Kinematics - Changing Frame-Of-Reference

The space coordinate system that we choose is somewhat arbitrary, perhaps we may choose orthogonal (mutually perpendicular) x,y and z coordinates that align with the surface of the earth where we happen to be, or we could choose any other convenient alignment. In the same way the origin point of the coordinate system is similarly arbitrary.

This can be useful as changing the frame of reference (different observers) in an appropriate way could simplify the mathematics.

When we go on to look at dynamics we shall see that the laws of motion are independent of certain changes in the coordinate system such as:

- time shift
- space translation
- space rotation
- uniform linear velocity

We can think of these as symmetries.

This means that if we do a mechanics experiment where the frame of reference is transformed in any of these ways then we will get the same result (provided that we are consistent: if we are using some equation then all the quantities in that equation must be measured in the same coordinate system). in fact this can be stated in a stronger way: there is no way, by doing physics experiments that we can determine an absolute time, space or velocity, these things always have to be measured relative to something. This relativity principle applies both to Newtonian and Einsteinian physics although, as we will see below, the nature of the velocity transform depends on which of these we are using.

There may be other symmetries, for instance, reflection in space or time, however there are some frames of reference which will alter the laws of physics, for example, where the frame of reference is accelerating or in a constant angular velocity.

We will now look at these changes or transforms individually in more detail.

Time Shift Transform

We can change the frame of reference by shifting the timeframe as follows:

$t'=t - t_0$

where:

- t' = time in new frame of reference (scalar value)
- t_0 = time in new frame of reference when t=0 (scalar value)
- t = time in original frame of reference (scalar value)

Alternatively we could use $t'=t + t_0$ if we defined t_0 as time in original frame of reference when t'=0

Another way to express this is to represent space-time as a 4 dimensional vector in which case, for newtonian physics, we get:

$$\begin{bmatrix} t' \\ x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} t_0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} t \\ x \\ y \\ z \end{bmatrix}$$

Time is not absolute in either Newtonian or Einsteinian physics in the sense that there is no preferred origin (except perhaps the time of the big bang?) but in Einsteinian physics time is even less absolute in that time intervals change due to relative velocity.

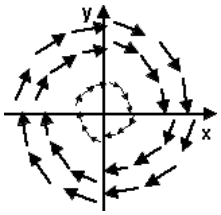
Space Translation Transform

We can change the frame of reference by shifting the origin as follows:

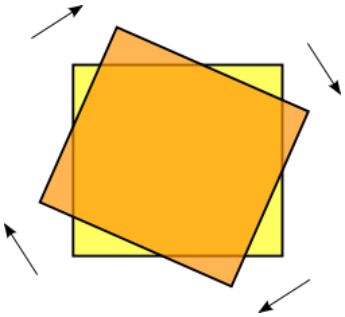
$x'=x - x_0$

where:

Transforms



A transform maps every point in a vector space to a possibly different point.



When transforming a computer model we transform all the vertices.

To model this using mathematics we can use matrices, quaternions or other algebras which can represent multidimensional linear equations.

This page explains this.

Galilean transform

- x' = position in new frame of reference (vector value)
- x_0 = position in new frame of reference when x=0 (vector value)
- x = position in original frame of reference (vector value)

Alternatively we could use $x'=x + x_0$ if we defined x_0 as time in original frame of reference when x'=0

Another way to express this is to represent space-time as a 4 dimensional vector in which case, for newtonian physics, we get:

$$\begin{bmatrix} t' \\ x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 0 \\ x_0 \\ y_0 \\ z_0 \end{bmatrix} + \begin{bmatrix} t \\ x \\ y \\ z \end{bmatrix}$$

Position is not absolute in either Newtonian or Einsteinian physics in the sense that there is no preferred origin but in Einsteinian physics distance is even less absolute in that distances change due to relative velocity.

Space Rotation Transform

In the same way we can rotate the space frame as follows:

$$\begin{bmatrix} t' \\ x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & r_{xx} & r_{xy} & r_{xz} \\ 0 & r_{yx} & r_{yy} & r_{yz} \\ 0 & r_{zx} & r_{zy} & r_{zz} \end{bmatrix} \begin{bmatrix} t \\ x \\ y \\ z \end{bmatrix}$$

Uniform Linear Velocity Transform

The transform between moving frames depends on whether we are using Newtonian or Einsteinian physics in Newtonian physics we use the Galilean transform and in Einsteinian physics we use the Lorentz transform.

Galilean transform

Assuming the relative motion 'v' is along the x dimension then $x \rightarrow x_0 + v_x t$

or if we have components of velocity in all dimensions the transform will be:

$$\begin{bmatrix} t' \\ x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ v_x & 1 & 0 & 0 \\ v_y & 0 & 1 & 0 \\ v_z & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t \\ x \\ y \\ z \end{bmatrix}$$

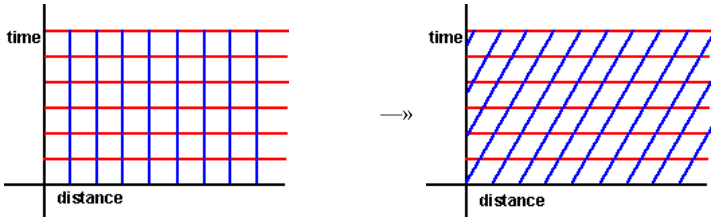
where:

- v_x,v_y,v_z= relative velocity of the two reference frames in x,y and z directions.
- x,y,z= position in original frame.
- x',y',z'= position in transformed frame.

in other words point:

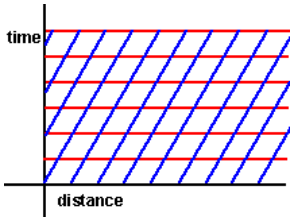
$$\begin{bmatrix} t \\ x \\ y \\ z \end{bmatrix}$$ is transformed to: $$\begin{bmatrix} t \\ x + v_x t \\ y + v_y t \\ z + v_z t \end{bmatrix}$$

The nature of this transform is a shear (also known as skew) transform:



We can take some of the ideas of relativity and use them to study classical Newtonian physics. For instance we can investigate the transform between different frames of reference which are moving relative to each other at a constant velocity.

- In Newtonian physics we use the Galilean transform.
- In Einsteinian physics we use the Lorentz transform.

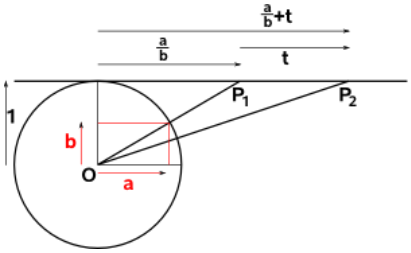


The Galilean transform can be represented by a shear matrix, the Lorentz transform is a hyperbolic rotation.

Projective Geometry

Has many applications including:

- Represent points, lines, planes, etc. which may be offset from origin.
- Combine these using meet and join.
- Projecting a 3D scene onto a 2D screen (used by openGL).
- Representing isometries (translations and rotations) as a single multiplication operation. This can greatly simplify the representation of objects moving in space.
- 4x4 matrix algebra provides a way to represent these transforms.



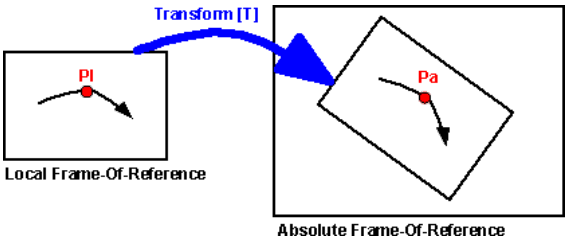
This page investigates this.

When doing this we choose to make time 'absolute' in that the time lines are left horizontal wheras the position lines are skewed althogh I guess that this is just a covention and we could have skewed the time and made the distance absolute.

more about Galilean transform.

Practical Issues

At first, it may seem a bit academic to transform these quantities into different frames-of-reference but, this is important for solving practical problems like collisions. For instance, the two objects colliding will have their inertial tensors defined in their own local coordinates, but when we work out the collision response, the impulse will have to be calculated in some common coordinate system.



What we are doing here is observing the same particle or solid object from different frames-of-reference. When the frames-of-reference are static (not moving relative to each other). Then the Newtonian laws will apply, regardless of where, or which direction, that we are looking at them from, provided that we are consistent about measuring all quantities on the same frame-of-reference.

It may be that the transform is changing with time, for example, if we are trying to solve a collision response, we might want to work in the frame-of-reference of one of the objects that is colliding. This object may be moving, so its frame-of-reference is moving with respect to the absolute coordinate system.

Do all the laws of physics apply when observing them from a moving frame-of-reference? If the frame-of-reference is moving with a constant linear velocity, then the Newtonian laws will apply just the same. A stationery object in one frame-of-reference may appear to be moving in another, but provided that we are constant about which frame-of-reference that we are working in, neither will contravene the laws. (Einsteins laws may not apply, the speed of light is the same in each frame-of-reference, relativity is not relative to the frame-of-reference - we are interested in slow speed interactions so this is not an issue for us).

However, if the frame-of-reference has angular motion (even if its constant), or if the frame-of-reference is accelerating, or if it has some irregular motion, then the Newtonian laws will not apply in this frame-of-reference.

To take an example, imagine a solid object travailing in free space, it may be spinning and have linear velocity as well. We may want to work in this objects own coordinate system, because this frame-of-reference is rotating, the Newtonian laws may not apply, for example an object which is stationary in the absolute frame-of-reference will appear to be traveling in a spiral in this objects frame of reference. An object with no forces acting on it should not be moving in a spiral, so Newtonian laws do not apply in this frame-of-reference. However, if we want to apply this inertia tensor, then we have to work in the local coordinate system of the rotating object.

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} i_{xx} & i_{xy} & i_{xz} \\ i_{yx} & i_{yy} & i_{yz} \\ i_{zx} & i_{zy} & i_{zz} \end{bmatrix} \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix}$$

So, we have to be careful. When we are calculating torques and forces, we often want to work in its local coordinate system, but when we are calculating motion we probably want to work in absolute coordinates.

Using matrix algebra to calculate transforms to other frames-of-reference

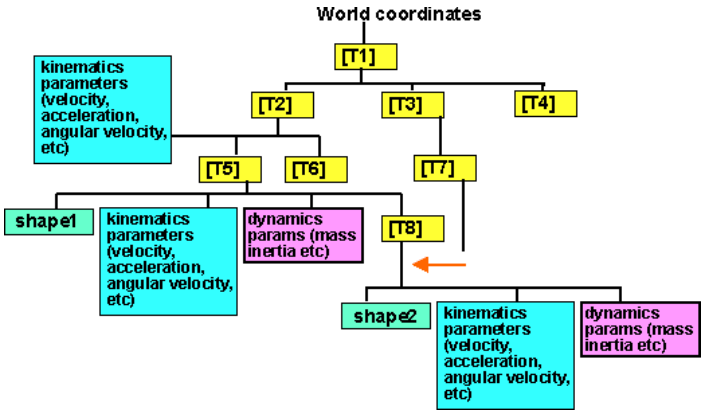
As described here, a 4x4 matrix can be used to represent a rotation and a translation in 3 dimensions. So we can use matrix algebra to translate from one frame of reference to another.

The transform will take a 3d vector representing a point in absolute coordinates and convert it into a 3d vector representing a point in absolute coordinates: $\vec{\mathbf{p}}_a = [T]\vec{\mathbf{p}}_l$

It is possible to have many layers of one frames-of-reference inside another. For example, if we know the position of the moon relative to the earth $[T_{me}]$ and we know the position of the earth in the frame-of-reference of the sun $[T_{es}]$, then we could work out the position of a point on the moon in the frame of reference of the sun $[T_{ms}]$.

It turns out that transforms can be concatenated by multiplying their corresponding matrices. So $[T_{ms}] = [T_{es}]*[T_{me}]$.

One way to represent this is to use a scene graph. We could use a scene graph is a similar way to VRML. We could put dynamics information into the scene graph in the same way that shape information is. In this page I would like to work out the effect of transforming dynamics information in this way.



The transform allows us to plug in a coordinate in the local frame-of-reference and get the coresponding coordinate in the absolute frame-of-reference.

$$\begin{bmatrix} P_{ax} \\ P_{ay} \\ P_{az} \\ 1 \end{bmatrix} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} & t_x \\ r_{yx} & r_{yy} & r_{yz} & t_y \\ r_{zx} & r_{zy} & r_{zz} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{lx} \\ P_{ly} \\ P_{lz} \\ 1 \end{bmatrix} = \begin{bmatrix} P_{lx} * r_{xx} + P_{ly} * r_{xy} + P_{lz} * r_{xz} + t_x \\ P_{lx} * r_{yx} + P_{ly} * r_{yy} + P_{lz} * r_{yz} + t_y \\ P_{lx} * r_{zx} + P_{ly} * r_{zy} + P_{lz} * r_{zz} + t_z \\ 1 \end{bmatrix}$$

For Solid objects, we need to specify both the location and the orientation of the object to define its position. So for solid objects we can easily translate its centre-of-mass (c-of-m) in this way, but how do we translate its orientation? One way that occurred to me would be to, take a point an infinitesimal distance away from from the c-of-m, translate that, and then see the orientation of this point relative to the translated c-of-m.

$$\begin{bmatrix} P_{ax}' \\ P_{ay}' \\ P_{az}' \\ 1 \end{bmatrix} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} & t_x \\ r_{yx} & r_{yy} & r_{yz} & t_y \\ izx & izy & izz & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{lx} + dP_{lx} \\ P_{ly} + dP_{ly} \\ P_{lz} + dP_{lz} \\ 1 \end{bmatrix} = \begin{bmatrix} (P_{lx} + dP_{lx}) * r_{xx} + (P_{ly} + dP_{ly}) * r_{xy} + (P_{lz} + dP_{lz}) * r_{xz} + t_x \\ (P_{lx} + dP_{lx}) * r_{yx} + (P_{ly} + dP_{ly}) * r_{yy} + (P_{lz} + dP_{lz}) * r_{yz} + t_y \\ (P_{lx} + dP_{lx}) * r_{zx} + (P_{ly} + dP_{ly}) * r_{zy} + (P_{lz} + dP_{lz}) * r_{zz} + t_z \\ 1 \end{bmatrix}$$

So the relative position is given by:

$$\begin{bmatrix} dP_{ax} \\ dP_{ay} \\ dP_{az} \\ 1 \end{bmatrix} = \begin{bmatrix} P_{ax}' \\ P_{ay}' \\ P_{az}' \\ 1 \end{bmatrix} - \begin{bmatrix} P_{ax} \\ P_{ay} \\ P_{az} \\ 1 \end{bmatrix} = \begin{bmatrix} dP_{lx} * r_{xx} + dP_{ly} * r_{xy} + dP_{lz} * r_{xz} \\ dP_{lx} * r_{yx} + dP_{ly} * r_{yy} + dP_{lz} * r_{yz} \\ dP_{lx} * r_{zx} + dP_{ly} * r_{zy} + dP_{lz} * r_{zz} \\ 1 \end{bmatrix} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} & 0 \\ r_{yx} & r_{yy} & r_{yz} & 0 \\ izx & izy & izz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} dP_{lx} \\ dP_{ly} \\ dP_{lz} \\ 1 \end{bmatrix}$$

So to transform a movement, then we transform it by the rotational part of the transform without the translational part of the matrix. Alternatively, if we don't want to modify the transform matrix, we could just use 0 for the 4th row of a relative movement vector, then the translational part will automatically be ignored.

$$\begin{bmatrix} dP_{ax} \\ dP_{ay} \\ dP_{az} \\ 0 \end{bmatrix} = [T] \begin{bmatrix} dP_{lx} \\ dP_{ly} \\ dP_{lz} \\ 0 \end{bmatrix}$$

But if we measuring orientation with 3 angles, θx, θy and θz. How can we translate this with [T]? This may seem a bit stange as we are using [T] to define the rotation of the frame-of-reference and θx, θy and θz to define the orientation in each frame-of-reference. The advantage of using θx, θy and θz would be that this notation would fit in better with the physics equations.

Using 6 dimensional vectors to calculate transforms to other frames of reference

For Solid objects, we need to specify both the location and the orientation of the object to define its position. It might be useful if we could represent the transform of both location and orientation in one equation. To do this we need a 6x6 matrix as the position has 6 degrees of freedom.

I am not sure if this will work for position because the normal rules of algebra don't apply when combining rotations (see rotations). Can anyone help me prove if this will work?

$$\begin{bmatrix} \theta a_x \\ \theta a_y \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} & m_{04} & m_{05} \\ m_{10} & m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \end{bmatrix} \begin{bmatrix} \theta l_x \\ \theta l_y \end{bmatrix}$$

θa_z		m_{20}	m_{21}	m_{22}	m_{23}	m_{24}	m_{25}	θl_z
P_{ax}		m_{30}	m_{31}	m_{32}	m_{33}	m_{34}	m_{35}	P_{lx}
P_{ay}		m_{40}	m_{41}	m_{42}	m_{43}	m_{44}	m_{45}	P_{ly}
P_{az}		m_{50}	m_{51}	m_{52}	m_{53}	m_{54}	m_{55}	P_{lz}

However I am sure it would work for velocities because infinitesimally small rotations can be combined in the usual way.

w_{ax}		m_{00}	m_{01}	m_{02}	m_{03}	m_{04}	m_{05}	w_x
w_{ay}		m_{10}	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}	w_y
w_{az}		m_{20}	m_{21}	m_{22}	m_{23}	m_{24}	m_{25}	w_z
v_{ax}	=	m_{30}	m_{31}	m_{32}	m_{33}	m_{34}	m_{35}	v_x
v_{ay}		m_{40}	m_{41}	m_{42}	m_{43}	m_{44}	m_{45}	v_y
v_{az}		m_{50}	m_{51}	m_{52}	m_{53}	m_{54}	m_{55}	v_z

This 6x6 matrix does not contain any extra information than the 4x4 transform [T] (even the 4x4 transform has redundant information when we are considering only a rotation and translation).

However, multiplication of a 6d vector by a 6x6 matrix, might be a lot easier than multiP_{ly}ing 4x4 matrix by a 4x4 matrix (which we would have to do several times if we are going to represent rotations by using matrices).

The advantage of this notation is that:

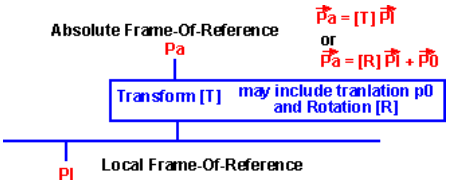
- The linear and angular quantities are dealt with is the same equation.
- The angular velocities and accelerations are defined in a way that is consistent with the usual dynamics equations.

Issues:

- Not sure if we can use this to transform points.
- Not sure how we would concatinate transforms (transform within a transform).

Point ($\vec{p}$)

Every coordinate point in the local frame of reference can be transformed to a coordinate in the absolute frame of reference $\vec{p}a = [T]\ \vec{p}1$. This can be thought of as the same point just measured in a different coordinate reference. A solid object can be transformed in this way by seperately transforming each point that makes it up.



This transform [T] is a 4x4 matrix and is capable of representing lots of tranforms such as sheer, scaleing, reflection, etc. which are not valid operations for a solid body, also it is very wasteful using a 4x4 matrix. Since we only want to represent translations and rotations, it may be more efficient to represent it in terms of a 3x3 rotation matrix [R] and a translation $\vec{P}0$.

$\vec{P}_a = [R]\ \vec{P}_1 + \vec{P}_0$

Where:

- $\vec{P}_a$ is the position in absolute coordinates.
- [R] is a 3x3 rotational matrix
- $\vec{P}_1$ is the position in local coordinates.
- $\vec{P}_0$ is the linear translation.

But we still have the problem that [R] could be used to repesent invalid operations for a solid object, we need to ensure that [R] is orthogonal. The problem is that if a calculation involves lots of transforms then rounding errors could distort [R], therefore we need to normalise (or should that be orthogonalise) after tranforms are combined.

One way round this would be to work from the quaternion [Q] to represent the rotation, as the quaternion is easier to normalise:

$\vec{P}_a =$

1 - 2 * (yy + zz)	2 * (xy - zw)	2 * (xz + yw)
2 * (xy + zw)	1 - 2 * (xx + zz)	2 * (yz - xw)
2 * (xz - yw)	2 * (yz + xw)	1 - 2 * (xx + yy)

$\vec{P}_1 + \vec{P}_0$

where:

- xx = qx* qx
- xy = qx * qy
- xz = qx * qz
- xw = qx * qw
- yy = qy * qy
- yz = qy * qz
- yw = qy * qw

- zz = qz * qz
- zw = qz * qw
- qx, qy, qz and qw are the values of the quaternion.

Linear Velocity($\vec{v}$)

$\vec{v}_a = d\vec{P}_a / dt = d[T]\ \vec{P}_l / dT$

$\vec{v}_a = d\vec{P}_a / dt = d$

$r_{xx} * P_{lx} + r_{xy} * P_{ly} + r_{xz} * P_{lz} + t_x$
$r_{yx} * P_{lx} + r_{yy} * P_{ly} + r_{yz} * P_{lz} + t_y$
$r_{zx} * P_{lx} + r_{zy} * P_{ly} + r_{zz} * P_{lz} + t_z$
0

$/dt$

using partial differential equations:

d (u * v) / dt = u dv/dt + v du/dt

so,

$\vec{v}_a =$

$r_{xx} * d\ P_{lx}/dt + d\ r_{xx}/dt * P_{lx} + r_{xy} * d\ P_{ly}/dt + d\ r_{xy}/dt * P_{ly} + r_{xz} * d\ P_{lz}/dt + d\ r_{xz}/dt * P_{lz} + d\ t_x/dt$
$r_{yx} * d\ P_{lx}/dt + d\ r_{yx}/dt * P_{lx} + r_{yy} * d\ P_{ly}/dt + d\ r_{yy}/dt * P_{ly} + r_{yz} * d\ P_{lz}/dt + d\ r_{yz}/dt * P_{lz} + d\ t_y/dt$
$r_{zx} * d\ P_{lx}/dt + d\ r_{zx}/dt * P_{lx} + r_{zy} * d\ P_{ly}/dt + d\ r_{zy}/dt * P_{ly} + r_{zz} * d\ P_{lz}/dt + d\ r_{zz}/dt * P_{lz} + d\ t_z/dt$
0

$\vec{v}_a =$

$r_{xx} * d\ P_{lx}/dt + r_{xy} * d\ P_{ly}/dt + r_{xz} * d\ P_{lz}/dt$
$r_{yx} * d\ P_{lx}/dt + r_{yy} * d\ P_{ly}/dt + r_{yz} * d\ P_{lz}/dt$
$r_{zx} * d\ P_{lx}/dt + r_{zy} * d\ P_{ly}/dt + r_{zz} * d\ P_{lz}/dt$
0

+

$d\ r_{xx}/dt * P_{lx} + d\ r_{xy}/dt * P_{ly} + d\ r_{xz}/dt * P_{lz}$
$d\ r_{yx}/dt * P_{lx} + d\ r_{yy}/dt * P_{ly} + d\ r_{yz}/dt * P_{lz}$
$d\ r_{zx}/dt * P_{lx} + d\ r_{zy}/dt * P_{ly} + d\ r_{zz}/dt * P_{lz}$
0

+

$d\ t_x/dt$
$d\ t_y/dt$
$d\ t_z/dt$
0

therefore:

$\vec{v}_a = [T1]\vec{v}_l + [T2]\vec{P}_l + \vec{v}_r$

where:

- $\vec{v}_a$ = linear velocity of point in absolute coordinates
- $\vec{v}_l$ = linear velocity of point in local coordinates
- $\vec{v}_r$ = linear velocity of local coordinates relative to absolute coordinates
- [T1] = transform [T] with only rotational components ($m_{03} = m_{02} = m_{01} = 0$)
- [T2] = rate of change of rotational components of [T]
-

$d\ r_{xx} / dt$	$d\ r_{xy} / dt$	$d\ r_{xz} / dt$	$d\ t_x / dt$
$d\ r_{yx} / dt$	$d\ r_{yy} / dt$	$d\ r_{yz} / dt$	$d\ t_y / dt$
d izx / dt	d izy / dt	d izz / dt	$d\ t_z / dt$
0	0	0	1

If the local coordinates are not rotated then:

$\vec{v}_a = \vec{v}_l + \vec{v}_r$

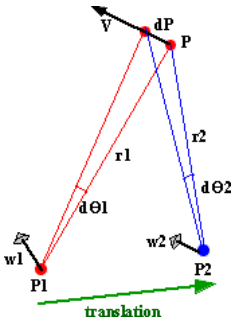
[T1]=

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

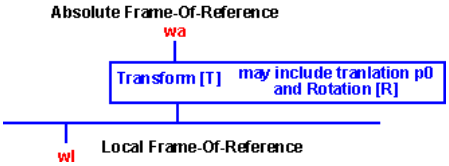
[T2]=

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Angular Velocity ($\vec{w}$)



We have particle or solid object which has an angular velocity wl about point pl when measured in local coordinates. What will be its angular velocity when these quantities are measured in absolute coodinates? In the most general case the transform itself may be changing itself, it may be representing a rotating frame-of-reference about a different point than the centre of the local rotation.



As shown here $\vec{v} = \vec{r} \times \vec{w}$

So when we are working in absolute coordinates:

$\vec{v}_a = \vec{r}a \times \vec{w}_a$

and when we are working in local coordinates:

$\vec{v}_l = \vec{r}l \times \vec{w}_l$

So, substituting this in the above equations:

$\vec{r}a \times \vec{w}_a = [T1](\vec{r}l \times \vec{w}_l) + [T2]\vec{p}_l + \vec{v}r$

Here I am stuck. can anyone help me continue? I think I am making incorrect assumptions here because the angular velocity of a solid object is just the rate of change of its orientation. The angular velocity of a particle on the object takes into account both the translation and the orientation of the object. So the angular velocity of a solid object and its component particles is different.

If the frame of reference is rotating about $\vec{p}_l = \vec{p}_a$ then the angular velocities can be added:

$\vec{w}_a = \vec{w}_l + \vec{w}_r$

So although the frame-of-reference is moving about a different centre than the point is rotating, the absolute rotation is still the sum of the roations of the f-of-r and the local rotation. The actual motion may follow a much more complex path. See the following examples:

- planetary motion
- rotating platform

Thank you to the following for helping me with this:

- Mati
- John
- Todd Wasson

Further information about angular velocity.

- Angular Velocity of particle
- Angular Velocity of solid body

Method using Curl

I would like to try and prove that $\vec{w}_a = \vec{w}_l + \vec{w}_r$ using vector calculus.

Given that $\vec{v}_a = \vec{v}_l + \vec{v}_r$

It is shown here that, for a solid object, that $\text{curl}(\vec{v}_a) = 2\,\vec{w}_a$

So if curl(a + b) = curl(a) + curl(b) we could prove that $\vec{w}_a = \vec{w}_l + \vec{w}_r$

The trouble is I can't find any indentity like curl(a + b) = curl(a) + curl(b)?

Can anyone help me?

Angular Acceleration ($\vec{\alpha}$)

As with angular velocity, the absolute angular acceleration is the sum of the angular acceleration of the frame-of-reference and the local angular acceleration:

$$\vec{\alpha}_a = \vec{\alpha}_l + \vec{\alpha}_r$$

- Angular Acceleration of particle
- Angular Acceleration of solid body

---

metadata block

see also:

- joints
- using a scene graph to model kinematics
- keyframeing
- dynamics
- physics of human animation
- sprung mass
- 3D programming

Correspondence about this page

Book Shop - Further reading.

Where I can, I have put links to Amazon for books that are relevant to the subject, click on the appropriate country flag to get more details of the book or to buy it from them.

Robot Dynamics Algorithms (Kluwer International Series in Engineering and Computer Science, 22)

Commercial Software Shop

Where I can, I have put links to Amazon for commercial software, not directly related to the software project, but related to the subject being discussed, click on the appropriate country flag to get more details of the software or to buy it from them.

This site may have errors. Don't use for critical systems.