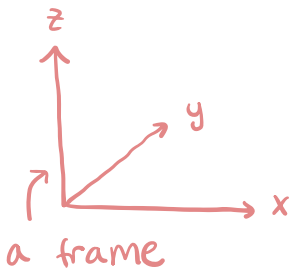
Week 6 - homogeneous transforms

In robotics we use **frames** (coordinate systems) to describe a robots position and movement. The frames are important for the robot to know where it is and how it should move in order to complete a task. We usually use 3 frames:



a frame

- **Base frame**: where the robot is placed - attached to the robot
- **Task frame**: where the robot is supposed to do a task
  → frame can f.ex be placed on an object in the world if the task is to pick up an objekt - frame is relative to task
- **World frame**: the entire workspace - a fixed reference frame.
- **Tool frame**: frame placed on the end effector

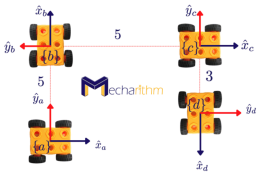
Homogeneous transforms let us go from one frame to another. They represent the position and orientation of a rigid body (a body that has a solid shape) in space. A homogeneous transform matrix combines both translation and rotation transformations.

→ Useful when determining how to move the end-effector to a desired position and orientation in space (task frame) starting from a known position and orientation (base).

→ Homogeneous transforms are for example used in forward kinematics, as you will see later.

→ We can use homogeneous transforms to translate any point represented in one frame to another frame.

→ Example:



Homogeneous transforms can be used to describe the relationship between configurations {a} and {b}, {b} and {c} and {d}. These can then be used to get the relationship between {a} and {d} which is the configuration {d} relative to frame {a}.

→ Homogeneous transformations are described by 4x4 matrices. Their general form is like this:

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}$$

→ R is a 3x3 rotation matrix
→ d is a 1x3 traslation vector

R = **rotation matrix** → describes how a frame is rotated by an angle.

→ We have 3 axis on our coordinate systems, so we have 3 rotation matrices: one for rotation around the x-axis, one for y-axis and one for z-axis. They are:

these are written in the formula sheet so you don't have to remember them

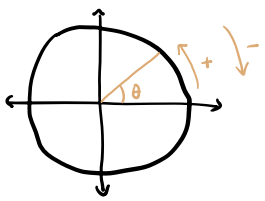
$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$ → rotating the x axis by an angle of θ.

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$ → rotating the y axis by an angle of θ.

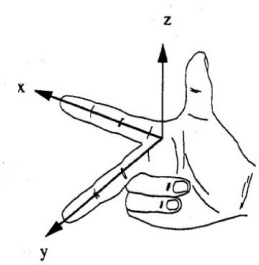
$$R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ → rotating the z axis by an angle of θ.

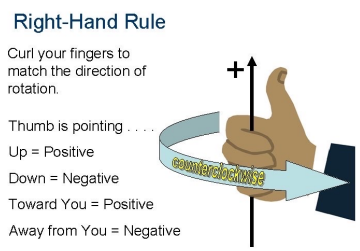
→ We often write cos and sin in a shorter way, to take up less space. We have $\cos\theta_1 = c_{\theta_1} = c_1$ and $\sin\theta_2 = s_{\theta_2} = s_2$

The axes can either be rotated clockwise or anti clockwise. The angle is positive when we rotate anti clockwise and negative when we rotate clockwise. Think about the unit circle:



We use the right hand rules to determine the angle and if its negative or positive. These are the rules:





d = translation vector → describes how far the origin has moved in space

→ this is described by a vector containing numbers about how the origin moves along each of the axes: x, y and z

$$d = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$ → translation in x direction
→ translation in y direction
→ translation in z-direction.

PS: H and T mean the same thing in this document! H stands for homogenous transforms and T for transformations, but these represent the same thing here!

How to do calculations with homogeneous transforms

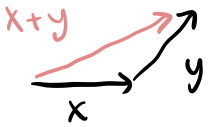
So let's see how we can use homogeneous transforms. In a simple example where we have a world frame for the environment around the robot, a base frame where the robot is standing and a task frame for where it should work. So usually what we want is the transformation between the base frame and task frame.

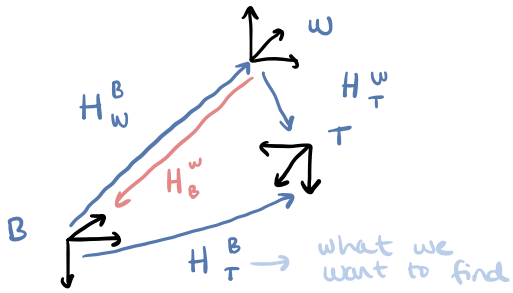
→ We can do this in two ways:

1) Directly find the transformation (rotation and translation) between base and task frame. Usually this is (a little) more complicated.

2) Go via the world frame (→ via a known reference point). This means calculating the transformation between base frame and the world frame, and then from world to the task frame.

→ the idea is that we can use the same method here as for vector addition (same way of thinking):





→ so in our example, to find H_T^B you need to find the transformation H_B^W and H_T^W.

we place the world frame in (0,0,0) to make it simple.

→ The reason why we use the world frame is because its origin is (0,0,0) so it's easy to find the translation from world origin to the base origin and task origin.

→ But to be able to to find the transformation between base and world (T_W^B), you first have to find the transformation between world and base (T_B^W) and then take the inverse. This is because we have to follow the way of thinking for vector addition.

→ When we have these transformations we can use a rule:

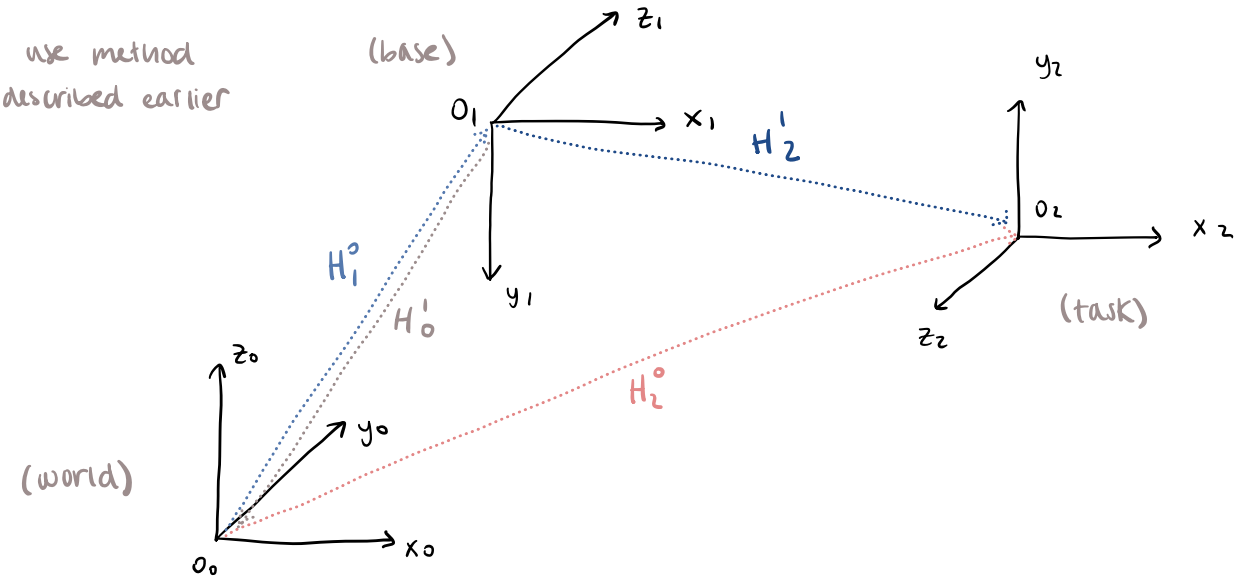
$T_y^x = T_z^x T_y^z$ → we simply multiply T_z^x and T_y^z (in this order).

→ we make a diagonal line (always in this direction, not \). z and z then cancel out so we end up with H_y^x).

The notation ${}^{x}T_{y}$ means the homogeneous transform from frame x to frame y. So, this is basically the rotation and translation of frame y with respect to frame x.

An example: we want to find ${}^{0}H_{1}$, ${}^{1}H_{0}$, ${}^{0}H_{2}$, and ${}^{1}H_{2}$

→ we use method 2) described earlier



We can set up the homogeneous transformation matrices like this:

$$H_1^0 = \begin{bmatrix} R_1^0 & d_1^0 \\ 0 & 1 \end{bmatrix} \qquad H_2^1 = \begin{bmatrix} R_2^1 & d_2^1 \\ 0 & 1 \end{bmatrix} \qquad H_2^0 = \begin{bmatrix} R_2^0 & d_2^0 \\ 0 & 1 \end{bmatrix}$$

To find H_0^1 we can use the inverse of H_1^0. This is given by the formula:

$$H_0^1 = (H_1^0)^{-1} = \begin{bmatrix} R_1^{0\,T} & -R_1^{0\,T} d_1^0 \\ 0 & 1 \end{bmatrix}$$

T means transposed

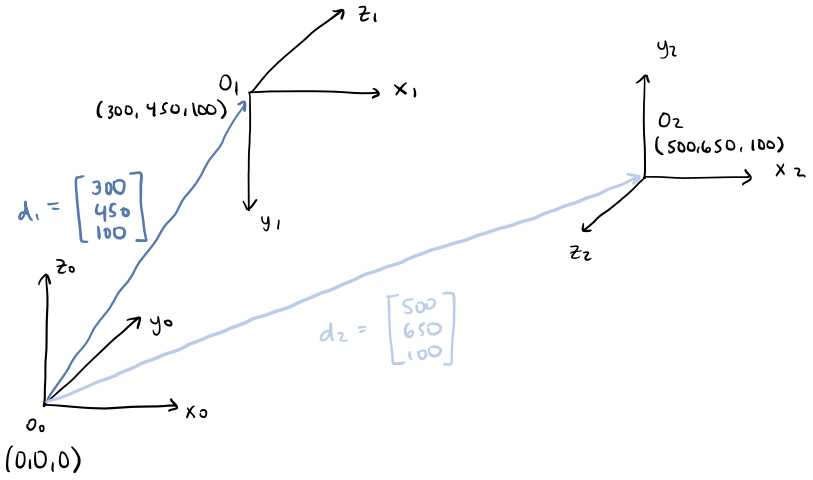
→ this formula is given in the formula sheet so you don't have to remember it

- R_1^0 is the rotation matrix from frame 0 to frame 1. This means how frame 0 rotates to match frame 1.
  ↳ you can also say that it is the rotation of frame 1 relative to frame 0.
- d_1^0 is a vector that describes how the frame moves in space so the origin o_0 matches O_1.
  ↳ you can also say that it's the origin of frame 1 relative to frame 0
- The same applies to R_2^1, R_2^0, d_2^1, d_2^0 for their respective frames

If we say that $o_0 = (0,0,0)$, $o_1 = (300, 450, 100)$, $o_2 = (500, 650, 100)$. what are the homogeneous transforms that we defined earlier?

The technique is to find the rotation matrix and then the translation vector for each homogeneous transform.

$$H_1^0 = \begin{bmatrix} R_1^0 & d_1^0 \\ 0 & 1 \end{bmatrix}$$



First we have to figure out about which axis the frames are rotated. From frames 0 and 1 we can see that x_0 and x_1 are parallel, which means that x_0 is not rotated, to match x_1, so the rotation must be around the x-axis.

We then use the right hand rules to figure out the angle and the sign of the angle. Following this, the angle of rotation between frame 0 and 1 is $-90°$ $(-\pi/2)$. We put this in the rotation matrix for the x-axis:

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \rightarrow R_1^0 = R_{x,-\pi/2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

The vector d is found by $o_1 - o_0$ which just gives o_1:

$$d = \begin{bmatrix} 300 \\ 450 \\ 100 \end{bmatrix}$$

We insert these into the homogeneous matrix formula to get:

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \rightarrow H_1^0 = \begin{bmatrix} 1 & 0 & 0 & 300 \\ 0 & 0 & 1 & 450 \\ 0 & -1 & 0 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation matrix

Translation vector

a convention - always like this

We do the similar calculations for H_2^0.

$$H_2^0 = \begin{bmatrix} R_2^0 & d_2^0 \\ 0 & 1 \end{bmatrix}$$

Frame the x-axis is rotated 90° ($\pi/2$) to match frame 1, so $\theta = \pi/2$.

$$R_2^0 = R_{x,\pi/2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$
(x_2, y_2, z_2)

frame 2 relative to frame 0:
- column vec 1 says x_2 and x_0 are the same
- column vec 2 says y_2 is the z from frame 0 (z_0)
- column vec 3 says z_2 is the old y (y_0) and negative

d_2^0 is found by $O_2 - O_0$ which gives $d_2^0 = (500, 650, 100)$.

$$H_2^0 = \begin{bmatrix} 1 & 0 & 0 & 500 \\ 0 & 0 & -1 & 650 \\ 0 & 1 & 0 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

this describes the position of frame 2 origin in terms of frame 1, so it is described in this coordinate system.

To find H_0^1 we use the formula for the inverse:

$$H_0^1 = (H_1^0)^{-1} = \begin{bmatrix} R_1^{0\,T} & -R_1^{0\,T} d_1^0 \\ 0 & 1 \end{bmatrix}$$

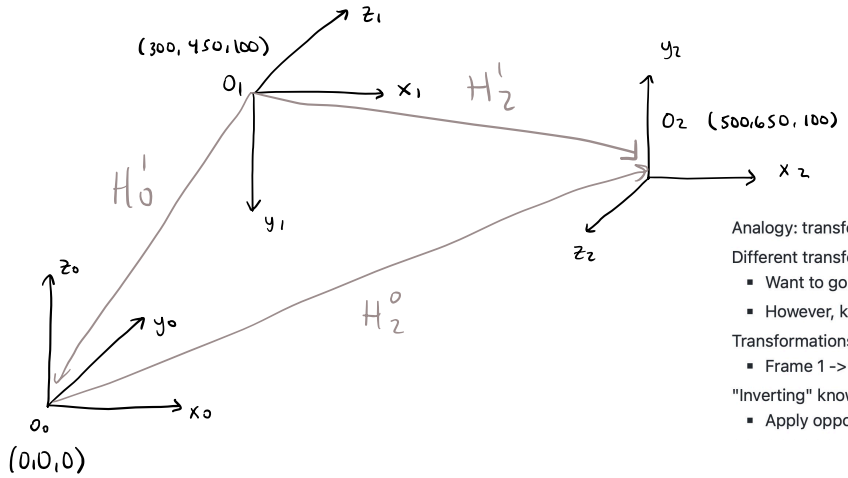
$$R_1^{0\,T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad -R_1^{0\,T} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$-R_1^{0\,T} d_1^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 300 \\ 450 \\ 100 \end{bmatrix} = \begin{bmatrix} -300 \\ 100 \\ -450 \end{bmatrix}$$

$$H_0^1 = \begin{bmatrix} 1 & 0 & 0 & -300 \\ 0 & 0 & -1 & 100 \\ 0 & 1 & 0 & -450 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To find H_2^1, then we use the "vector addition method" from above along with the formula $T_y^x = T_z^x T_y^z$. This is how we think:

$$H_2^1 = \begin{bmatrix} R_2^1 & d_2^1 \\ 0 & 1 \end{bmatrix}$$



Analogy: transformation from one frame to another - going between two places (example: home -> school)

Different transformation paths possible. Example:

- Want to go from school to the store, but direct path is unknown.
- However, know these paths: school -> home **and** home -> store .

Transformations between frames are similar, all paths are valid:

- Frame 1 -> Frame 3 **OR** Frame 1 -> Frame 2 -> Frame 3

"Inverting" known transformation is possible:

- Apply opposite rotation, then opposite translation

So to find T_2^1 we need to use T_0^1 and T_2^0 because by the formula given above we get

$T_2^1 = T_0^1 T_2^0$ → we can find T_2^1 by simply multiplying the matrices T_0^1 and T_2^0 together

We do some matrix multiplication:

$$H_2^1 = \begin{bmatrix} 1 & 0 & 0 & -300 \\ 0 & 0 & -1 & 100 \\ 0 & 1 & 0 & -450 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 500 \\ 0 & 0 & -1 & 650 \\ 0 & 1 & 0 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 200 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 200 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

→ how to go from frame 1 to 2 / frame 2 described relative to frame 1

PS: If you have to solve a task where the frames are rotated around more than one axis, you can multiply the rotation matrices together:

$R = R_{y,\phi} R_{z,\theta}$ → here a frame is rotated an angle ϕ around the y-axis and then around z-axis at an angle θ.

→ the order of multiplication must be the same as order of rotation

If you have a point p^0 represented in frame 0 and you want to represent it in frame 1 (as p') you can use the rotation matrix between the frames like this: $p' = R_0^1 p^0$

If you have to take both rotation and orientation into account the formula for finding a point with respect to one frame (A) when it is defined in another frame (B):

$p^A = T_B^A p^B$ → you use the transformation matrix

Compositions of rotations

→ consecutive rotations of a frame with respect to different axes.

A frame can be rotated with respect to more than one axis, so when we want to find the rotation matrix, we have to separate the rotations to find the rotation matrix with respect to one axis and then the other... Then all these are multiplied, but the order matters.

We can rotate the axis with respect to a current frame or fixed frame and this affects how we multiply the rotations together.

- The meaning of **current frame** is that when you do a first rotation then you have a new orientation for the frame and when you do another rotation, it is with respect to this current new orientation.

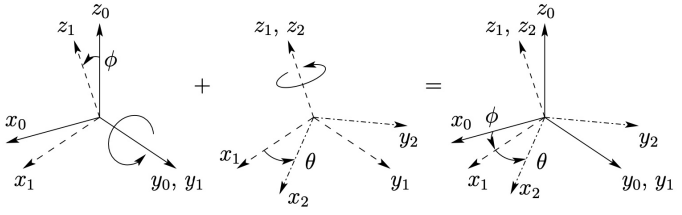
  → Then you **post multiply** → multiply the new rotation matrices after eachother.

  → This is what we would do if our frame rotations were composed of more than one rotation with respect to each other.

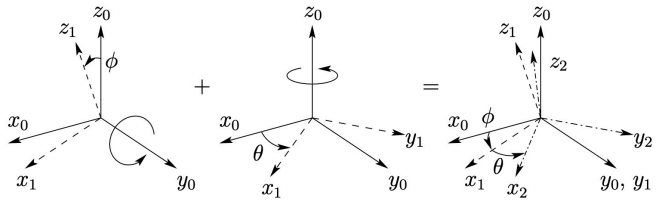
- The meaning of **fixed frame** is that you do all rotations with respect to a fixed frame. This means that you do a first rotation to obtain a new orientation of the next frame. The second rotation is <u>not</u> done with respect to this new orientation, but with respect to the original (fixed) frame.

  → Then you **pre multiply** → multiply the new rotation matrices before eachother (we get them in reverse order)

  → We don't use this soo much, but it's good to know.



↳ current frame rotation



↳ fixed frame rotation

For more info about this, read in the book and do the tasks for this week!

Parametrizations of rotations

3 ways in which a rotation can be represented using only 3 independent variables:

1) **Euler angle representation**: Any set of rotation can be described by

   1) a rotation by ϕ about the z-axis.
   2) a rotation by θ about the current y-axis
   3) a rotation ψ about the current z-axis.

   This is given by the formula:

$$R_{ZYZ} = R_{z,\phi} R_{y,\theta} R_{z,\psi} = \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(postmultiplying)

$$= \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}$$

(ZYZ - euler angle transformation)

2) **Roll, pitch, yaw angles**: Any set of rotations can be described by 3 rotations around the fixed x, y, z-axes.

   - yaw - x_0 - ψ (rotate x_0 about an angle ψ)
   - pitch - y_0 - θ (rotate y_0 about an angle θ)
   - roll - z_0 - ϕ (rotate z_0 about an angle ϕ)

   This is given by the formula:

$$R_{XYZ} = R_{z,\phi} R_{y,\theta} R_{x,\psi}$$

$$= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix}$$

(pre-multiplying)

$$= \begin{bmatrix} c_\phi c_\theta & -s_\phi c_\psi + c_\phi s_\theta s_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\phi c_\theta & c_\phi c_\psi + s_\phi s_\theta s_\psi & -c_\phi s_\psi + s_\phi s_\theta c_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}$$

3) **Axis/angle representation**: Any set of rotations can be achieved by a single rotation about a suitable axis in space by a suitable angle. This is given by the formula:

$$R_{k,\theta} = \left[\begin{array}{ccc|ccc|ccc} k_x^2 v_\theta + c_\theta & & & k_x k_y v_\theta - k_z s_\theta & & & k_x k_z v_\theta + k_y s_\theta & & \\ k_x k_y v_\theta + k_z s_\theta & & & k_y^2 v_\theta + c_\theta & & & k_y k_z v_\theta - k_x s_\theta & & \\ k_x k_z v_\theta - k_y s_\theta & & & k_y k_z v_\theta + k_x s_\theta & & & k_z^2 v_\theta + c_\theta & & \end{array}\right]$$

k - unit vector defining the axis
θ - angle of rotation around k

Basic homogeneous transforms:

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}$$

$$\text{Trans}_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Rot}_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Trans}_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Rot}_{y,\beta} = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Trans}_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Rot}_{z,\gamma} = \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

only describes translation

only describes rotation

The idea is that we can combine these basic homogeneous matrices to find the homogeneous transformations in another way than we did earlier. F.ex if we were to find H_1^0 from our example, we would have to: find the rotation angle and axis and the translation in x, y and z direction, put this info in the matrices and multiply.

$$H_1^0 = \text{Rot}_{x,-90}\ \text{Trans}_{x,300}\ \text{Trans}_{y,450}\ \text{Trans}_{z,100}$$