to be reviewed for realism. Consider this example. At the start of the time step, the truck's front wheels are slightly above the ground, but headed downwards. It is possible that, at the end of the time step, the vertical location of the front wheels y_{twf} turns out to be less than the radius of the front wheels R_{twf}. This is not possible physically -- the wheels do not go into the ground. It occurs during numerical integration because we assume that the rig remains in the same state for the entire duration of a time step. At the end of every time step, the code checks for such physical imperfections and rounds them to zero.

Part XIII -- The transition conditions

At any instant in time, the rig can be in one of eight states, or cases. I will henceforth refer to the eight states, or cases, as follows:

Case ID	Sled moving or stopped	Truck's front wheels	Truck's rear wheels
1	Moving	On ground	Not slipping
2	Moving	On ground	Slipping
3	Moving	Airborne	Not slipping
4	Moving	Airborne	Slipping
5	Stopped	On ground	Not slipping
6	Stopped	On ground	Slipping
7	Stopped	Airborne	Not slipping
8	Stopped	Airborne	Slipping

It is pretty obvious what it means for the rig to be running in one of these states.

State	Mathematical condition	
Sled is moving	$\frac{dx_{sc}}{dt} > 0$	$(72A)$
Sled is stopped	$\frac{dx_{sc}}{dt} = 0$	$(72B)$
Front wheels on ground	$y_{twf} = R_{twf}$	$(72C)$
Front wheels airborne	$y_{twf} > R_{twf}$	$(72D)$
Rear wheels gripping	$R_{twr}\frac{d\varphi_{twr}}{dt} = \frac{dx_{twr}}{dt}$	$(72E)$
Rear wheels slipping	$R_{twr}\frac{d\varphi_{twr}}{dt} > \frac{dx_{twr}}{dt}$	$(72F)$

These conditions are stated in terms of speeds and locations only. They do not involve accelerations. This is important. It is the main reason why I placed this section after the one that describes the numerical integration procedure. At the end of each time step, we will know the values and speeds of the kinematic variables. We can therefore determine what state the rig is in at this instant.

What is of interest to us at the start of each time step is what state the rig is going to be in during the upcoming time step. Most often, it will continue forward in whatever state it has been in. But, not always. Sometimes, the rig will make a transition to a new state. We need to figure out what state the rig is going to be in during the upcoming time step so that we can use the correct equations of motion to calculate the correct accelerations. In short, we need to figure out when each of the foregoing six conditions has come to an end or is expected to come to an end during the upcoming time step. I am going to refer to these as "stopping" conditions. Let's examine them on a one-by-one basis.

<u>When the front wheels are airborne at the end of a time step</u>

If the front wheels are airborne, so that $y_{twf} > R_{twf}$, they will continue to be airborne until they come back into contact with the ground. We can get a rough idea about when the wheels will hit the ground by dividing their current height above ground, $y_{twf} - R_{twf}$, by their current speed dy_{twf}/dt. Of course, this test only makes sense if the speed is algebraically negative, so the wheels are heading down towards the ground. We will compare this estimated time to the length of a time step. If the front wheels will encounter the ground less than one-half time step away, then we will assume that the front wheels are on the ground during the whole of the time step. If the front wheels are more than one-half time step away from the ground, then we will assume that they are airborne throughout the time step. If we use the symbol ΔT as the duration of a time step, then we can express this test as follows:

Conditions at start of time step, with $\boldsymbol{y_{twf} > R_{twf}}$	**Expected state during time step**	**Flag settings**
$\frac{dy_{twf}}{dt} \geq 0$	Front wheels will be airborne	$B_{twfAir} = 1$
$\frac{dy_{twf}}{dt} < 0$ and $\frac{y_{twf} - R_{twf}}{\lvert dy_{twf}/dt \rvert} \leq \frac{1}{2}\Delta T$	Front wheels will be on ground	$B_{twfAir} = 0$ $B_{twfRTO} = 0$
$\frac{dy_{twf}}{dt} < 0$ and $\frac{y_{twf} - R_{twf}}{\lvert dy_{twf}/dt \rvert} > \frac{1}{2}\Delta T$	Front wheels will be airborne	$B_{twfAir} = 1$

<u>When the front wheels are on the ground at the end of a time step</u>

When the front wheels are on the ground, the only way (physically) they can become airborne is if their vertical acceleration is non-zero and positive for at least an instant. The analysis of the front wheels showed there is a connection between the vertical acceleration of the front wheels and the vertical force between the wheels and the ground. Generally speaking, either one or the other must be zero. In order to deal with the special instant when the front wheels on the ground but ready to take-off, we introduced the binary flag B_{twfRTO}.

To predict what is going to happen to the front wheels, we need to look at what is happening to the vertical force of the ground. If the vertical force is zero, or expected to reach zero during the upcoming time step, we should use the airborne equations, which we do by setting $B_{twfRTO} = 1$. As before, let's use the mid-point of the time step as the decision point.

One slight difficulty, but easily overcome, is that the equations of motion do not explicitly include the time-derivative of the vertical force on the front wheels. A first-order estimate can be made if we temporarily store the value of the vertical force at the beginning of each time time. Then, at the end of the time step, we can make a subtraction to compute the change which occurred in the vertical force. If we use the symbol $F_{v,g\rightarrow twf}$ as the current value of the vertical force (as always) and the symbol $F_{v,g\rightarrow twf,last}$

for the vertical force at the start of the previous time step, then we can approximate the time-rate-of-change of the vertical force as follows:

$$\frac{\Delta F_{v,g\to twf}}{\Delta t} \cong \frac{F_{v,g\to twf} - F_{v,g\to twf,last}}{\Delta T} \qquad (73)$$

Dividing the value of the vertical force at the end of a time step by this rate-of-change will give the approximate length of time before the vertical force reaches zero. The following table sets out the flowchart for deciding which equations of motion to solve.

Conditions at start of time step, with $y_{twf} = R_{twf}$	**Expected state during time step**	**Flag settings**
$\frac{\Delta F_{v,g\to twf}}{\Delta t} \geq 0$	Front wheels will be on ground	$B_{twfAir} = 0$ $B_{twfRTO} = 0$
$\frac{\Delta F_{v,g\to twf}}{\Delta t} < 0$ and $\frac{F_{v,g\to twf}}{\left\lvert\Delta F_{v,g\to twf}/\Delta t\right\rvert} \leq \frac{1}{2}\Delta T$	Front wheels will be airborne	$B_{twfAir} = 0$ $B_{twfRTO} = 1$
$\frac{\Delta F_{v,g\to twf}}{\Delta t} < 0$ and $\frac{F_{v,g\to twf}}{\left\lvert\Delta F_{v,g\to twf}/\Delta t\right\rvert} > \frac{1}{2}\Delta T$	Front wheels will be on ground	$B_{twfAir} = 0$ $B_{twfRTO} = 0$

<u>When the sled is moving at the end of a time step</u>

If the sled is moving, it will continue to move until it comes to a stop. This is a kinematic test of the same ilk as the front wheels being airborne. If the sled is decelerating, we can estimate the time until it will come to a stop by dividing its current speed by the rate of deceleration. It happens that the sled's horizontal acceleration is included in the equations of motion, so we do not need to save former speeds to approximate the rate of change. We will use the following flowchart

Conditions at start of time step, with $\frac{dx_{sc}}{dt} > 0$	**Expected state during time step**	**Flag settings**
$\frac{d^2x_{sc}}{dt^2} \geq 0$	Sled will be moving	$B_{SledMove} = 1$
$\frac{d^2x_{sc}}{dt^2} < 0$ and $\frac{dx_{sc}/dt}{\lvert d^2x_{sc}/dt^2\rvert} \leq \frac{1}{2}\Delta T$	Sled will be stopped	$B_{SledMove} = 0$
$\frac{d^2x_{sc}}{dt^2} < 0$ and $\frac{dx_{sc}/dt}{\lvert d^2x_{sc}/dt^2\rvert} > \frac{1}{2}\Delta T$	Sled will be moving	$B_{SledMove} = 1$

Let me describe the physical ramifications of using the second test. The condition is met if the sled is decelerating fast enough that it will come to a stop very shortly. Since it is going to be stopped very soon, we will solve the equations of motion assuming it is already stopped. We are, in effect, jolting the sled to a stop. The jolt will never be more than one-half step. We can reduce the effect of the jolt as much as we want by making the time steps shorter and shorter. Note that the same kind of jolt will occur even when we use the third test. If it happens that the sled's speed reaches zero during the second half of the time step, the deceleration will continue for the rest of the time step, with the result that the sled will have a very small negative speed at the end of the time step. We will round that negative speed back up to zero, in effect jolting the sled to a stop for the start of the following time step. A similar kind of jolt occurs when the front wheels of the truck are airborne and come back into contact with the ground. These are not physical jolts, but a natural result of discretizing time.

When the sled is stopped at the end of a time step

If the sled is stopped, the only kinematic variable which can get it moving is its horizontal acceleration. We did not notice any problems above when solving the equations of motion for the sled's horizontal acceleration. Therefore, when the sled is stopped, we should be able to set the flag $B_{SledMove} = 0$ and then solve the equations. But, that is not the end of it. The sled will remain stopped only if the frictional force the ground exerts on the pan is less than the critical maximum force. We need to compare the value of $D_{g\to sp}$ with the maximum possible value, $C_{ps}F_{v,g\to sp}$. This is the inequality in Equation $(68Z)$ which, being an inequality, was not included in the matrix equations *per se*. If the former exceeds the latter, the sled will be jostled into motion. The flowchart is as follows:

Conditions at start of time step, with $\frac{dx_{sc}}{dt} = 0$	**Expected state during time step**	**Flag settings**
$D_{g\to sp} > C_{ps}F_{v,g\to sp}$	Sled will be moving	$B_{SledMove} = 1$
$D_{g\to sp} \leq C_{ps}F_{v,g\to sp}$	Sled will be stopped	$B_{SledMove} = 0$

Alert readers may notice a small delay inherent in this test. When we apply the test, the value of the ground drag $D_{g\to sp}$ is the one calculated when the equations of motion were solved at the start of he previous time step. We are therefore basing our test on a force which is one time step delayed. We could do better. We could do a trial solution of the equations of motion to update the value of $D_{g\to sp}$ and decide whether the sled will be moving or stopped. That would give a slightly better decision but would require that the equations of motion be solved twice at the start of every time step. An alternative approach would be to look at how fast $D_{g\to sp}$ has been changing, and to make a prediction about whether it will pass through the critical value within the next half-time step. That is what we did in the tests above. This test is a little different from the previous ones because the critical value is not constant but is itself changing. I have decided not to bother at all with prediction.

When the rear wheels are not slipping at the end of a time step

Slippage of the truck's rear wheels is a friction-driven condition like that which governs the sled. In this case, the variable to test is the horizontal force of the ground on the rear wheels $F_{h,g\to twr}$ and the maximum frictional force permitted is $C_{twr,s}F_{v,g\to twr}$. If the former exceeds the latter, the rear wheels will be jostled into slipping. The flowchart is as follows:

Conditions at start of time step, with $R_{twr}\frac{d\varphi_{twr}}{dt} = \frac{dx_{twr}}{dt}$	**Expected state during time step**	**Flag settings**
$F_{h,g\to twr} > C_{twr,s}F_{v,g\to twr}$	Rear wheels will be slipping	$B_{twrNoSlip} = 0$
$F_{h,g\to twr} \leq C_{twr,s}F_{v,g\to twr}$	Rear wheels will not be slipping	$B_{twrNoSlip} = 1$

Just like the test for starting up the sled, this test involves a one time step delay.

When the rear wheels are slipping at the end of a time step

I suspect this transition is rarely seen during a truck pull. It seems to me that the run is pretty much over once the wheels start slipping. Or, alternatively, that the run must continue with the wheels slipping. It is possible to keep moving, and perhaps complete the course, with slipping wheels. Once the wheels are slipping, their rotational speed must be reduced substantially before they will again grip the ground. Think about how slipping starts. The tread's horizontal force must exceed the static friction of the ground. As soon as slipping starts, he wheels' rotation is resisted by the dynamic friction of the ground, which will always be less than the static friction. Before their rotational speed will have decreased to the point where the wheels grip the ground, two unrelated things will have happened: (i) the moveable weight will have advanced further towards the front of the sled and (ii) the whole rig will have slowed down. If the wheels started slipping before, they are even more likely to slip under these two more adverse conditions. In any event, it is prudent to give the program the capability to deal with this kind of transition.

The rear wheels will stop slipping, by definition, when the tread speed decreases enough to equal the horizontal forward speed of the wheels. This is a speed-related test, like the sled continuing in motion, and we can handle it in the same way. Here, the speed we are interested in is the relative speed between the tread and the ground. We can calculate the time rate-of-change of the relative speed RS as follows:

$$RS \triangleq R_{twr}\frac{d\varphi_{twr}}{dt}-\frac{dx_{twr}}{dt} \qquad (74A)$$
$$\frac{dRS}{dt} = R_{twr}\frac{d^2\varphi_{twr}}{dt^2}-\frac{d^2x_{twr}}{dt^2} \qquad (74B)$$

and the decision tree is the following flowchart:

Conditions at start of time step, with $R_{twr}\frac{d\varphi_{twr}}{dt} > \frac{dx_{twr}}{dt}$	**Expected state during time step**	**Flag settings**
$\frac{dRS}{dt} \geq 0$	Rear wheels will be slipping	$B_{twrNoSlip} = 0$
$\frac{dRS}{dt} < 0$ and $\frac{RS}{\lvert dRS/dt\rvert} \leq \frac{1}{2}\Delta T$	Rear wheels will not be slipping	$B_{twrNoSlip} = 1$
$\frac{dRS}{dt} < 0$ and $\frac{RS}{\lvert dRS/dt\rvert} > \frac{1}{2}\Delta T$	Rear wheels will be slipping	$B_{twrNoSlip} = 0$

State at the start of the truck pull

The six transition conditions apply once the truck pull gets underway. Some of them rely on the rig's recent trajectory. At the start of a run, there is no such history, and one needs to make a guess about the starting state. A sensible default state has the sled at rest, the truck's front wheels on the ground and the truck's rear wheels no slipping. When the equations of motion are solved assuming that state, however, the tests as described above fail to detect certain states. For example, the equations of motion may produce $F_{v,g\rightarrow twf}$ less than zero. This means the front wheels must be airborne, even during the first time step. If this is not corrected, the second time step will produce an even more negative vertical ground force. The related test may never put the wheels mathematically back on the ground. The remedy is to detect forces which are not physically realistic and reset them to realistic values, or even zero. Subsequent solutions of the equations of motion will then converge to the realistic trajectory.

Part IX -- Simulation results

In the earlier paper, we looked at a stock 1997 Ford F-150 SuperCab two wheel drive pickup truck and a sled in "Hillbilly" mode. I carried out a numerical simulation of that rig based on the equations of motion developed above. The results given below are the distance covered during the run ("Distance") and the time taken ("Time").

Benchmarking against the solution from the simplified model

The place to start, of course, is to run a simulation based on the same assumptions as used in the earlier paper. To the extent possible, all parameters were set to the same values as before. For example, the wheels' radii, sled's deck height and the two hitch heights were all set to the same value, 15¾ inches. There was one exception. The enhanced model assumes that each set of wheels has a moment of inertia. Some of the equations of motion degenerate when the moments of inertia are set identically equal to zero. (Think of trying to apply Newton's Law to an object having zero mass.) In order to use the equations of motion without any modification, I set the moments of inertia to a very small but non-zero, value: 0.00001 pound-feet-squared. The results were:

$$\text{Distance} = 179.2\text{ feet} \quad \text{Time} = 19.00\text{ seconds} \quad (75)$$

The result in the earlier paper was:

$$\text{Distance} = 190.3\text{ feet} \quad \text{Time} = 19.14\text{ seconds}$$

After some checking, I believe the difference arises because we restricted the interim calculations in the earlier paper to three significant digits. The numerical simulation was done by computer in double precision. The difference is not caused by the non-zero moments of inertia, which would tend to reduce the distance run. Changing the moments of inertia to even smaller values, first to 1×10^{-7} and then to 1×10^{-9}, did not the results in Equation (75).

Sensitivity to the length of each time step

The simulation which produced Equation (75) was run using a time step of 10 microseconds. The simulation was re-run using various time steps ΔT, with the following results:

ΔT	Distance	Time	
$5\ \mu\text{s} = 0.000005$ sec	179.2 feet	19.00 seconds	$(76A)$
$10\ \mu\text{s} = 0.00001$ sec	179.2 feet	19.00 seconds	$(76B)$
$100\ \mu\text{s} = 0.0001$ sec	179.2 feet	19.00 seconds	$(76C)$
$1\text{ ms} = 0.001$ sec	179.2 feet	19.00 seconds	$(76D)$
$10\text{ ms} = 0.01$ sec	178.9 feet	18.99 seconds	$(76E)$

The simulation dos not begin to show any effects from lengthening the time step until it becomes greater than one millisecond, which corresponds to 1,000 time steps per second. Just to be on the safe side, the simulations whose results are shown below were all carried out with a time step ten times smaller than this, or 100 microseconds, which corresponds to 10,000 time steps per second.

An aside:

> I had initially thought that the simulation would require a time step of 10 microseconds. It turns out that the physical events take place on a much longer time scale than this. In any event, I had some concern that the computations would take too long for the results to be displayed in real time. I planned on taking two steps to reduce execution time. I ended up implementing the first, but not the second.
>
> If the matrix equation is implemented in the form described in the table following Equation (68), it will be a 19×15 matrix. Exactly four of the rows will be indentically zero. In my original coding of the Eulerian elimination scheme, I set the coefficients in a matrix $A(19{,}15)$ as per the 19 equations in Equation (68). Then, I caused the code to check to make sure that four rows were zero. Doing things the long way makes debugging easier, but is not very efficient. Lugging four useless equations through the process does take time. The easiest way to save some time is simply not to enter the four zero-equations in the matrix. Recall that the four zero equations arise from a set of eight equations which involve the four binary flags. The eight equations come in pairs, and it happens that one of each pair vanishes whatever the setting of the flags. In the final version of the code, the main matrix is declared as square $A(15{,}15)$. The 19 equations are examined sequentially and only the 15 non-zero equations have their coefficients entered into the matrix.
>
> I did not implement any further reduction from 15×15. Since the computation time required to invert an $N \times N$ matrix rises by the order of N^2, even minor reductions of scale give a disproportionate saving in time. Some of the unknowns are referred to only two or three times in the entire set of equations. The rotational accelerations of the three sets of wheels are examples. It would be possible to substitute some of the equations of motion into the remaining equations without too much increase in the complexity of the elements in matrix A.
>
> As it turns out, a 100 microsecond time step can easily be handled in real time. In order to give a realistic display, the screen is refreshed every 200 time steps, or nominally 50 times per second. Since it takes the operating system a little bit of time to do the calculations required to refresh the screen, the observed refresh rate is about 25 times per second.

<u>Including realistic moments of inertia</u>

In this and the following sections, I will add realistic features to the rig. I will add the features one-by-one so their effects on the run can be evaluated individually. First, I gave the three sets of wheels some moments of inertia.

I assumed that the truck's front and rear wheels were the same, and weighed 300 pounds per axle $\left(M_{twf} = M_{twr} = 300 \text{ lbs}\right)$. In order to keep the truck's total weight and front-to-rear weight distribution unchanged, I removed 300 pounds from the masses assumed to be concentrated at the front and rear axles.

I approximated the truck's wheels as uniform disks with diameters of 31½ inches and masses of 300 pounds. The formula given in the earlier paper gives the moment of inertia as $\frac{1}{2}MR^2$, or 258 pound-feet-squared $\left(I_{twf} = I_{twr} = 258 \text{ lb}-\text{ft}^2\right)$.

I assumed that the rear wheels on the sled were smaller, only 20 inches in diameter ($R_{swr} = 20''$), but also weighed 300 pounds ($M_{swr} = 300 \text{ lbs}$). Their weight was removed from the concentrated mass at the rear end of the sled, and the wheels were assigned a moment of inertia of 104 pound-feet-squared $\left(I_{swr} = 104\ \text{lb–ft}^2\right)$. The results of the simulation were:

$$\text{Distance} = 173.5\ \text{feet} \quad \text{Time} = 19.00\ \text{seconds} \quad (77)$$

Although the moments did not reduce the run time, they did reduce the distance covered by about 5.7 feet, or 3.2%.

Increasing the deck height

In the simplified model, and so far in the simulation, the deck height of the sled has been the same as the radius of the truck's wheels. For this next change, I increased the deck height to 25 inches, so it clears the rear wheels. I left the heights of both hitch balls at 15¾ inches for this next run. The results were unchanged:

$$\text{Distance} = 173.5\ \text{feet} \quad \text{Time} = 19.00\ \text{seconds} \quad (78)$$

Setting more realistic hitch dimensions

In the simplified model, and so far in the simulation, the tow chain has been horizontal. In reality, the tow chain normally slopes upwards towards the truck. I believe that more realistic hitch heights are 14 inches at the truck ($Y_t = 14''$)and eight inches at the sled ($Y_s = 8''$).

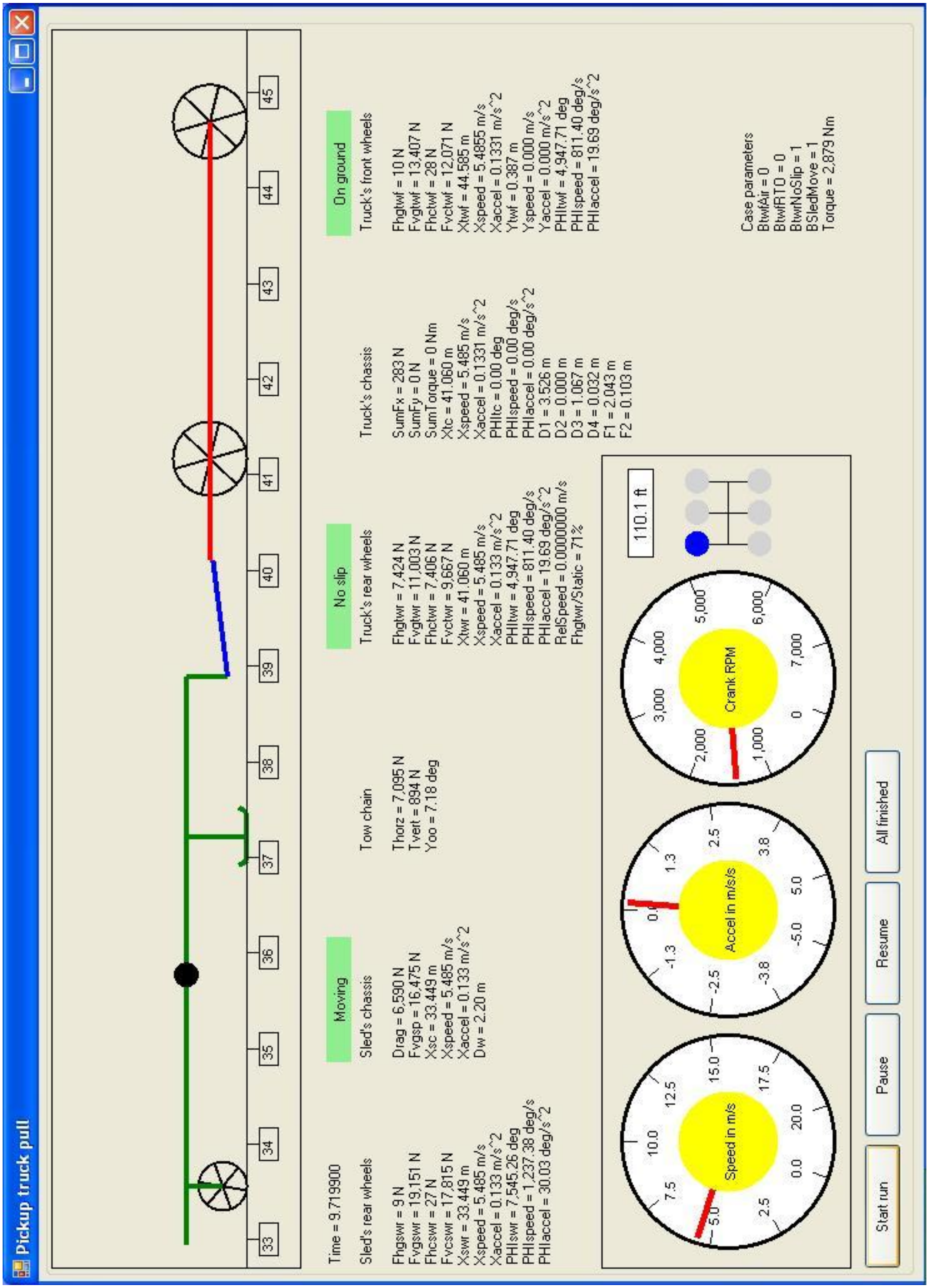
When the tow chain is horizontal, the length of the tow chain and the distances of the two hitches from their two vehicles do not matter. Once the tow chain is not horizontal, they do matter. For this simulation, I made the following assumptions.

- The horizontal distance from the truck's rear axle to the hitch (X_t) is 3'6";
- The horizontal distance from the sled's hitch to the pan's center-line (X_s) is 5'6" and
- The length of the tow chain L_{tow} is four feet.

With these dimensions, the slope of the hitch is 7.2°. This, of course, will be the inclination angle of the tension force in the tow chain. But, the other dimensions, such as the distance of the hitch from the wheels, are also important. They determine how much torque the tension exerts on the vehicles. An upward-sloping tow chain will reduce the effective weight on the pan but reduce the horizontal force accelerating the sled. The former increases the run distance; the latter reduces it. It seems that the former outweighs the latter, as the following simulation result shows.

$$\text{Distance} = 243.7\ \text{feet} \quad \text{Time} = 21.69\ \text{seconds} \quad (79)$$

I will look at a couple of more sensitivities after the following figure. It shows the screen part-way through this run. The top third of the screen is a visual display of the rig, with meter-markers on the ground to track its progress. The middle third allows the viewer to keep track of significant variables during the run. The bottom third is a representation of the dashboard with, from left-to-right, a speedometer in meters per second, an accelerometer in meters per second squared, a tachometer in RPM and a diagram of the gearshift H set up for six forward gears. Although this paper only deals with the more complex dynamics of the rig, I set up the dashboard in anticipation of a more complex model for the powertrain to be described in a subsequent paper.


Pickup truck pull
33
34
35
36
37
38
39
40
41
42
43
44
45
Time = 9.719900
Sled's rear wheels
Fhgswr = 9 N
Fvgswr = 19,151 N
Fhcswr = 27 N
Fvcswr = 17,815 N
Xswr = 33.449 m
Xspeed = 5.485 m/s
Xaccel = 0.133 m/s^2
PHIswr = 7,545.26 deg
PHIspeed = 1,237.38 deg/s
PHIaccel = 30.03 deg/s^2
Moving
Sled's chassis
Drag = 6,590 N
Fvgsp = 16,475 N
Xsc = 33.449 m
Xspeed = 5.485 m/s
Xaccel = 0.133 m/s^2
Dw = 2.20 m
Tow chain
Thorz = 7,095 N
Tvert = 894 N
Yoo = 7.18 deg
No slip
Truck's rear wheels
Fhgtwr = 7,424 N
Fvgtwr = 11,003 N
Fhctwr = 7,406 N
Fvctwr = 9,667 N
Xtwr = 41.060 m
Xspeed = 5.485 m/s
Xaccel = 0.133 m/s^2
PHItwr = 4,947.71 deg
PHIspeed = 811.40 deg/s
PHIaccel = 19.69 deg/s^2
RelSpeed = 0.0000000 m/s
Fhgtwr/Static = 71%
Truck's chassis
SumFx = 283 N
SumFy = 0 N
SumTorque = 0 Nm
Xtc = 41.060 m
Xspeed = 5.485 m/s
Xaccel = 0.1331 m/s^2
PHItc = 0.00 deg
PHIspeed = 0.00 deg/s
PHIaccel = 0.00 deg/s^2
D1 = 3.526 m
D2 = 0.000 m
D3 = 1.067 m
D4 = 0.032 m
F1 = 2.043 m
F2 = 0.103 m
On ground
Truck's front wheels
Fhgtwf = 10 N
Fvgtwf = 13,407 N
Fhctwf = 28 N
Fvctwf = 12,071 N
Xtwf = 44.585 m
Xspeed = 5.4855 m/s
Xaccel = 0.1331 m/s^2
Ytwf = 0.387 m
Yspeed = 0.000 m/s
Yaccel = 0.000 m/s^2
PHItwf = 4,947.71 deg
PHIspeed = 811.40 deg/s
PHIaccel = 19.69 deg/s^2
Speed in m/s
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Accel in m/s/s
-5.0
-3.8
-2.5
-1.3
0.0
1.3
2.5
3.8
5.0
Crank RPM
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
110.1 ft
Case parameters
BtwfAir = 0
BtwfRTO = 0
BtwrNoSlip = 1
BSledMove = 1
Torque = 2,879 Nm
Start run
Pause
Resume
All finished

Further increases in the height of the truck's hitch

The previous simulation constitutes what I consider to be the basic case for the 1997 F-150 SuperCab. As expected, adding an upwards slope to the tow chain improved the run distance. I mentioned that increasing the slope is a trade-off between reducing the effective load on the pan (since the truck pulls the chain upwards) but also reducing the horizontal acceleration. There should be an optimal slope for the tow chain, where the effects are in balance. To explore this, I ran simulations with the height of the truck's hitch (Y_t) raised further and further off the ground. The results are:

Y_t	Chain slope	Distance	Time	
14"	7.2°	243.7 feet	21.69 seconds	$(80A)$
16"	9.6°	263.9 feet	22.47 seconds	$(80B)$
18"	12.0°	285.7 feet	23.33 seconds	$(80C)$
24"	19.5°	366.6 feet	26.58 seconds	$(80D)$

There is no sign of abatement. Raising the hitch further is not necessary, since the rig already travels further than most courses. On balance, though, it seems as if distances would continue to increase with increasing hitch height. Class rules will need to be consulted to determine if there are statutory constraints on this parameter.

Changing the horizontal location of the hitch on the truck

Anything that changes the slope of the tow chain will affect the distance achieved during a run. The horizontal position of the hitch on the truck may also have an effect. To explore this possibility, I changed the distance of the hitch aft of the truck's rear wheels (X_t). In the stock truck, the distance is 3'6". I tried it at six inches less and 12 inchs less, to model the hitch point being moved closer and closer to the rear axle. In all the cases, I used the same "base case" height above ground: $Y_t = 14"$. The results are:

X_t	Chain slope	Distance	Time	
$3'6"$	7.2°	243.7 feet	21.69 seconds	$(81A)$
$3'$	7.2°	243.7 feet	21.69 seconds	$(81B)$
$2'6"$	7.2°	243.7 feet	21.69 seconds	$(81C)$

There was no change. The sled was merely closer to the tailgate. A common characteristic of all the runs so far is that the front wheels of the truck remained on the ground and the rear wheels did not slip. Changing the horizontal placement of the truck's hitch ball does change the overall torque around the rear axle. But, that only affects the dynamics if it leads to either wheelies or wheel-slipping. Neither of these ever occurred during these runs so the effects of horizontal placement did not express themselves.

Changing the coefficient of dynamic friction of the pan

In the "base case", both here and in the earlier paper, I assumed that the coefficient of dynamic friction of the pan was equal to 0.4 $(C_{pd} = 0.4)$. The retarding drag which the ground exerts on the bottom of the pan is proportional to this coefficient. Increasing the coefficient increases the drag, leading to reduced run distances. The following simulation results show by how much.

C_{pf}	Distance	Time	
0.40	243.7 feet	21.69 seconds	$(82A)$
0.41	228.6 feet	20.99 seconds	$(82B)$
0.42	214.8 feet	20.35 seconds	$(82C)$

Reducing the coefficient will lead to increases in the run distance by the same factor, about 15 feet per basis point of coefficient. The participant in a truck pull event may or may not have some control over when he runs. If so, he will want to run as soon as possible after a brief rain shower, or perhaps to wait for one if one is expected. If track maintenance is not performed after every run, it would be best to run as soon as possible after maintenance is done. I assume that the track is compacted every time the pan passes over. Compaction increases the coefficient. Running after the surface has been loosened affords a lower coefficient.

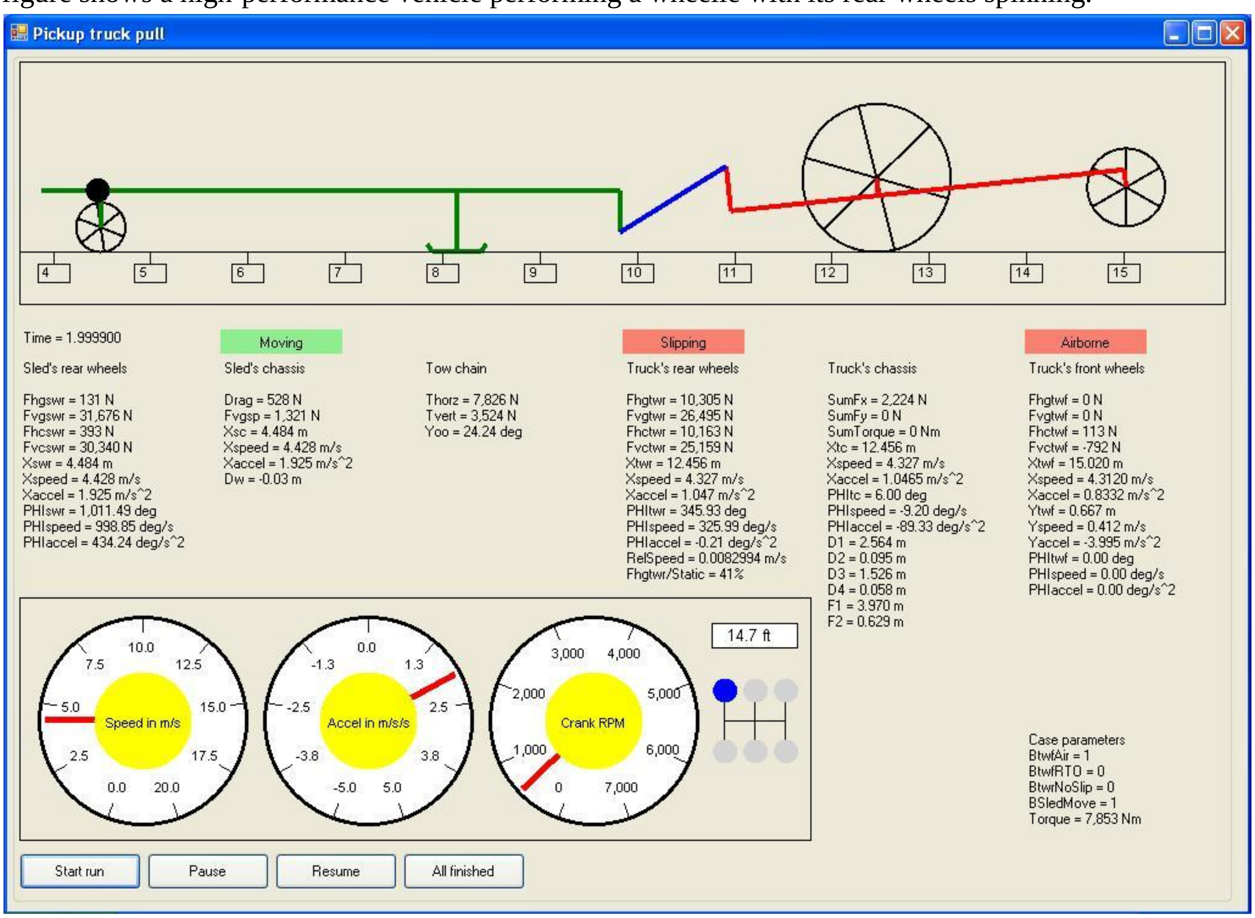
The benefits of smaller tires

The simplified model alerted us to the benefits of using smaller diameter tires. The following simulations confirm that finding.

$R_{twr} = R_{twf}$	Distance	Time	
31½" dia.	243.7 feet	21.69 seconds	$(83A)$
31" dia.	258.7 feet	22.21 seconds	$(83B)$
30½" dia.	275.0 feet	22.78 seconds	$(83C)$

A typical high-performance vehicle

High-performance vehicles can do more interesting things than a stock pickup truck. The following figure shows a high-performance vehicle performing a wheelie with its rear wheels spinning.



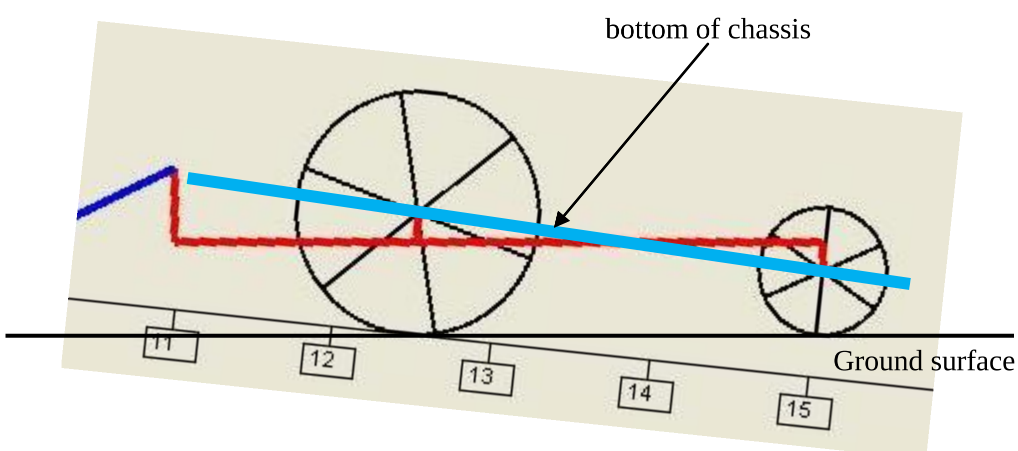
To model this vehicle, I made the following changes to the stock pickup truck. Each of these changes is made in the direction, that is, an increase or decrease, to increase the tendency of the truck to rotate around its rear axle.

- Reduced the wheelbase from 138.8" to 100";
- Reduced the mass over the front wheels by 2,000 pounds, from 2,940 pounds to 940 pounds;
- Added that weight (2,000 pounds) to the mass over the rear axle;
- Increased the tire size from 31½" to 60";
- Increased the truck's hitch height to 2'10";
- Increased the horizontal distance from the rear axle to the hitch to 5' and
- Increased the crankshaft torque from 330 foot-pounds to 900 foot-pounds.

Spot the problem? The above screenshot was taken early in the run, two seconds after the start. The wheels are turning slowly and, because of the big wheels, the engine is turning slowly as well. In the simplified model, the crankshaft speed is the wheels' rotational speed, increased by the transmission and differential gear ratios. This is the RPM shown on the tachometer. When the screenshot was taken, the engine was turning over at about 600 RPM. This is certainly less than the RPM which corresponds to maximum torque. It is also likely to be less than the idling speed. In other words, the real-world vehicle would probably be stalled.

Using a mathematical model based on an average amount of torque applied as a constant during a run has its shortcomings. This shortcoming will be fixed in a subsequent paper which describes an enhanced model for the powertrain.

Some readers might also question what appears to be an unaccepatbly high hitch point. The figure drawn for the truck is misleading. As I described in the earlier paper, I modeled the chassis as a horizontal massless line with concentrated masses at the axles. Using such a horizontal line has no effect on the dynamcis but obscures something else. The follwing figure shows only the truck from the previous figue. I have rotated the image to put the bottom of the wheels parallel to the ground. Then, I have drawn a heavy blue line through the axles, marking the approximate bottom of the truck's body. If anything, the hitch could be raised even higher.



Part X -- The computer program

The simulation was carried out by a Visual Basic program written in Microsoft's Visual Basic 2010 Express. There are several modules, most of which deal with the input of model parameters and the output of results, both to the screen and to an Excel file. I have set out in Appendix "A" attached hereto a listing of the code for a module named *EOMSolver*, which prepares matrix $A(15,15)$ and inverts it. EOMSolver also contains the subroutine which integrates the knienamtic variables through a time step. I have set out in Appendix "B" a listing of module *EOMDecider*, which implements the testing requires at the beginning of each time step to predict the state of the rig during the upcoming time step.

Jim Hawley
November 2013

An e-mail setting out errors and omissions would be appreciated.

Appendix "A"

Listing of module *EOMSolver*

```
Option Strict On
Option Explicit On

' This module contains two principal subroutines, which: (i) solve the equations of
' motion and (ii) integrate the speed and distance variables from the accelerations
' computed by solving the equations of motion.
'   Eulerian_Elimination()
'   Integrate_One_Step()

' The matrix is named A(15,15).  The vector ReductionRows(15) is initialized to ones.
' Every time a row is in A(,) is used as the basis for normalization during the first
' phase of the process, its ReductionRows() entry is set to zero.  New ReductionRows
' can only be selected from among the remaining unused rows.  Vector SubstitutionRows(19)
' is similar.  It is also initialized to ones.   Every time a row is used as the basis
' for back substitution during the second phase of the process, its SubstitutionRows()
' entry is set to zero.  New SubstitutionRows can only be selected from among the
' remaining unused rows.

Public Module EOMSolver

    Public Sub Euler_Elimination( _
        ByVal DebugEquations As Boolean, ByVal DebugStartTime As Double, _
        ByVal DisplayResults As Boolean, ByVal DisplayStartTime As Double)
        Dim A(15, 15) As Double
        Dim Unknown(15) As Double
        Dim B(15) As Double
        Dim ReductionRows(15) As Int32
        Dim SubstitutionRows(15) As Int32
        '
        ' Calculate the D* distances
        Dim HypotenuseFront As Double = Math.Sqrt(((RADtwr - RADtwf) ^ 2) + (Wt ^ 2))
        Dim ArcTanFactorFront As Double = Math.Atan2(Wt, RADtwr - RADtwf)
        Dim AnglePlus As Double = ArcTanFactorFront + PHItc
        D1 = HypotenuseFront * Math.Sin(AnglePlus)
        D2 = HypotenuseFront * Math.Cos(AnglePlus)
        Dim HypotenuseRear As Double = Math.Sqrt(((RADtwr - Yt) ^ 2) + (Xt ^ 2))
        Dim ArcTanFactorRear As Double = Math.Atan2(Xt, RADtwr - Yt)
        Dim AngleMinus As Double = ArcTanFactorRear - PHItc
        D3 = HypotenuseRear * Math.Sin(AngleMinus)
        D4 = HypotenuseRear * Math.Cos(AngleMinus)
        ' Calculate the F* functions
        Dim FTerm1 As Double = RADtwr - D4 - Ys
        Dim FTerm2 As Double = (Ltow ^ 2) - (FTerm1 ^ 2)
        Dim FTerm3 As Double = Math.Sqrt(FTerm2)
        Dim FTerm4 As Double = (D3 ^ 2) + (FTerm1 * D4)
        Dim FTerm5 As Double = FTerm1 * D3
        Dim FTerm6 As Double = FTerm5 * D3
        F1 = 1 + (FTerm4 / FTerm3) + (FTerm6 / (FTerm2 * FTerm3))
        F2 = (FTerm5 / FTerm3) - D4
        ' Calculate the Yoo function
        Yoo = FTerm1 / FTerm3
        '
        ' Zero-out all entries in matrix A(,) and vector Unknown()
```

```
For Irow As Int32 = 1 To 15 Step 1
    For Icol As Int32 = 1 To 15 Step 1
        A(Irow, Icol) = 0
    Next Icol
Next Irow
For Irow As Int32 = 1 To 15 Step 1
    Unknown(Irow) = 0
    ReductionRows(Irow) = 1
    SubstitutionRows(Irow) = 1
Next Irow
'
' Calculate the coefficients in the matrices
' Row #1, Equation (68R)
A(1, 1) = Mtruck
A(1, 2) = (Mtruck * F2) + (Mtwf * D2)
A(1, 8) = 1
A(1, 13) = -1
A(1, 14) = 1
B(1) = ((Mtwf * D1) - (Mtruck * F1)) * PHIDottc * PHIDottc
' Row #2, Equation (68T)
A(2, 1) = Mtwf * D2
A(2, 2) = MOItc + (Mtwf * ((D1 * D1) + (F2 * D2) + (D2 * D2)))
A(2, 7) = -D3
A(2, 8) = D4
A(2, 14) = D2
A(2, 15) = -D1
B(2) = (-Mtwf * F1 * D2 * PHIDottc * PHIDottc) + _
    Torque - ((Mtwf + Mtcf) * D1 * G)
' Row #3, Equation (68C)
A(3, 3) = MOItwf
A(3, 14) = -RADtwf
B(3) = 0
' Row #4, Equation (68K)
A(4, 1) = 1
A(4, 4) = -RADswr
B(4) = 0
' Row #5, Equation (68N)
A(5, 5) = MOItwr
A(5, 13) = RADtwr
B(5) = Torque
' Row #6, Equation (68W)
A(6, 6) = Ws
A(6, 7) = Ws + Xs
A(6, 8) = RADswr - Ys
A(6, 12) = -RADswr
B(6) = (Mscf * G * Ws) + (Msm * G * Dw)
' Row #7, Equation (68AA)
A(7, 7) = 1
A(7, 8) = -Yoo
B(7) = 0
' Row # 8, Equation (68U)
A(8, 1) = Msled
A(8, 8) = -1
A(8, 11) = 1
A(8, 12) = 1
B(8) = 0
' Row #9, Equation (68S)
A(9, 2) = Mtwf * D1
```

```
A(9, 7) = 1
A(9, 9) = -1
A(9, 15) = -1
B(9) = (-Mtwf * D2 * PHIDottc * PHIDottc) - (Mtruck * G)
' Row #10, Equation (68V)
A(10, 6) = 1
A(10, 7) = 1
A(10, 10) = 1
B(10) = Msled * G
' Row #11, Equation (68J)
A(11, 4) = MOIswr
A(11, 11) = -RADswr
B(11) = 0
' Beginning of rows which depend on the binary flags
Dim MatrixRow As Int32 = 11
' Row #12, Equation (68X)
If (BSledMove <> 0) Then
    MatrixRow = MatrixRow + 1
    A(MatrixRow, 6) = -BSledMove * Cpd
    A(MatrixRow, 12) = BSledMove
    B(MatrixRow) = 0
end if
' Row #13, Equation (68P)
If (BtwrNoSlip <> 1) Then
    MatrixRow = MatrixRow + 1
    A(MatrixRow, 9) = (BtwrNoSlip - 1) * _
        Math.Min(Ctwrd, Torque / (Fvgtwr * RADtwr))
    A(MatrixRow, 13) = 1 - BtwrNoSlip
    B(MatrixRow) = 0
End If
' Row #14, Equation (68D)
If (BtwfAir <> 0) Then
    MatrixRow = MatrixRow + 1
    A(MatrixRow, 14) = BtwfAir
    B(MatrixRow) = 0
End If
' Row #15, Equation (68E)
If ((BtwfAir + ((1 - BtwfAir) * BtwfRTO)) <> 0) Then
    MatrixRow = MatrixRow + 1
    A(MatrixRow, 15) = BtwfAir + ((1 - BtwfAir) * BtwfRTO)
    B(MatrixRow) = 0
End If
' Row #16, Equation (68F)
If (BtwfAir <> 1) Then
    MatrixRow = MatrixRow + 1
    A(MatrixRow, 1) = 1 - BtwfAir
    A(MatrixRow, 2) = (1 - BtwfAir) * (F2 + D2)
    A(MatrixRow, 3) = (BtwfAir - 1) * RADtwf
    B(MatrixRow) = (BtwfAir - 1) * (F1 - D1) * PHIDottc * PHIDottc
End If
' Row #17, Equation (68G)
If ((BtwfAir <> 1) And (BtwfRTO <> 1)) Then
    MatrixRow = MatrixRow + 1
    A(MatrixRow, 2) = (1 - BtwfAir) * (1 - BtwfRTO) * D1
    B(MatrixRow) = (BtwfAir - 1) * (1 - BtwfRTO) * D2 * PHIDottc * PHIDottc
End If
' Row #18, Equation (68O)
If (BtwrNoSlip <> 0) Then
```

```
        MatrixRow = MatrixRow + 1
        A(MatrixRow, 1) = BtwrNoSlip
        A(MatrixRow, 2) = BtwrNoSlip * F2
        A(MatrixRow, 5) = -BtwrNoSlip * RADtwr
        B(MatrixRow) = -BtwrNoSlip * F1 * PHIDottc * PHIDottc
    End If
    ' Row #19, Equation (68Y)
    If (BSledMove <> 1) Then
        MatrixRow = MatrixRow + 1
        A(MatrixRow, 1) = 1 - BSledMove
        B(MatrixRow) = 0
    End If
    If (MatrixRow <> 15) Then
        MsgBox("Logic error: Matrix has " & Trim(Str(MatrixRow)) & " rows.")
    End If
    '
    ' For debugging purposes, print out the matrices
    If ((DebugEquations = True) And (Time >= DebugStartTime)) Then
        Dim OutStr As String
        OutStr = _
            "Time = " & Trim(Str(Time)) & vbCrLf & _
            "BtwfAir=" & Trim(Str(BtwfAir)) & "      " & _
            "BtwfRTO = " & Trim(Str(BtwfRTO)) & "      " & _
            "BtwrNoSlip=" & Trim(Str(BtwrNoSlip)) & "      " & _
            "BSledMove=" & Trim(Str(BSledMove)) & vbCrLf
        For Irow As Int32 = 1 To 15 Step 1
            OutStr = OutStr & "Row # " & Trim(Str(Irow)) & ":" & vbCrLf
            For Icol As Int32 = 1 To 15 Step 1
                OutStr = OutStr & Trim(Str(A(Irow, Icol))) & ", "
            Next Icol
            OutStr = OutStr & Trim(Str(B(Irow))) & vbCrLf
        Next Irow
        MsgBox(OutStr, , "Elements in original matrices A and B")
    End If
    '
    ' Ensure that the number of non-zero rows and columns are equal
    Dim RowIsZero As Boolean
    Dim ColIsZero As Boolean
    For Irow As Int32 = 1 To 15 Step 1
        RowIsZero = True
        For Icol As Int32 = 1 To 15 Step 1
            If (A(Irow, Icol) <> 0) Then
                RowIsZero = False
                Exit For
            End If
        Next Icol
        If (RowIsZero = True) Then
            MsgBox("Logic error: Row #" & Trim(Str(Irow)) & " is zero.")
            Exit Sub
        End If
    Next Irow
    For Icol As Int32 = 1 To 15 Step 1
        ColIsZero = True
        For Irow As Int32 = 1 To 15 Step 1
            If (A(Irow, Icol) <> 0) Then
                ColIsZero = False
                Exit For
            End If
```

```
        Next Irow
        If (ColIsZero = True) Then
            MsgBox("Logic error: Column #" & Trim(Str(Icol)) & " is zero.")
            Exit Sub
        End If
    Next Icol
    '
    ' Eulerian elimination
    ' MasterCol is the main column counter
    For MasterCol As Int32 = 1 To 15 Step 1
        ' Look for the first unused row in the master column which has a non-zero
        ' element.  Use that row as a pivot for normalization.
        Dim PivotRow As Int32 = 0
        For Irow As Int32 = 1 To 15 Step 1
            If (ReductionRows(Irow) = 1) Then
                If (A(Irow, MasterCol) <> 0) Then
                    PivotRow = Irow
                    Exit For
                End If
            End If
        Next Irow
        ' For speed's sake, do not confirm that PivotRow > 0.  Just continue.
        ReductionRows(PivotRow) = 0
        ' Reduce the leading coefficient in the PivotRow to one
        Dim PivotValue As Double = A(PivotRow, MasterCol)
        For Icol As Int32 = MasterCol To 15 Step 1
            A(PivotRow, Icol) = A(PivotRow, Icol) / PivotValue
        Next Icol
        B(PivotRow) = B(PivotRow) / PivotValue
        ' Normalize all following rows
        If (PivotRow <> 15) Then
            For Irow As Int32 = (PivotRow + 1) To 15 Step 1
                If (A(Irow, MasterCol) <> 0) Then
                    Dim Denominator As Double
                    Denominator = A(Irow, MasterCol)
                    For Icol As Int32 = MasterCol To 15 Step 1
                        A(Irow, Icol) = A(Irow, Icol) / Denominator
                    Next Icol
                    B(Irow) = B(Irow) / Denominator
                End If
            Next Irow
        End If
        ' Subtract the PivotRow from all following rows
        If (PivotRow <> 15) Then
            For Irow As Int32 = (PivotRow + 1) To 15 Step 1
                If (A(Irow, MasterCol) <> 0) Then
                    For Icol As Int32 = MasterCol To 15 Step 1
                        A(Irow, Icol) = A(Irow, Icol) - A(PivotRow, Icol)
                    Next Icol
                    B(Irow) = B(Irow) - B(PivotRow)
                End If
            Next Irow
        End If
    Next MasterCol
    '
    ' For debugging purposes, print out the reduced matrices
    If ((DebugEquations = True) And (Time >= DebugStartTime)) Then
        Dim OutStr As String
```

```
        OutStr = _
            "Time = " & Trim(Str(Time)) & vbCrLf & _
            "BtwfAir=" & Trim(Str(BtwfAir)) & "      " & _
            "BtwfRTO = " & Trim(Str(BtwfRTO)) & "      " & _
            "BtwrNoSlip=" & Trim(Str(BtwrNoSlip)) & "      " & _
            "BSledMove=" & Trim(Str(BSledMove)) & vbCrLf
        For Irow As Int32 = 1 To 19 Step 1
            OutStr = OutStr & "Row # " & Trim(Str(Irow)) & ":" & vbCrLf
            For Icol As Int32 = 1 To 15 Step 1
                OutStr = OutStr & Trim(Str(A(Irow, Icol))) & ", "
            Next Icol
            OutStr = OutStr & Trim(Str(B(Irow))) & vbCrLf
        Next Irow
        MsgBox(OutStr, , "Elements in reduced matrices A and B")
    End If
    '
    ' Back-substitute from the lower right upwards
    For MasterCol As Int32 = 15 To 1 Step -1
        ' Look through the unused rows for the last row whose first non-zero element
        ' is in the master column.  Use that row as the next substitution row.
        Dim PivotRow As Int32 = 0
        For Irow As Int32 = 15 To 1 Step -1
            If (SubstitutionRows(Irow) = 1) Then
                If (A(Irow, MasterCol) <> 0) Then
                    PivotRow = Irow
                    Exit For
                End If
            End If
        Next Irow
        ' For speed's sake, do not confirm that PivotRow > 0.  Just continue.
        SubstitutionRows(PivotRow) = 0
        ' As a reality check, ensure the pivot element is equal to one
        If (A(PivotRow, MasterCol) <> 1) Then
            MsgBox("Logic error: Substitution element is not equal to one.")
            Exit Sub
        End If
        ' Add up the rest of the row products
        Dim RowProductSum As Double
        RowProductSum = B(PivotRow)
        If (MasterCol < 15) Then
            For Icol As Int32 = (MasterCol + 1) To 15 Step 1
                RowProductSum = RowProductSum - _
                    (A(PivotRow, Icol) * Unknown(Icol))
            Next Icol
        End If
        ' Evaluate the corresponding unknown
        Unknown(MasterCol) = RowProductSum
    Next MasterCol
    '
    ' For de-bugging purposes, test the integrity of the equation
    If ((DebugEquations = True) And (Time >= DebugStartTime)) Then
        Dim TotalError As Double
        TotalError = 0
        For Irow As Int32 = 1 To 15 Step 1
            Dim RowError As Double
            Dim RowSum As Double
            RowError = 0
            RowSum = 0
```

```
            For Icol As Int32 = 1 To 15 Step 1
                RowSum = RowSum + (A(Irow, Icol) * Unknown(Icol))
            Next Icol
            RowError = RowSum - B(Irow)
            TotalError = TotalError + Math.Abs(RowError)
        Next Irow
        If (TotalError > Val("0.000000001")) Then
            MsgBox("Logic error: Matrix equation error exceeds 10^-9.")
        End If
    End If
    '
    ' Organize the results
    XDot2sc = Unknown(1)
    PHIDot2tc = Unknown(2)
    PHIDot2twf = Unknown(3)
    PHIDot2swr = Unknown(4)
    PHIDot2twr = Unknown(5)
    Fvgsp = Unknown(6)
    Tvtowt = Unknown(7)
    Thtowt = Unknown(8)
    Fvgtwr = Unknown(9)
    Fvgswr = Unknown(10)
    Fhgswr = Unknown(11)
    Dgsp = Unknown(12)
    Fhgtwr = Unknown(13)
    Fhgtwf = Unknown(14)
    Fvgtwf = Unknown(15)
    '
    ' Calculate the dependent accelerations
    XDot2tc = XDot2sc + (F1 * PHIDottc * PHIDottc) + (F2 * PHIDot2tc)
    XDot2twf = XDot2tc - (D1 * PHIDottc * PHIDottc) + (D2 * PHIDot2tc)
    YDot2twf = (D2 * PHIDottc * PHIDottc) + (D1 * PHIDot2tc)
    XDot2swr = XDot2sc
    XDot2twr = XDot2tc
    '
    ' Back-substitute for the internal forces
    Fhctwf = (Mtwf * XDot2twf) + Fhgtwf
    Fvctwf = Fvgtwf - (Mtwf * G) - (Mtwf * YDot2twf)
    Fhcswr = Fhgswr + (Mswr * XDot2swr)
    Fvcswr = Fvgswr - (Mswr * G)
    Fhctwr = Fhgtwr - (Mtwr * XDot2twr)
    Fvctwr = Fvgtwr - (Mtwr * G)
    '
    ' For de-bugging purposes, print out the results
    If ((DisplayResults = True) And (Time >= DisplayStartTime)) Then
        Dim OutStr As String
        OutStr = _
            "Time = " & Trim(Str(Time)) & vbCrLf & _
            "BtwfAir=" & Trim(Str(BtwfAir)) & "      " & _
            "BtwfRTO = " & Trim(Str(BtwfRTO)) & "      " & _
            "BtwrNoSlip=" & Trim(Str(BtwrNoSlip)) & "      " & _
            "BSledMove=" & Trim(Str(BSledMove)) & vbCrLf & _
            "XDot2sc = " & Trim(Str(XDot2sc)) & vbCrLf & _
            "PHIDot2tc = " & Trim(Str(PHIDot2tc)) & vbCrLf & _
            "PHIDot2twf = " & Trim(Str(PHIDot2twf)) & vbCrLf & _
            "PHIDot2swr = " & Trim(Str(PHIDot2swr)) & vbCrLf & _
            "PHIDot2twr = " & Trim(Str(PHIDot2twr)) & vbCrLf & _
            "Fvgsp = " & Trim(Str(Fvgsp)) & _
```

```
            "   Dgsp = " & Trim(Str(Dgsp)) & _
            "   %H/V = " & Trim(Str(Dgsp / Fvgsp)) & vbCrLf & _
            "Tvtowt = " & Trim(Str(Tvtowt)) & _
            "   Thtowt = " & Trim(Str(Thtowt)) & _
            "   %V/H = " & Trim(Str(Tvtowt / Thtowt)) & vbCrLf & _
            "Fvgtwr = " & Trim(Str(Fvgtwr)) & _
            "   Fhgtwr = " & Trim(Str(Fhgtwr)) & _
            "   %H/V = " & Trim(Str(Fhgtwr / Fvgtwr)) & vbCrLf & _
            "Fvgswr = " & Trim(Str(Fvgswr)) & _
            "   Fhgswr = " & Trim(Str(Fhgswr)) & _
            "   %H/V = " & Trim(Str(Fhgswr / Fvgswr)) & vbCrLf & _
            "Fvgtwf = " & Trim(Str(Fvgtwf)) & _
            "   Fhgtwf = " & Trim(Str(Fhgtwf))
        MsgBox(OutStr, , "Results of Eulerian elimination")
    End If
    '
    ' For de-bugging purposes, print out Equation (68)
    If ((DebugEquations = True) And (Time >= DebugStartTime)) Then
        Dim OutStr As String
        OutStr = _
            "Time = " & Trim(Str(Time)) & vbCrLf & _
            "BtwfAir=" & Trim(Str(BtwfAir)) & "      " & _
            "BtwfRTO = " & Trim(Str(BtwfRTO)) & "      " & _
            "BtwrNoSlip=" & Trim(Str(BtwrNoSlip)) & "      " & _
            "BSledMove=" & Trim(Str(BSledMove)) & vbCrLf & _
            "(68C) " & Trim(Str(MOItwf * PHIDot2twf)) & " = " & _
            Trim(Str(RADtwf * Fhgtwf)) & vbCrLf & _
            "(68D) " & Trim(Str(BtwfAir * Fhgtwf)) & " = 0" & vbCrLf & _
            "(68E) " & Trim(Str((BtwfAir + ((1 - BtwfAir) * BtwfRTO)) * _
            Fhgtwf)) & " = 0" & vbCrLf & _
            "(68F) " & Trim(Str((1 - BtwfAir) * _
            (XDot2sc + ((F1 - D1) * PHIDottc * PHIDottc) + _
            ((F2 + D2) * PHIDot2tc) - (RADtwf * PHIDot2twf)))) & " = 0" & vbCrLf & _
            "(68G) " & Trim(Str((1 - BtwfAir) * (1 - BtwfRTO) * _
            ((D2 * PHIDottc * PHIDottc) + (D2 * PHIDot2tc)))) & " = 0" & vbCrLf & _
            "(68J) " & Trim(Str(MOIswr * PHIDot2swr)) & " = " & _
            Trim(Str(RADswr * Fhgswr)) & vbCrLf & _
            "(68K) " & Trim(Str(XDot2sc)) & " = " & _
            Trim(Str(RADswr * PHIDot2swr)) & vbCrLf & _
            "(68N) " & Trim(Str(MOItwr * PHIDot2twr)) & " = " & _
            Trim(Str(Torque - (RADtwr * Fhgtwr))) & vbCrLf & _
            "(68O) " & Trim(Str(BtwrNoSlip * _
            (XDot2sc + (F1 * PHIDottc * PHIDottc) + (F2 * PHIDot2tc) + _
            (-RADtwr * PHIDot2twr)))) & " = 0" & vbCrLf & _
            "(68P) " & Trim(Str((1 - BtwrNoSlip) * _
            (Fhgtwr - (Ctwrd * Fvgtwr)))) & " = 0" & vbCrLf & _
            "(68R) " & Trim(Str((Mtruck * XDot2sc) + _
            (((Mtruck * F1) - (Mtwf * D1)) * PHIDottc * PHIDottc) + _
            (((Mtruck * F2) + (Mtwf * D2)) * PHIDot2tc))) & " = " & _
            Trim(Str(-Fhgtwf + Fhgtwr - Thtowt)) & vbCrLf & _
            "(68S) " & Trim(Str(Mtwf * _
            ((D2 * PHIDottc * PHIDottc) + (D1 * PHIDot2tc)))) & _
            " = " & Trim(Str(Fvgtwf + Fvgtwr - Tvtowt - (Mtruck * G))) & vbCrLf & _
            "(68T) " & Trim(Str((Mtwf * D2 * XDot2sc) + _
            (Mtwf * F1 * D2 * PHIDottc * PHIDottc) + _
            ((MOItc + (Mtwf * ((D1 * D1) + (F2 * D2) + (D2 * D2)))) * _
            PHIDot2tc))) & " = " & _
            Trim(Str(Torque + (Fvgtwf * D1) - (Fhgtwf * D2) + (Tvtowt * D3) + _
```

```
            (-Thtowt * D4) - ((Mtwf + Mtcf) * D1 * G))) & vbCrLf & _
            "(68U) " & Trim(Str(Msled * XDot2sc)) & " = " & _
            Trim(Str(Thtowt - Dgsp - Fhgswr)) & vbCrLf & _
            "(68V) 0 = " & Trim(Str(Tvtowt + Fvgsp + Fvgswr - _
            (Msled * G))) & vbCrLf & _
            "(68W) 0 = " & Trim(Str((Tvtowt * (Ws + Xs)) + _
            (Thtowt * (RADswr - Ys)) + ((Fvgsp - (Mscf * G)) * Ws) + _
            (-Dgsp * RADswr) - (Msm * G * Dw))) & vbCrLf & _
            "(68X) " & Trim(Str(BSledMove * (Dgsp - (Cpd * Fvgsp)))) & _
            " = 0" & vbCrLf & _
            "(68Y) " & Trim(Str((1 - BSledMove) * XDot2sc)) & " = 0" & vbCrLf & _
            "(68AA) " & Trim(Str(Tvtowt)) & " = " & _
            Trim(Str(Yoo * Thtowt)) & vbCrLf & _
            "(68Q) " & Trim(Str(BtwrNoSlip * Fhgtwr)) & " <= " & _
            Trim(Str(BtwrNoSlip * Ctwrs * Fvgtwr)) & vbCrLf & _
            "(68Z) " & Trim(Str((1 - BSledMove) * Dgsp)) & " <= " & _
            Trim(Str((1 - BSledMove) * Cps * Fvgsp))
        MsgBox(OutStr, , "Equality of Equation (68)")
    End If

End Sub

Public Sub Integrate_One_Step( _
    ByVal DisplayResults As Boolean, ByVal DisplayStartTime As Double)
    ' Primary EOM kinematic variables, direct integration
    Dim DeltaXsc As Double = (XDotsc * DelT) + (0.5 * XDot2sc * DelT * DelT)
    Dim DeltaXDotsc As Double = XDot2sc * DelT
    Dim DeltaPHItc As Double = (PHIDottc * DelT) + (0.5 * PHIDot2tc * DelT * DelT)
    Dim DeltaPHIDottc As Double = PHIDot2tc * DelT
    DeltaXsc = (XDotsc * DelT) + (0.5 * XDot2sc * DelT * DelT)
    DeltaXDotsc = XDot2sc * DelT
    DeltaPHItc = (PHIDottc * DelT) + (0.5 * PHIDot2tc * DelT * DelT)
    DeltaPHIDottc = PHIDot2tc * DelT
    ' Deduced changes and variables
    Dim DeltaXtc As Double
    Dim DeltaXDottc As Double
    ' Sled's chassis - horizontal position, direct integration
    Xsc = Xsc + DeltaXsc
    XDotsc = XDotsc + DeltaXDotsc
    ' Sled's rear wheels - horizontal position, Equation (42A)
    Xswr = Xsc
    XDotswr = XDotsc
    ' Sled's rear wheels - rotational position, Equation (38F)
    PHIswr = PHIswr + (DeltaXsc / RADswr)
    PHIDotswr = PHIDotswr + (DeltaXDotsc / RADswr)
    ' Truck's chassis - horizontal position, Equations (47) and (52)
    Dim XtcBefore As Double = Xtc
    Xtc = Xsc + Ws + Xs + (Ltow / Math.Sqrt(1 + (Yoo * Yoo))) + D3
    DeltaXtc = Xtc - XtcBefore
    Dim XDottcBefore As Double = XDottc
    XDottc = XDotsc + (((Yoo * D3) - D4) * PHIDottc)
    DeltaXDottc = XDottc - XDottcBefore
    ' Truck's chassis - rotational position, direct integration
    PHItc = PHItc + DeltaPHItc
    PHIDottc = PHIDottc + DeltaPHIDottc
    ' Truck's rear wheels - horizontal position, Equation (42B)
    Xtwr = Xtc
    XDottwr = XDottc
```

```
' Truck's rear wheels - rotational position, Equation
If (BtwrNoSlip = 1) Then     ' No slipping, Equation (21B)
    PHItwr = PHItwr + (DeltaXtc / RADtwr)
    PHIDottwr = XDottc / RADtwr
Else                         ' Slipping, Equation (41I) and direct integration
    PHItwr = PHItwr + (PHIDottwr * DelT) + (0.5 * PHIDot2twr * DelT * DelT)
    PHIDottwr = PHIDottwr + (PHIDot2twr * DelT)
End If
' Truck's front wheels - horizontal position, Equations (43A) and (45A)
Xtwf = Xtc + D1
XDottwf = XDottc + (D2 * PHIDottc)
' Truck's front wheels - vertical position, Equations (43B) and (45B)
Ytwf = RADtwr - D2
YDottwf = -D1 * PHIDottc
' Truck's front wheels - rotational position, Equations (4D) and (5F)
If (BtwfAir = 0) Then        ' Front wheels on ground
    PHItwf = PHItwf + (DeltaXsc / RADtwf)
    PHIDottwf = PHIDottwf + (DeltaXDotsc / RADtwf)
Else                         ' Front wheels airborne and direct integration
    PHItwf = (PHIDottwf * DelT) + (0.5 * PHIDot2twf * DelT * DelT)
    PHIDottwf = PHIDot2twf * DelT
End If
'
' Detect and force zero conditions
If (Math.Abs(Ytwf - RADtwf) < EqualityFuzziness) Then
    Ytwf = 0
End If
If (Math.Abs(XDotsc) < EqualityFuzziness) Then
    XDotsc = 0
End If
If (Math.Abs(PHItc) < EqualityFuzziness) Then
    PHItc = 0
End If
If (Math.Abs(Fvgtwf) < EqualityFuzziness) Then
    Fvgtwf = 0
End If
'
' Detect and correct unrealistic conditions.
' The truck will not rotate into the ground.
If (PHItc < 0) Then
    PHItc = 0
End If
' The truck's front wheels will not go below ground.
If (Ytwf < RADtwf) Then
    Ytwf = RADtwf
End If
' The sled will not travel backwards.
If (XDotsc < 0) Then
    XDotsc = 0
    XDotswr = 0
End If
' The sled's rear wheels will not turn backwards.
If (PHIDotswr < 0) Then
    PHIDotswr = 0
End If
'
' For de-bugging purposes, print out the results
If ((DisplayResults = True) And (Time >= DisplayStartTime)) Then
```

```
            Dim OutStr As String
            OutStr = _
                "Time = " & Trim(Str(Time)) & vbCrLf & _
                "BtwfAir=" & Trim(Str(BtwfAir)) & "      " & _
                "BtwfRTO = " & Trim(Str(BtwfRTO)) & "      " & _
                "BtwrNoSlip=" & Trim(Str(BtwrNoSlip)) & "      " & _
                "BSledMove=" & Trim(Str(BSledMove)) & vbCrLf & _
                "XDot2sc = " & Trim(Str(XDot2sc)) & _
                "    XDotsc = " & Trim(Str(XDotsc)) & _
                "    Xsc = " & Trim(Str(Xsc)) & vbCrLf & _
                "PHIDot2tc = " & Trim(Str(PHIDot2tc)) & _
                "    PHIDottc = " & Trim(Str(PHIDottc)) & _
                "    PHItc = " & Trim(Str(PHItc)) & vbCrLf & _
                "PHIDot2twf = " & Trim(Str(PHIDot2twf)) & _
                "    PHIDottwf = " & Trim(Str(PHIDottwf)) & _
                "    PHItwf = " & Trim(Str(PHItwf)) & vbCrLf & _
                "YDot2twf = " & Trim(Str(YDot2twf)) & _
                "    YDottwf = " & Trim(Str(YDottwf)) & _
                "    Ytwf = " & Trim(Str(Ytwf)) & vbCrLf & _
                "PHIDot2swr = " & Trim(Str(PHIDot2swr)) & _
                "    PHIDotswr = " & Trim(Str(PHIDotswr)) & _
                "    PHIswr = " & Trim(Str(PHIswr)) & vbCrLf & _
                "PHIDot2twr = " & Trim(Str(PHIDot2twr)) & _
                "    PHIDottwr = " & Trim(Str(PHIDottwr)) & _
                "    PHItwr = " & Trim(Str(PHItwr)) & vbCrLf & _
                "XDot2twr = " & Trim(Str(XDot2twr)) & _
                "    XDottwr = " & Trim(Str(XDottwr)) & _
                "    Xtwr = " & Trim(Str(Xtwr)) & vbCrLf & _
                "Fvgsp = " & Trim(Str(Fvgsp)) & _
                "    Dgsp = " & Trim(Str(Dgsp)) & _
                "    %H/V = " & Trim(Str(Dgsp / Fvgsp)) & vbCrLf & _
                "Tvtowt = " & Trim(Str(Tvtowt)) & _
                "    Thtowt = " & Trim(Str(Thtowt)) & _
                "    %V/H = " & Trim(Str(Tvtowt / Thtowt)) & vbCrLf & _
                "Fvgtwr = " & Trim(Str(Fvgtwr)) & _
                "    Fhgtwr = " & Trim(Str(Fhgtwr)) & _
                "    %H/V = " & Trim(Str(Fhgtwr / Fvgtwr)) & vbCrLf & _
                "Fvgswr = " & Trim(Str(Fvgswr)) & _
                "    Fhgswr = " & Trim(Str(Fhgswr)) & _
                "    %H/V = " & Trim(Str(Fhgswr / Fvgswr)) & vbCrLf & _
                "Fvgtwf = " & Trim(Str(Fvgtwf)) & _
                "    Fhgtwf = " & Trim(Str(Fhgtwf))
            MsgBox(OutStr, , "Results at end of time step")
        End If

    End Sub

End Module
```

Appendix "B"

Listing of module *EOMDecider*

```
Option Strict On
Option Explicit On

' This module contains one principal subroutine, which determines which equations of
' motion should be applied during the next time step.  It also contains a minor
' subroutine to assign cardinal numbers to the various cases.  These case ID numbers
' are used only for information and display purposes.
'   Set_Binary_Flags()
'   Set_CaseID_Numbers()

Public Module EOMDecider

    Public Sub Set_Binary_Flags(ByVal Debug As Boolean)
        '
        ' Determine the previous states
        If (Math.Abs(XDotsc) <= EqualityFuzziness) Then
            SledIsMoving = False
        Else
            If (XDotsc > 0) Then
                SledIsMoving = True
            Else
                MsgBox("Logic error: Sled is moving backwards.")
                Exit Sub
            End If
        End If
        If (Math.Abs(Ytwf - RADtwf) <= EqualityFuzziness) Then
            FrontIsAirborne = False
        Else
            If (Ytwf > RADtwf) Then
                FrontIsAirborne = True
            Else
                MsgBox("Logic error: Front wheels are below ground.")
                Exit Sub
            End If
        End If
        If (Math.Abs((RADtwr * PHIDottwr) - XDottwr) <= EqualityFuzziness) Then
            RearIsNotSlipping = True
        Else
            RearIsNotSlipping = False
        End If
        '
        ' First case
        ' Determine the future state of the front wheels, if they are now airborne
        If (FrontIsAirborne = True) Then
            If (YDottwf >= 0) Then
                BtwfAir = 1
                BtwfRTO = 0
            Else
                If (Math.Abs((Ytwf - RADtwf) / YDottwf) <= (DelT / 2)) Then
                    BtwfAir = 0
                    BtwfRTO = 0
                Else
                    BtwfAir = 1
```

```
            BtwfRTO = 0
        End If
    End If
End If
'
' Second case
' Determine the future state of the front wheels, if they are now on the ground
If (FrontIsAirborne = False) Then
    '
    ' *************************************************************
    ' ** Special test subsequently added for Fvgtwf < 0 at Time = 0
    ' *************************************************************
    If (Fvgtwf < 0) Then
        BtwfAir = 1
        BtwfRTO = 0
    Else
        Dim Slope As Double
        Slope = (Fvgtwf - FvgtwfLast) / DelT
        If (Slope >= 0) Then
            BtwfAir = 0
            BtwfRTO = 0
        Else
            If (Math.Abs(Fvgtwf / Slope) <= (DelT / 2)) Then
                BtwfAir = 0
                BtwfRTO = 1
            Else
                BtwfAir = 0
                BtwfRTO = 0
            End If
        End If
    End If
End If
'
' Third case
' Determine the future state of the sled's motion, if it is now moving
If (SledIsMoving = True) Then
    Dim Slope As Double
    Slope = XDot2sc
    If (Slope >= 0) Then
        BSledMove = 1
    Else
        If (Math.Abs(XDotsc / Slope) <= (DelT / 2)) Then
            BSledMove = 0
        Else
            BSledMove = 1
        End If
    End If
End If
'
' Fourth case
' Determine the future state of the sled's motion, if it is now stopped
If (SledIsMoving = False) Then
    If (Dgsp > (Cps * Fvgsp)) Then
        BSledMove = 1
    Else
        BSledMove = 0
    End If
End If
```

```
    '
    ' Fifth case
    ' Determine the future state of the rear wheels, if they are not slipping
    If (RearIsNotSlipping = True) Then
        If (Fhgtwr > (Ctwrs * Fvgtwr)) Then
            BtwrNoSlip = 0
        Else
            BtwrNoSlip = 1
        End If
    End If
    '
    ' Sixth case
    ' Determine the future state of the rear wheels, if they are now slipping
    If (RearIsNotSlipping = False) Then
        Dim RS As Double
        Dim RSSlope As Double
        RS = (RADtwr * PHIDottwr) - XDottwr
        RSSlope = (RADtwr * PHIDot2twr) - XDot2twr
        If (RSSlope >= 0) Then
            BtwrNoSlip = 0
        Else
            If (Math.Abs(RS / RSSlope) <= (DelT / 2)) Then
                BtwrNoSlip = 1
            Else
                BtwrNoSlip = 0
            End If
        End If
    End If
    '
    ' For debugging purposes, print out the conclusions
    If (Debug = True) Then
        Dim OutStr As String
        OutStr = _
            "Time = " & Trim(Str(Time)) & vbCrLf & _
            "Current state is:" & vbCrLf & _
            "  FrontIsAirborne = " & FrontIsAirborne.ToString & vbCrLf & _
            "  RearIsNotSlipping = " & RearIsNotSlipping.ToString & vbCrLf & _
            "  SledIsMoving = " & SledIsMoving.ToString & vbCrLf & _
            "Expected state is:" & vbCrLf & _
            "  BtwfAir = " & Trim(Str(BtwfAir)) & vbCrLf & _
            "  BtwfRTO = " & Trim(Str(BtwfRTO)) & vbCrLf & _
            "  BtwrNoSlip = " & Trim(Str(BtwrNoSlip)) & vbCrLf & _
            "  BSledMove = " & Trim(Str(BSledMove))
        MsgBox(OutStr, , "EOMDecider results")
    End If

End Sub

Public Sub Set_CaseID_Numbers()
    If (SledIsMoving = True) Then
        If (FrontIsAirborne = False) Then
            If (RearIsNotSlipping = True) Then
                CaseID = 1
            Else
                CaseID = 2
            End If
        Else
            If (RearIsNotSlipping = True) Then
```

```
                    CaseID = 3
                Else
                    CaseID = 4
                End If
            End If
        Else
            If (FrontIsAirborne = False) Then
                If (RearIsNotSlipping = True) Then
                    CaseID = 5
                Else
                    CaseID = 6
                End If
            Else
                If (RearIsNotSlipping = True) Then
                    CaseID = 7
                Else
                    CaseID = 8
                End If
            End If
        End If
    End Sub

End Module
```