# Theory of Robotics & Mechatronics

Institute of Robotics and Intelligent Systems

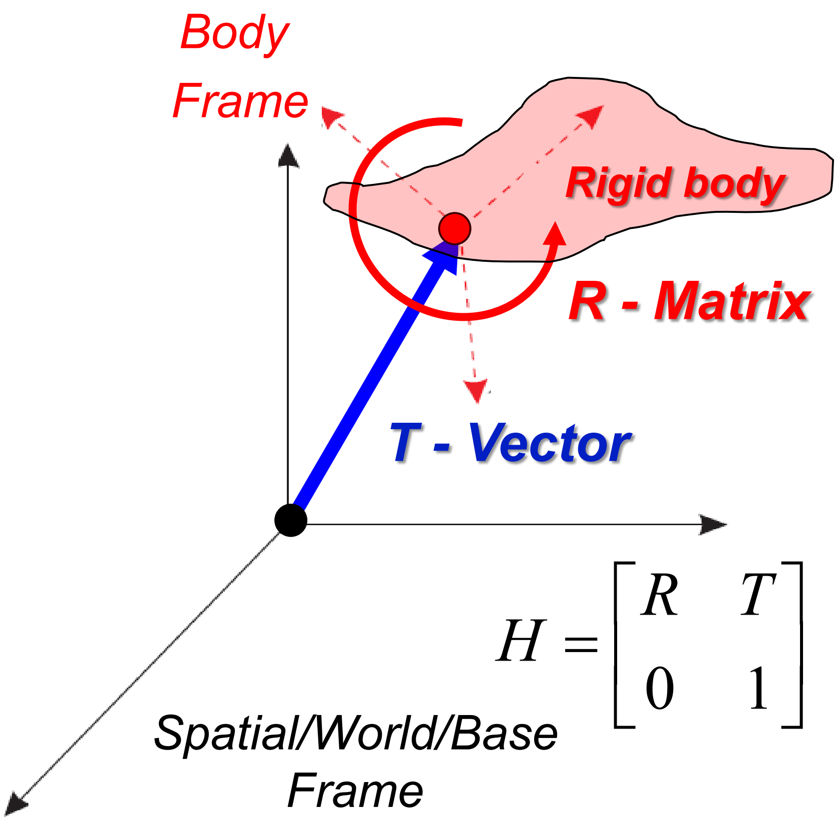
## Lecture: Spatial Descriptions 2 (Screw Theory)

**P. Korba**

Institute of Robotics and Intelligent Systems
ETH Zurich

# Repetition – Lecture SD 1

SD1 – Conventional formulation // You need these mathematical tools & some spatial imagination




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

$$H^{-1} = \begin{bmatrix} R^T & -R^T T \\ 0 & 1 \end{bmatrix}$$

***T* – linear position/translation**
- **– based on vector algebra**
- – Simple !

***R – Orientation / Rotation***
- **– based on matrix algebra**
- – not so simple
- – Problems start with:

  **M1 * M2 = M2 * M1**
- – corresponds to rotations in fix or current frames

***H – Homogeneous transformation***
- – We put both R & T together
- – Elegant and efficient way how to work
- – widely spread in robotics

# Warming up – Lecture SD 2

SD2 – Modern technique / Screw theory // You need these mathematical tools & much less spatial imagination than with the conventional tools SD1

- Important terms to learn and to understand today:

(When you see/hear this, increase your attention – you do not have to understand it now but after this lecture)

  - Existence of matrix exponential and how to calculate it -- $e^{at}$ vs. $e^{At}$
  - What is a skew-symmetric (or anti-symmetric) matrix
  - Axis of rotation $\omega$ & How to encode it in a skew symmetric matrix $\widehat{\omega}$
  - How to get quickly a rotation matrix ***R*** for a given axis of rotation $\omega$ and angle $\theta$

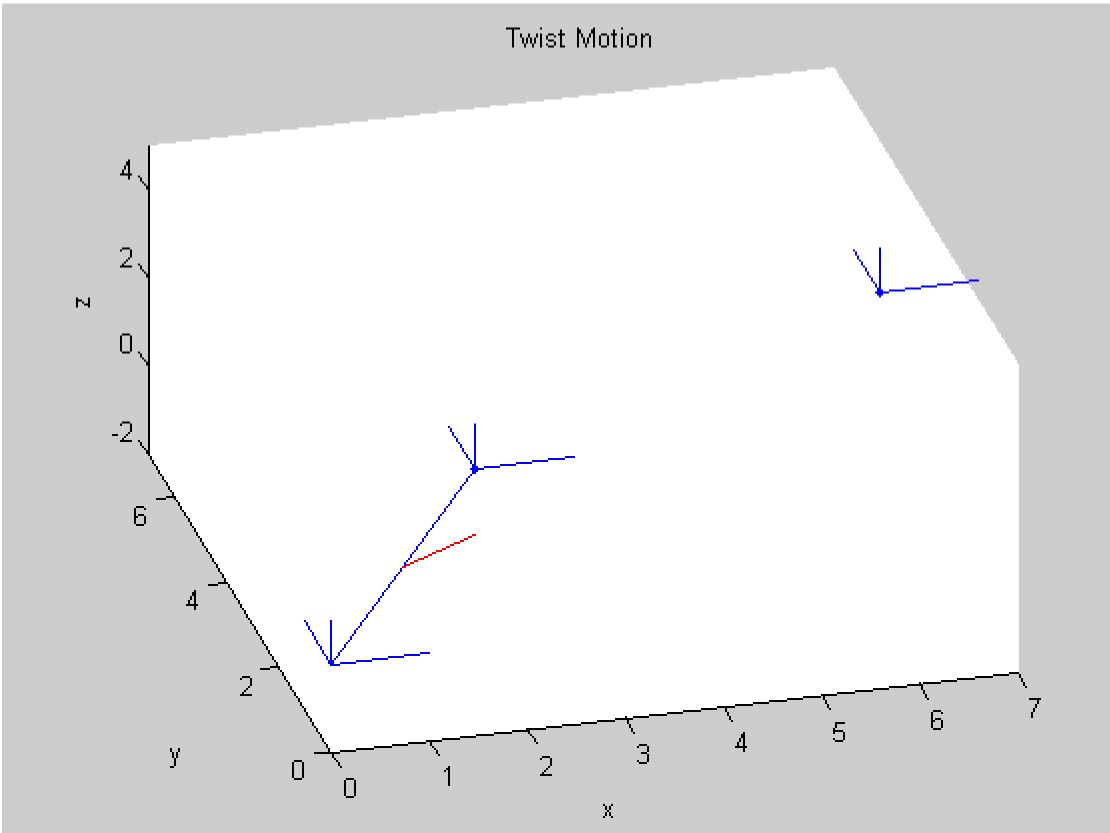
$$\boxed{R = e^{\hat{\omega}\theta}}$$

  - What is a twist $\xi$, pitch "h", axis "l" and the magnitude "M"
  - How to get twist coordinates $\xi$ and the corresponding twist $\hat{\xi}$
  - How to get complete rigid body motion (the same as the homogeneous transformation)

$$\boxed{H = g(\theta) = e^{\hat{\xi}\theta} g(0)}$$

# Screw Theory

*Based on Twists and "Screw Motions"…*




- Base frame
- Screw axis
- Starting point (with its local frame)
- Final point (with its local frame)
- Animation shows a screw motion with the magnitude of 1080 deg (3x)

- Do you see a robot behind this ?

# Screw Theory (ST) in Robotics

## *Today's lecture: Quick introduction to ST from "User Point of View"*

- There are alternatives for spatial descriptions in robotics
  - Recall drawbacks we observed in classical approaches
    - Euler angles → one singularity in any sequence/representation
    - Angle-axis → axis not always defined (for th=0 $\pm k\pi$)
  - e.g. not easy to program a very generic function based on above approaches (recall Q1) and this motivates to invent another approach…

- ***Screw Theory*** is mathematically elegant and powerful
- Relatively new to robotics, many robotics researchers are not familiar with it
- A lot of different words are used to described it:
  (a) ***Screws*** (b) ***Twists*** (c) ***Product of matrix exponentials*** (POE)
- The entire method is based on understanding of rigid-body ***motion*** rather than ***location*** as it was the case with HT

  Observe:
  - ***Time*** was not included in previously introduced (homogeneous) transformations.
  - Since the screw theory will introduce the notion of time through a differential equation, we move "*from location to motion*" now.

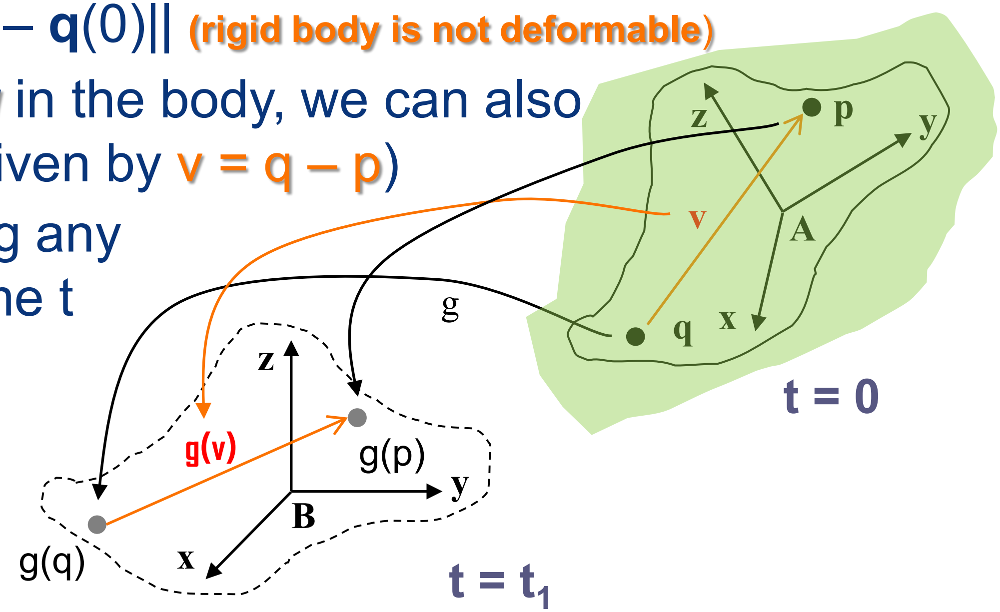
# Classical Formulation vs. Screw Theory (ST)

- ***Classical formulation*** is focused on transforming points & vectors in ***one (arbitrary) robot-link frame with respect to another*** *(different) robot-link frame*, based on the current configuration
- ***This method (ST)*** essentially treats each robot link as equally important (***all done directly in base (world/spatial) frame***)
- We are typically most interested in the last robot-link frame, usually called the ***tool frame***
- Screw theory is ***very good for direct description*** of the configuration and motion of ***the tool frame with respect to the base coordinates*** (this will become more evident when we discuss rigid-body velocity)
- In general, the other robot links are just elements/steps which position the tool frame and we have less interest in them (although we could always get this information if necessary, see later the demo).
- Perhaps, this can be seen as an disadvantage of ST … ☺

# Rigid Body Motion

## Introductory facts you already understand...

- Forget robots for a while, and just consider a ***rigid body*** in space
- ***Attach a body frame*** to the body to describe its pose (3rd approach)
- ***Move the body*** from its initial location at time t = 0 to time = t1
- For example, the initial frame A be the ***base/spatial/reference frame***, from which we describe the pose of the body throughout time
- Any two points (**p** and **q)** on the rigid body have always the same distance $||\mathbf{p}(t) - \mathbf{q}(t)|| = ||\mathbf{p}(0) - \mathbf{q}(0)||$ **(rigid body is not deformable)**
- Given any two points ***p*** and ***q*** in the body, we can also find a vector ***v*** from ***p*** to ***q*** (given by $v = q - p$)
- There is a function ***g*** mapping any point to its new location at time t
- And, therefore it also maps embedded vectors:
  $g(\underline{\mathbf{v}}) = g(\mathbf{q}) - g(\mathbf{p})$

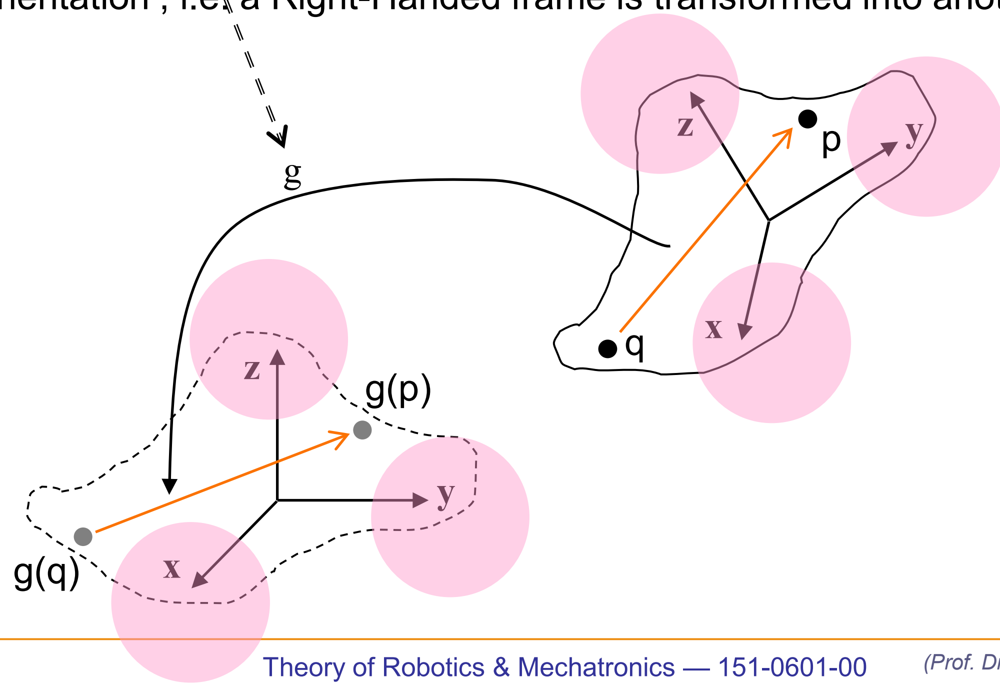


# Rigid Body Transformation

## Definition: When do we call a mapping g a rigid body transformation?

- A mapping **g** is ***a rigid body transformation*** if it satisfies the following properties:
  - The ***length (i.e. distances between points)*** is preserved:
    ||g(**p**) - g(**q**)|| = ||**p** - **q**|| for all points **p** and **q**
  - The ***cross product*** is preserved:
    g(**v** x **w**) = g(**v**) x g(**w**) for all vectors **v** and **w**

(i.e. orientation , i.e. a Right-Handed frame is transformed into another R-H frame)



# Linear Form of Cross Product
# How to get it?

- The cross (vector) product between two vectors is defined as

$$a \times b = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix} = \boxed{\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}} b = \hat{a} b = (a)^\wedge b$$

Axis of rotation "omega"

- (**a**)^ is a skew-symmetric matrix built from the elements of **a**
- A skew-symmetric matrix S is defined by $S + S^T = 0$ or $S = -S^T$
- We will also use the equivalent notation a^ (wedge): $\hat{a} = (a)^\wedge$

- How to calculate *a x b* ? ➔ det([x y z; a1 a2 a3; b1 b2 b3])
  or use the above linear transform skew(a)*b
  which preserves the geometric meaning

$$\boxed{a \times b = \hat{a} b}$$

# Exponential Coordinates for Rotation

## Why will we deal with **Matrix Exponentials** and **Init. Cond?**

- If we ***rotate a body at constant unit velocity*** about axis $\omega$, where $\|\omega\| = 1$, the velocity of the point **q** is written as

$$\boxed{\dot{q}(t) = \omega \times q(t) = \hat{\omega} q(t)}$$

*You know very well that $v = \omega r$*
*(but what about its spatial representation?)*

- This ***time-invariant linear differential equation*** can be integrated

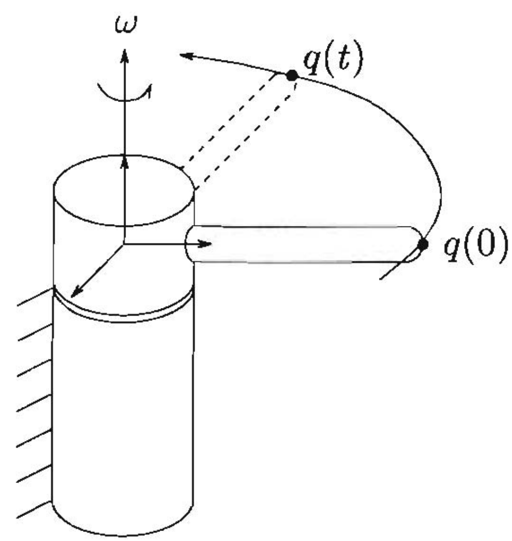
$$q(t) = e^{\hat{\omega} t} q(0)$$

- We can make use of ***series expansion*** of the matrix exponential

$$e^{\hat{\omega} t} = I + \hat{\omega} t + \frac{(\hat{\omega} t)^2}{2!} + \frac{(\hat{\omega} t)^3}{3!} + \cdots$$

- Considering ***$\theta$ units of time***, we compute the net rotation as

$$R(\omega, \theta) = e^{\hat{\omega} \theta}$$



# Exponential Coordinates for Rotation
## Overview

- The matrix exponential – $e^{\hat{\omega}}$ of a skew-symmetric matrix is a ***rotation matrix R***
- The ***axis of rotation*** is the vector encoded in the skew-symmetric matrix, normalized to unit length, **i.e. our vector omega where** $\|\omega\| = 1$
- The ***magnitude of rotation*** (angle in radians) is the length of the vector encoded in the skew-symmetric matrix, **i.e. our theta we added to exponent**
- All ***rotation matrices $R(\omega,\theta)$*** can be represented by a matrix exponential of a skew-symmetric matrix
- There is a ***simple closed form for the matrix exponential***, known as ***Rodrigues' formula*** (where $\|\omega\| = 1$ ):

$$e^{\hat{\omega}\theta} = I + \hat{\omega}\theta + \frac{(\hat{\omega}\theta)^2}{2!} + \frac{(\hat{\omega}\theta)^3}{3!} + \cdots = \boxed{I + \hat{\omega}\sin\theta + \hat{\omega}^2(1-\cos\theta)}$$

# Exponential Coordinates for Rotation
# Olinde Rodrigues (1795-1850)

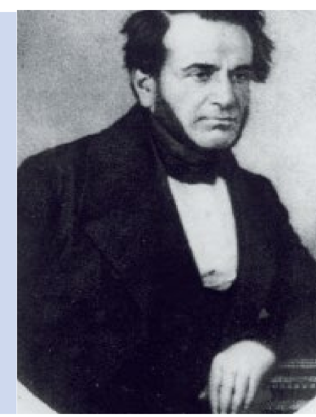
- Converted a vector and an angle into a rotation matrix → as simply as possible
  (formula goes back to properties of orthonormal matrices and its eigenvectors & -values)

- Observed the interesting fact that any arbitrary rotation can be parameterized by 4 numbers (3 for axis & 1 for angle of rotation). However, a unit vector can be used (with only 2 parameters) since

$$\omega_3 = \sqrt{1-\omega_1^2-\omega_2^2}$$

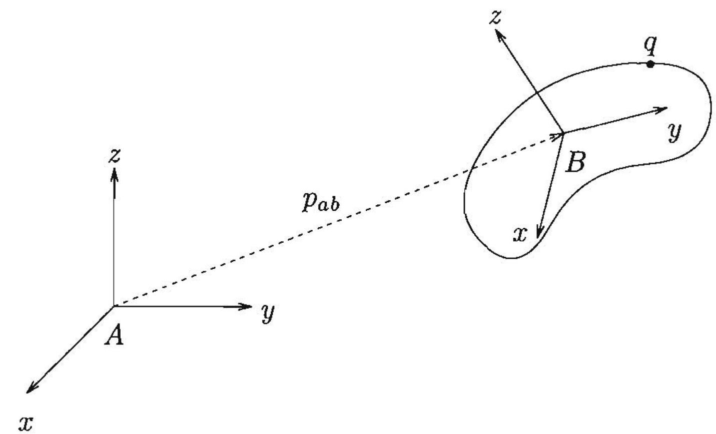
- Alternatively, we can multiply the vector with the angle to get another 3 parameter representation: $\omega\theta$

**Olinde Rodrigues (1795–1850)** was a French-Jewish banker and mathematician who wrote extensively on politics, social reform and banking. He received his doctorate in mathematics in 1816 from the University of Paris, for work on his first well known formula which is related to Legendre polynomials. His eponymous rotation formula was published in 1840 and is perhaps the first time the representation of a rotation as a scalar and a vector was articulated. His formula is sometimes, and inappropriately, referred to as the Euler-Rodrigues formula. He is buried in the Pere-Lachaise cemetery in Paris.



# Rigid Motions
# Groups: SE(3), SO(3) and O(3)

- But we want to ***completely describe*** the pose of the rigid body wrt an inertial reference frame, ***i.e. we have to deal with both rot. & translat.***
- Given by the ***orientation*** of the frame $R_{ab}$ and the ***position*** $p_{ab}$
- Let $q_a$ and $q_b$ be the coordinates of a point q relative to frames A and B, where $q_a = R_{ab}q_b + p_{ab}$ **(recall the 1st lecture)**
- We can also think of the configuration as an abstract item on its own, which contains two pieces of information:
  ***$g_{ab} = (p_{ab}, R_{ab})$ i.e. translation & rotation***
- We call this abstract object ***$g_{ab}$*** a member the 3D ***special Euclidean group SE(3)***
  - Rotation $R_{ab}$ is a member of ***SO(3)***
    ***Special Orthogonal group***
    - rotations in a right-handed coordinate frame det(R)=1;
  - ***Orthogonal group - O(3)*** as SO(3) but det(R)=±1
- We also think of ***$g_{ab}$ to be rigid transformation*** of a point: $q_a = g_{ab}(q_b)$



# Homogeneous Representation
# Extension of Points and Vectors

- ***Rigid transformations*** are <u>not linear but affine</u>, i.e. $q_a = R_{ab} q_b + p_{ab}$
- Can we construct a 'more' ***linear version*** of these motions?
- ***Yes - exactly as we did in the previous lecture on homogen. trans. !***
  We append systematically 1 to the coordinates of a point, and 0 to a the coordinates of a vector (both to become 4x1)
- Thus, we will work with ***vectors & points*** – however, we distinguish and treat them slightly differently:

point
$$\bar{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ 1 \end{bmatrix}$$

vector
$$\bar{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ 0 \end{bmatrix}$$

- ***Why?*** The above syntax helps us to reflect physical realities:
  - Sums and differences of vectors are vectors
  - The sum of a vector and a point is a point
  - The difference between two points is a vector
  - The sum of two points is meaningless

HINT: Have always a look into the 4th "extended" coordinate 'maintained' by convention 1 or 0 !

# Homogeneous Representation
## (From skew-matrix exponential – i.e. Rotation – to a full twist)

- Now we can express a ***rigid transformation*** $q_a = R_{ab}q_b + p_{ab}$ as a linear multiplication of a 4x4 matrix with a 4x1 vector

$$\bar{q}_a = \begin{bmatrix} q_a \\ 1 \end{bmatrix} = \begin{bmatrix} R_{ab} & p_{ab} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_b \\ 1 \end{bmatrix} \equiv g_{ab}\bar{q}_b$$

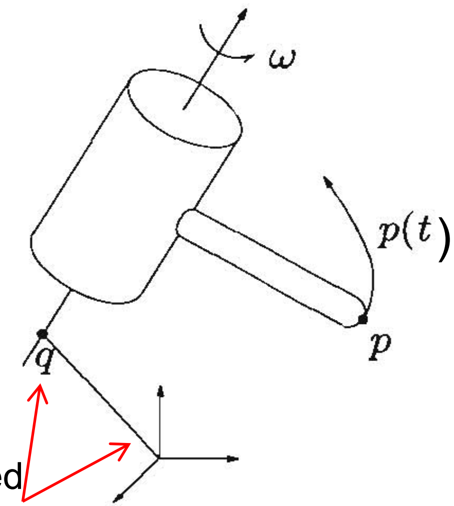
- We already found that a 3x3 ***rotation matrix*** is the matrix exponential of a 3x3 skew-symmetric matrix

- ***Note:***
  - **There is a 4x4 equivalent of a skew-symmetric matrix (we had it before to describe only rotations), called a *twist*.**
  - **The *matrix exponential of the twist* is a 4x4 homogeneous transformation matrix (which as we already know describes both rotation and translation)**

- Consider a one-link robot (or just a point *p*), rotating about axis $\omega$ (with $||\omega||=1$ , i.e. with a unit velocity) where $\omega$ is passing a point *q*)

$$\dot{p}(t) = \omega \times (p(t) - q)$$

$$\begin{bmatrix} \dot{p} \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{\omega} & -\omega \times q \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \hat{\xi} \begin{bmatrix} p \\ 1 \end{bmatrix} \Rightarrow \dot{\overline{p}} = \hat{\xi}\overline{p}$$




where $\boxed{\hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}, \quad v = -\omega \times q}$

Translation induced by rotation if its axis $\omega$ is not passing the origin

- Integrating gives

$$\overline{p}(t) = e^{\hat{\xi}t}\,\overline{p}(0) \quad \text{where} \quad e^{\hat{\xi}t} = \begin{bmatrix} e^{\hat{\omega}t} & (I - e^{\hat{\omega}t})(\omega \times v) \\ 0 & 1 \end{bmatrix}$$

Form for use with Rodrigues' formula given in terms of $e^{\hat{\omega}t}$

# The Twist – Special Case 2

## *Translation*

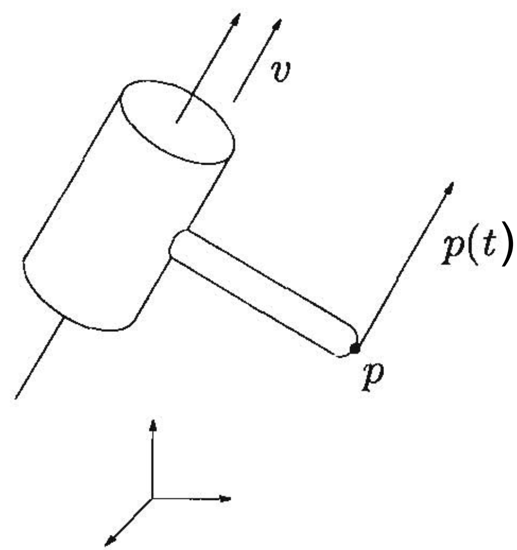
- Consider a one-link robot, sliding on a prismatic joint with unit velocity $v$ :

$$\dot{p}(t) = v$$

$$\begin{bmatrix} \dot{p} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \hat{\xi} \begin{bmatrix} p \\ 1 \end{bmatrix} \Rightarrow \dot{\overline{p}} = \hat{\xi}\,\overline{p}$$

$$\boxed{\hat{\xi} = \begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix}}$$

Observe that
- no point $q$ is required
- linear velocity vector $\mathbf{v}$ has no position only direction and magnitude




- Integrating gives

$$\overline{p}(t) = e^{\hat{\xi} t}\,\overline{p}(0) \quad \text{where} \quad e^{\hat{\xi} t} = \begin{bmatrix} I & vt \\ 0 & 1 \end{bmatrix}$$

# The Twist and Screw Motion
# Structure, Names, Notations

- A ***twist*** is a 4x4 matrix of the structure

$$\hat{\xi} = \begin{bmatrix} \underbrace{\hat{\omega}}_{\text{3x3 skew-symmetric matrix}} & \underbrace{v}_{\text{3x1 vector}} \\ \underbrace{0}_{\text{1x3 matrix of zeros}} & 0 \end{bmatrix}$$

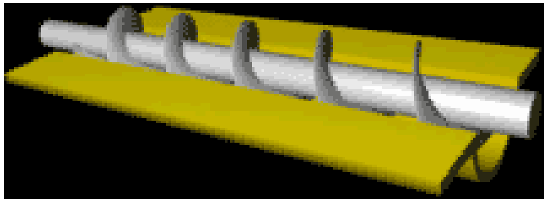
- We can also extract the 6x1 ***twist coordinates*** $\xi$

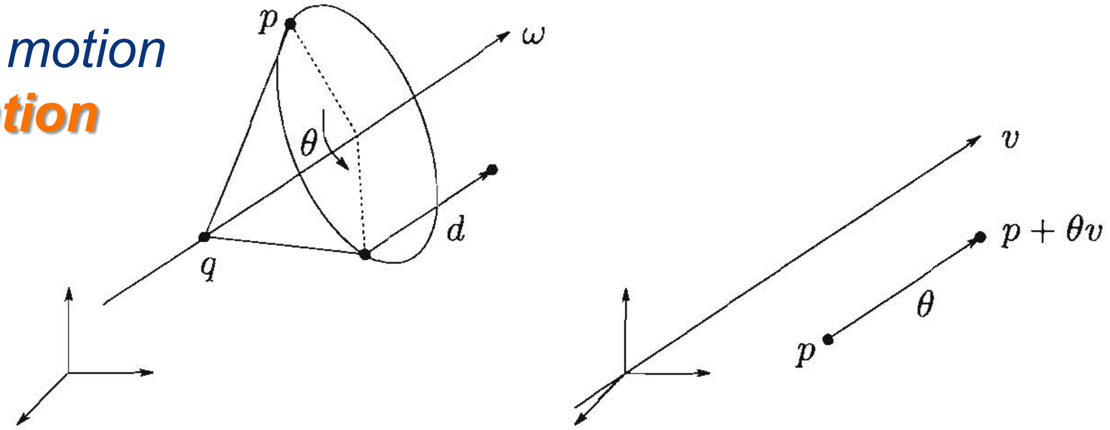
$$\xi = \hat{\xi}^{\vee} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}^{\vee} = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

- The twist is a characteristic of ***screw motions***
- ***Matrix exponential*** maps twist into screw motion

# Screw Motions
# Description by words ...

- *Theorem (Michel Chasles):*
  ***Every rigid body motion can be realized by a rotation about an axis combined with a translation parallel to that axis ("screw")***
- *A **screw** 'S' can be described by **axis** 'l', **pitch** 'h', and **magnitude** 'M'*
- *A **screw motion** represents rotation by an amount M about axis l followed by translation by an amount $h\theta$ parallel to axis l*
- *If h = ∞ then the screw motion consists of a pure **translation** along the axis 'ω' by a distance 'M'*
- *If h = 0 then the screw motion consists of a pure **rotation** about the axis 'l' by a distance 'M'*






(a) general screw (b) pure translation

# Screw Motions Mathematically…

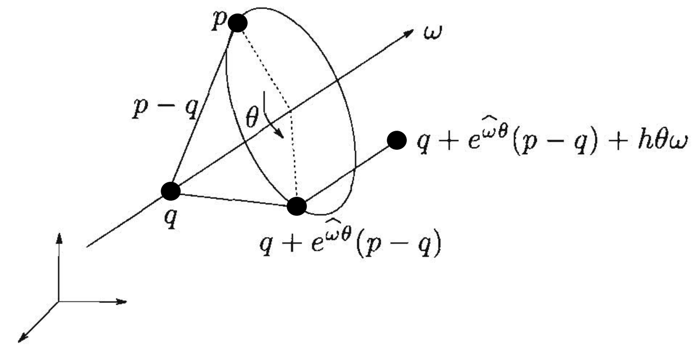
Linear offset

Rotation

Trans. because of rot. "Screw"

- ***Description*** of final location of the point ***p:***

$$gp = q + e^{\hat{\omega}\theta}(p-q) + h\theta\omega$$




or, in ***homogeneous coordinates:***

$$g\begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} e^{\hat{\omega}\theta} & (I - e^{\hat{\omega}\theta})q + h\theta\omega \\ 0 & 1 \end{bmatrix}\begin{bmatrix} p \\ 1 \end{bmatrix}$$

- This has the same form as the ***exponential of this twist*** (written only by means of $\boldsymbol{\omega}$ and $\boldsymbol{v}$ )

$$e^{\hat{\xi}\theta} = \begin{bmatrix} e^{\hat{\omega}\theta} & (I - e^{\hat{\omega}\theta})(\omega \times v) + \omega\omega^T v\theta \\ 0 & 1 \end{bmatrix}$$

$$v = -\omega \times q + h\omega, \quad \|\omega\| = 1, \quad \omega \neq 0$$

- In case of pure ***rotation*** ($h=0$), $\theta$ is the angle, and $g$ is simply:

$$\boxed{e^{\hat{\xi}\theta} = \begin{bmatrix} e^{\hat{\omega}\theta} & (I - e^{\hat{\omega}\theta})(\omega \times v) \\ 0 & 1 \end{bmatrix}}$$

***<u>In robotics</u>:***
***only prismatic & revolute joints,***
***i .e. we need to be able to describe***
***only translations & rotations…***

# Calculating a Screw from a Twist

*When we say "**screw**", we mean the solution in time-domain or how to get [h, l, M] from [ω,v ] – here, **less important***

- ***Pitch:*** The pitch of a twist is the ratio of translational motion to rotational motion. If $\omega$ = 0, we say $\xi$ has infinite pitch (= translation)

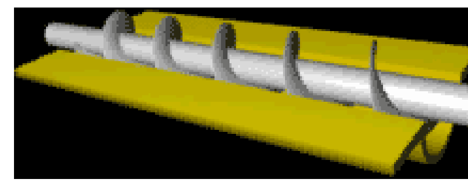
*We do not need to assume that $\|\omega\|^2 = 1$ as long as we can scale…*

$$h = \frac{\omega^T v}{\|\omega\|^2}$$

Scalar

**Screw coordinates**

[Pitch, Axis, Magnitude]



- ***Axis:*** The axis *l* is a directed line through a point **q**

$$l = \begin{cases} \frac{\omega \times v}{\|\omega\|^2} + \lambda\omega : \lambda \in \Re, & if\ \omega \neq 0 \\ \lambda v : \lambda \in \Re, & if\ \omega = 0 \end{cases}$$

Vector

Translation:
$\omega$ = 0 and therefore $h = \infty$
linear velocity does not have any position (only magnitude and direction, i.e. $v$ is a free vector )

- ***Magnitude:*** The magnitude of a screw is the net rotation if the motion contains a rotational component, or the net translation otherwise

$$M = \begin{cases} \|\omega\|^2, & if\ \omega \neq 0 \\ \|v\|^2, & if\ \omega = 0 \end{cases}$$

Scalar

(Translation: $h = \infty$ and $\omega = 0$)

# Calculating a Twist from a Screw

*When we say "**twist**", we mean the input for the matrix exp., i.e. how to get $\hat{\xi}$ - **very important***

- $h = \infty$: Let $l = \{\lambda v : ||v|| = 1, \lambda \text{ real}\}$, $\theta = M$ and define

$$\boxed{\hat{\xi} = \begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix}}$$

Translation
(h=inf and omega = 0)

the rigid body motion expm($\xi$^$\theta$) corresponds to pure translation along the screw axis by an amount $\theta$

- $h$ = *finite*: Let $l = \{q + \lambda\omega : ||\omega|| = 1, \lambda \text{ real}\}$, $\theta = M$ and define

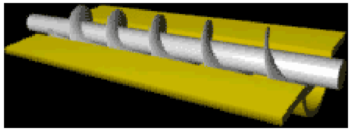
$$\hat{\xi} = \begin{bmatrix} \hat{\omega} & -\omega \times q + h\omega \\ 0 & 0 \end{bmatrix}$$

Rotation
(h=0)

*In general, axis of rotation can go through a point 'q' (offset/translation from the origin)*
*If we rotate about axis going through the origin then (Euler's angles are enough to use or) twist $\xi = [v\ \omega]^T$ with $v$ =[0 0 0] (this twist does not generate any linear velocity and* expm($\omega\theta$) ⇔ expm($\xi\theta$)

the appropriate screw motion is generated again by expm($\xi$^$\theta$)

***Twist coordinates***
$\xi=[v,\omega]^T$



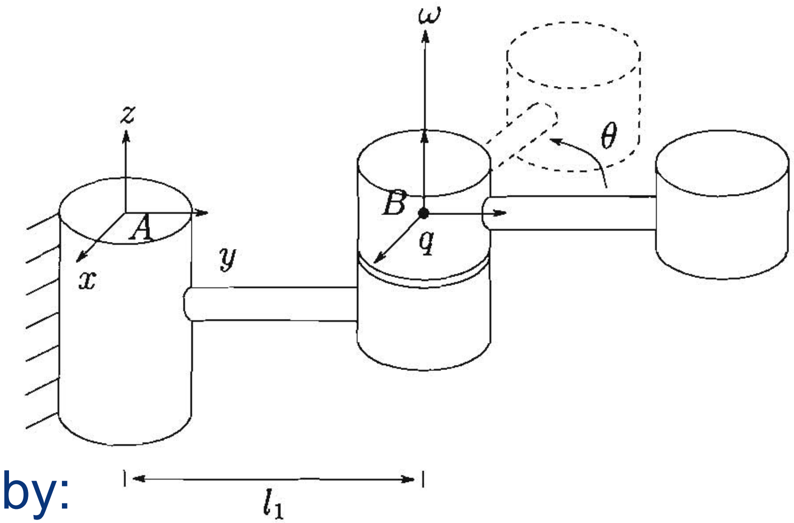
# Example: Rotation about any axis

## (axis not passing the origin Step 1 of 2: Get the twist)

- Consider motion corresponding to a zero-pitch screw (i.e. pure ***rotation,*** h=0) about an axis $\omega$ = (0,0,1) = z-axis (a free vector) passing through point $q = (0, l_1, 0)$. The corresponding ***twist*** is : (***twist coordinates***)

$$\xi = \begin{bmatrix} -\omega \times q \\ \omega \end{bmatrix} = \begin{bmatrix} l_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Important Observation: we describe point q and axis $\omega$ using directly the base frame A !!!




- The exponential of the twist is given by:
(in Matlab $e^{\hat{\xi}\theta}$ , otherwise as below using $e^{\hat{\omega}\theta}$ & Rodrigues)

$$\boxed{e^{\hat{\xi}\theta} = \begin{bmatrix} e^{\hat{\omega}\theta} & (I - e^{\hat{\omega}\theta})(\omega \times v) \\ 0 & 1 \end{bmatrix}} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & l_1 \sin\theta \\ \sin\theta & \cos\theta & 0 & l_1(1-\cos\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Example: Rotation about any axis
## (axis not passing the origin Step 2 of 2: Get the initial condition)

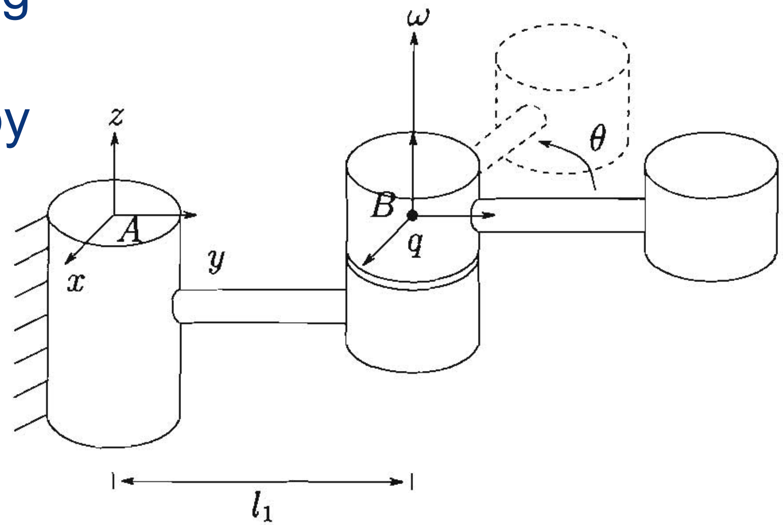
- The rigid transformation which maps points in B coordinates to A coordinates --- and hence describes the configuration of the rigid body --- is given by

$$g_{ab}(\theta) = e^{\hat{\xi}\theta} g_{ab}(0) \quad \textbf{where} \quad g_{ab}(0) = \begin{bmatrix} I & \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix}$$

What is init. cond. ?
for θ = 0, i.e.
g(θ) = [R(θ) q
0 0 0 1]

- Taking the exponential and performing the matrix multiplication yields the configuration, which can be verified by inspection in this case (recall HT):

$$g_{ab}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



# Example: Rotation about any axis
# Solution in Matlab - Simple code (1/2)

```
syms l1 th real  % define symbolic vars as "real"  (otherwise all considered complex)
% 1. Define axis of rotation as a free vector and select a point it goes through:
q  = [0 ;  l1 ; 0 ]
w = [0 ;  0  ;  1]
% 2. Define v & xi according to your slides:
 v  = cross( - w , q)
 xi = [v ; w]
% 3. Get 4x4 twist matrix from the above 6x1 vector xi called "Twist Coordinates"
xi_  = twist(xi)  % i.e. = [skew(w), v ; zeros(1,4)]
% 4. Calculate the matrix exponential:
exp_xi_th = expm(xi_ * th)
% 5. Get initial condition (by inspection from the picture):
g0 = [eye(3) , q; zeros(1,3) , 1]
% 6. Resulting rigid body transformation denoted as "g_ab" is given by:
g_ab = simplify(exp_xi_th * g0)
```

# Example: Rotation about any axis
## Solution in Matlab – Useful functions (2/2)

```
function S = skew(w)
%SKEW  generates a skew-symmetric matrix for a given a vector w
%
%    S = SKEW(w)
%
  S(1,2) = -w(3);
  S(1,3) =  w(2);
  S(2,3) = -w(1);

  S(2,1) =  w(3);
  S(3,1) = -w(2);
  S(3,2) =  w(1);

  S(1,1) = 0;
  S(2,2) = 0;
  S(3,3) = 0;

end
```

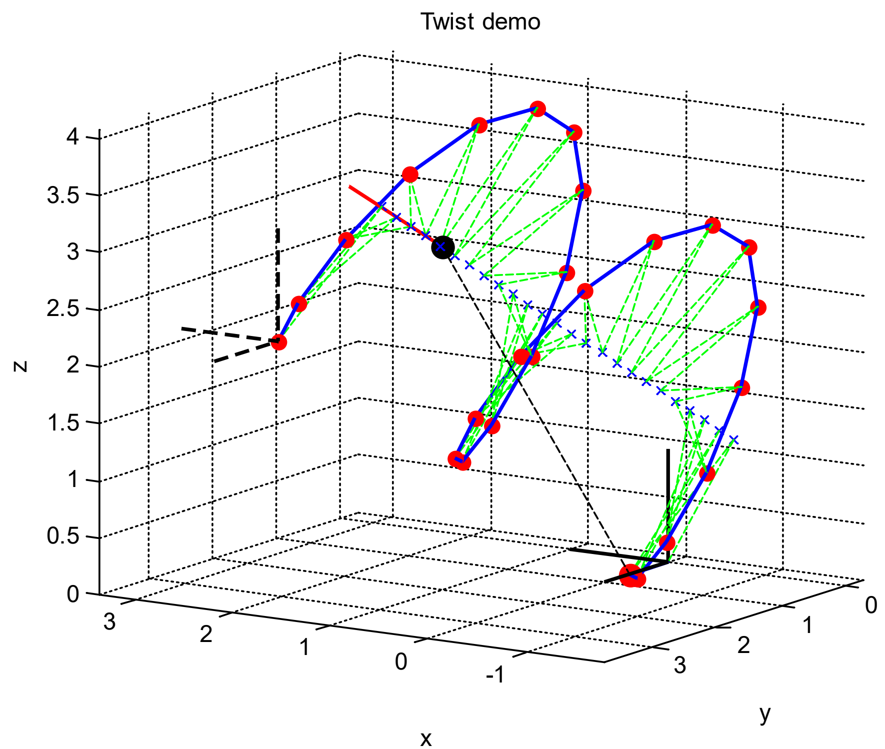
```
function tw = twist(xi)
%TWIST  converts xi from a 6-vector to a 4 x 4 matrix
%
%    tw = TWIST(xi)
%
% The form of xi is [v1; v2; v3; w1; w2; w3]
%

  tw(1:3, 1:3) = skew(xi(4:6));
  tw(1:3, 4  ) = xi(1:3);
  tw(4  , 1:4) = 0;

end
```

# Matlab demo (demo_twist.m)

- **Case 1 – Case 7**

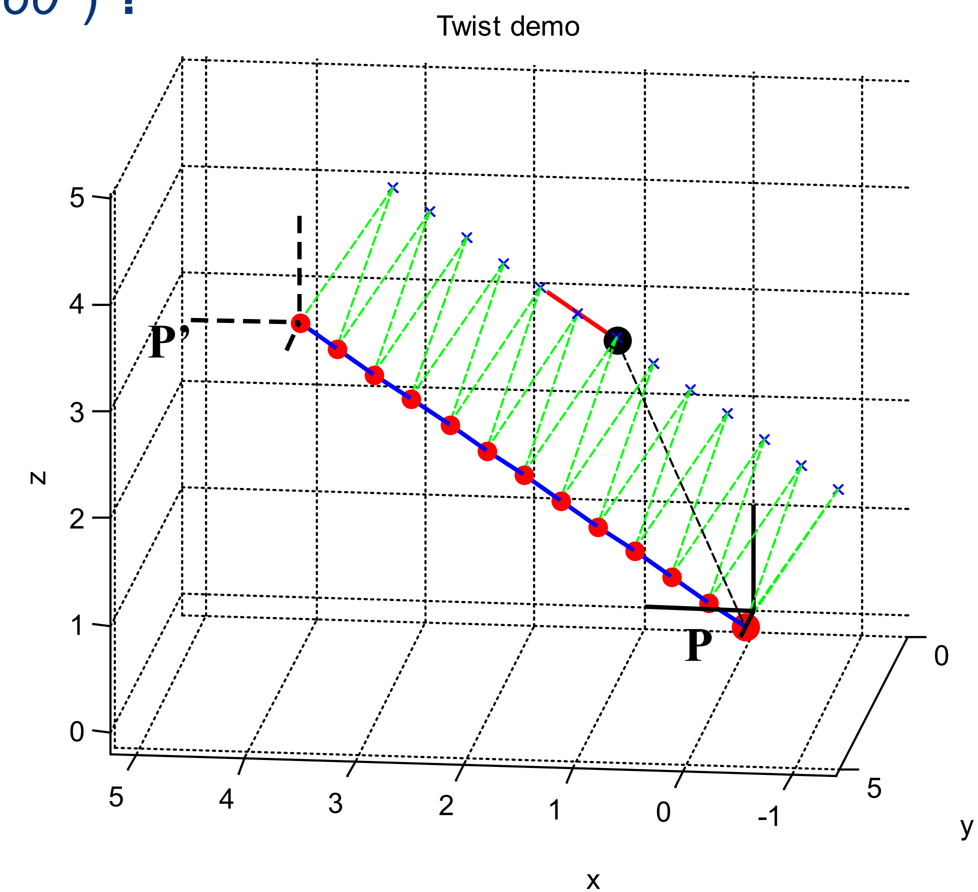
  twist of the same point around different axis passing different points by angle theta

- Case 1,2
  Rotations
- Case 3
  Translation
- Case 4,5
  Twist (both, trans & rot) by different angles theta
- Case 6,7,8
  Twists around the same axis passing different points



# Task/Question

- **Q3: How long will our blue line be (i.e. what is the distance *P'—P*) in case of a translation** ($h=\infty$) **for a "full turn"** ($\theta=360°$) **?**



# End