Geometric Algebra Computing

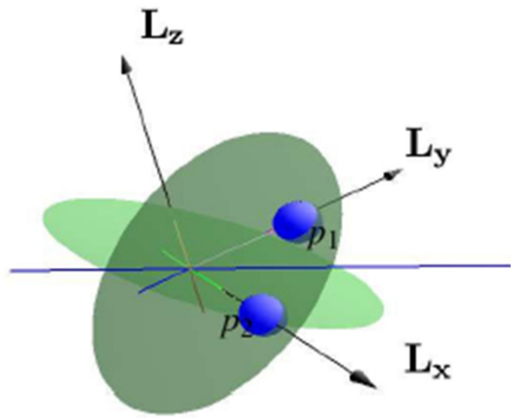
Inverse Kinematics of a Virtual Character

Part I

22.01.2015

Dr. Dietmar Hildenbrand

Technische Universität Darmstadt



Literature

- Chapter 3 and 7 of my dissertation
- Book chapters 4 and 9

Inverse kinematics application

TECHNISCHE UNIVERSITÄT DARMSTADT

- Optimizations on the algorithmic level
  - Embedding of quaternions
  - No need of trigonometric functions
  - [Hildenbrand et al. ICCA 2005]

Quaternions

- Identification in geometric algebra:

$$i = e_3 \wedge e_2,$$
$$j = e_1 \wedge e_3,$$
$$k = e_2 \wedge e_1.$$

Table 3.2: Quaternions in Conformal Geometric Algebra

grade	term	blades	nr.
0	scalar	$\overbrace{\underbrace{\mathbf{1}}}$	1
1	vector	$e_1, e_2, e_3, e_0, e_\infty$	5
2	bivector	$\overbrace{\underbrace{e_1 \wedge e_2}_{-\mathbf{k}},\ \underbrace{e_1 \wedge e_3}_{\mathbf{j}},\ \underbrace{e_2 \wedge e_3}_{-\mathbf{i}}},$ $e_1 \wedge e_\infty,\ e_2 \wedge e_\infty,\ e_3 \wedge e_\infty,$ $e_1 \wedge e_0,\ e_2 \wedge e_0,\ e_3 \wedge e_0,$ $e_0 \wedge e_\infty$	10

Quaternions in geometric algebra

The conformal representation of the Euclidean point
(v_1, v_2, v_3)

$$P = \mathbf{v} + \frac{1}{2}\mathbf{v}^2 e_\infty + e_o, \tag{3.9}$$

the line through the origin e_o and the point is
described by their outer product with the point at infinity e_∞

$$L^* = e_o \wedge P \wedge e_\infty. \tag{3.10}$$

The dualization calculation leads to the standard representation of the line (see figure 3.1 with a screenshot of the Maple computations)

$$L = v_1(e_3 \wedge e_2) + v_2(e_1 \wedge e_3) + v_3(e_2 \wedge e_1). \tag{3.11}$$

or

$$L = v_1 i + v_2 j + v_3 k \tag{3.12}$$

Quaternion computations in Maple

```
> i := e3 &w e2;
```
$$i := -e23$$

```
> j := e1 &w e3;
```
$$j := e13$$

```
> k := e2 &w e1;
```
$$k := -e12$$

```
> v3D := v1*e1 + v2*e2 + v3*e3;
```
$$v3D := v1\ e1 + v2\ e2 + v3\ e3$$

```
> v := conformal(v3D);
```
$$v := v1\ e1 + v2\ e2 + v3\ e3 + \frac{1}{2}(v1^2 + v2^2 + v3^2)\ e4 + \frac{1}{2}(v1^2 + v2^2 + v3^2)\ e5 - \frac{1}{2}\ e4 + \frac{1}{2}\ e5$$

```
> l_dual := e0 &w v &w einf;
```
$$l\_dual := v1\ e145 + v2\ e245 + v3\ e345$$

```
> l := dual(l_dual);
```
$$l := -v1\ e23 + v2\ e13 - v3\ e12$$

Figure 3.1: Quaternion computations in Maple

The imaginary units

$$i^2 = (e_3 \wedge e_2)^2 = e_3 e_2 \underbrace{e_3 e_2}_{-e_2 e_3} = -e_3 \underbrace{e_2 e_2}_{1} e_3 = -\underbrace{e_3 e_3}_{1} = -1$$

$$j^2 = (e_1 \wedge e_3)^2 = e_1 e_3 \underbrace{e_1 e_3}_{-e_3 e_1} = -e_1 \underbrace{e_3 e_3}_{1} e_1 = -\underbrace{e_1 e_1}_{1} = -1$$

$$k^2 = (-e_1 \wedge e_2)^2 = e_1 e_2 \underbrace{e_1 e_2}_{-e_2 e_1} = -e_1 \underbrace{e_2 e_2}_{1} e_1 = -\underbrace{e_1 e_1}_{1} = -1$$

For the multiplication of i and j we get

$$ij = (e_3 \wedge e_2)(e_1 \wedge e_3) = e_3 e_2 e_1 e_3 = e_2 e_3 e_3 e_1 = e_2 \wedge e_1 = k$$

The imaginary units

Accordingly

$$jk = i$$

$$ki = j$$

and

$$ijk = ii = -1$$

We recognize that the three imaginary units i, j, k are represented as the 3 axes in Conformal Geometric Algebra. As for the imaginary unit of section 3.1 i is representing the x-axis as well as j and k are representing the y-axis and the z-axis.

Quaternion product computations in Maple

```
> q1_pure := v11*i +v12*j + v13*k;
```

$$q1\_pure := -v11\ e23 + v12\ e13 - v13\ e12$$

```
> q2_pure := v21*i +v22*j + v23*k;
```

$$q2\_pure := -v21\ e23 + v22\ e13 - v23\ e12$$

```
> Q_pure := q1_pure &c q2_pure;
```

$$Q\_pure := -(v11\ v21 + v12\ v22 + v13\ v23) + (v13\ v21 - v11\ v23)\ e13 - (v12\ v23 - v13\ v22)\ e23 + (v12\ v21 - v11\ v22)\ e12$$

```
> Test := cross(v11,v12,v13,v21,v22,v23) &c e123;
```

$$Test := -(v13\ v21 - v11\ v23)\ e13 + (v12\ v23 - v13\ v22)\ e23 - (v12\ v21 - v11\ v22)\ e12$$

```
> Test2:= Q_pure+Test;
```

$$Test2 := -(v11\ v21 + v12\ v22 + v13\ v23)$$

- The product of pure quaternions is based on the scalar and cross product

Note : The square of a pure quaternion therefore is

$$Q_1^2 = -(v_{11}v_{11} + v_{12}v_{12} + v_{13}v_{13}) = -1 \tag{3.18}$$

Rotations based on unit quaternions

Rotations based on quaternions are restricted to rotations with a rotation axis going through the origin. They can be defined as

$$Q = e^{\frac{\phi}{2}L} \tag{3.19}$$

with

$$L = v_1 i + v_2 j + v_3 k$$

representing a normalized line through the origin according to the Euclidean direction vector of equation (3.8).
This leads to the well-known definition of general quaternions

$$Q = \cos(\frac{\phi}{2}) + L\sin(\frac{\phi}{2}) \tag{3.20}$$

- ! L has to be normalized!

Rotations based on unit quaternions

With the help of the Taylor series and the property $L^2 = -1$ (see equation (3.18))

$$Q = e^{\frac{\phi}{2}L}$$

$$= 1 + \frac{L\frac{\phi}{2}}{1!} + \frac{(L\frac{\phi}{2})^2}{2!} + \frac{(L\frac{\phi}{2})^3}{3!} + \frac{(L\frac{\phi}{2})^4}{4!} + \frac{(L\frac{\phi}{2})^5}{5!} + \frac{(L\frac{\phi}{2})^6}{6!} \dots$$

$$= 1 - \frac{(\frac{\phi}{2})^2}{2!} + \frac{(\frac{\phi}{2})^4}{4!} - \frac{(\frac{\phi}{2})^6}{6!} \dots$$

$$+ L\frac{\frac{\phi}{2}}{1!} - L\frac{(\frac{\phi}{2})^3}{3!} + L\frac{(\frac{\phi}{2})^5}{5!} \dots$$

$$= \cos(\frac{\phi}{2}) + L\sin(\frac{\phi}{2})$$

Rotations based on unit quaternions

In Conformal Geometric Algebra we rotate an object o with the help of the operation

$$o_{rotated} = Qo\tilde{Q} \tag{3.21}$$

with $\tilde{Q}$ being the reverse of Q,

$$\tilde{Q} = \cos(\frac{\phi}{2}) - L\sin(\frac{\phi}{2}) \tag{3.22}$$

which is also indicated as **conjugate** of a quaternion.

Please note that for $\phi = \pi$ the quaternion Q

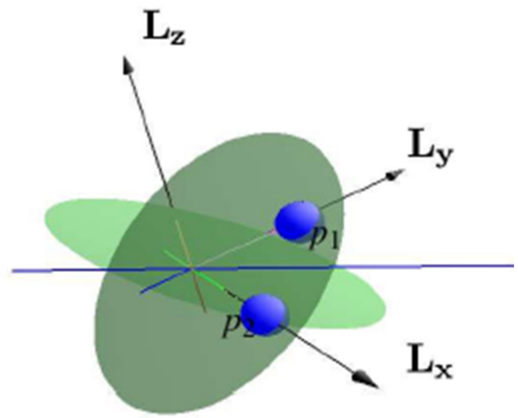
$$Q = v_1 i + v_2 j + v_3 k \tag{3.23}$$

represents a line through the origin and the 3D point represented by the normalized 3D vector (v_1, v_2, v_3) as well as a rotation with angle $\phi = \pi$ about this line.

- Note: we´ll see an inverse kinematics algorithm using this property

Direct computation of quaternions

- Compute a quaternion that rotates an object from P_1 to P_2



At first, we calculate the middle line L_m between the two points through the origin. In Conformal Geometric Algebra, a middle plane of two points is described by their difference (see [64])

$$\pi_m = P_1 - P_2. \tag{7.1}$$

We calculate the middle line with the help of the intersection of this plane and the plane through the origin and the points P_1 and P_2

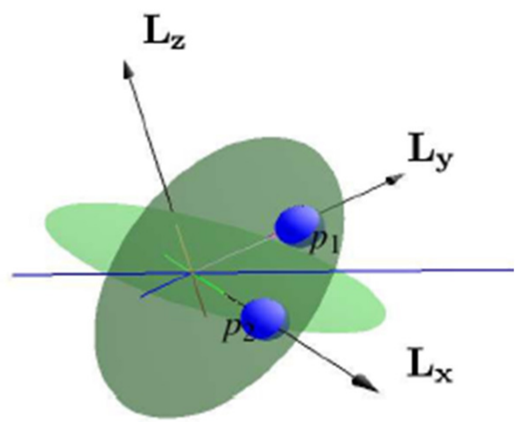
$$\pi_e^* = e_o \wedge P_1 \wedge P_2 \wedge e_\infty, \tag{7.2}$$

and get

$$L_m = \pi_e \wedge \pi_m. \tag{7.3}$$

Direct computation of quaternions

- Compute a quaternion that rotates an object from P_1 to P_2



Second, in order to rotate from P_1 to P_2 we have to rotate around the middle line with radius π. This results in a quaternion identical with the normalized line (see section 3.2.3)

$$Q = \frac{L_m}{|L_m|}. \tag{7.4}$$

In figure 7.1 the two points P_1 and P_2 are indicated by two blue spheres. They can be transformed into each other based on a rotation around the blue middle line.

Efficient computation of quaternions

For efficiency reasons we use an approach to calculate quaternions without the need of using trigonometric functions. According to equation (3.13) a quaternion describing a rotation can be computed with the help of half of an angle and a normalized rotations axis. For example, if $L = i = e_3 \wedge e_2$, the resulting quaternion

$$\begin{aligned} Q &= \cos(\frac{\phi}{2}) + i\sin(\frac{\phi}{2}) \\ &= \cos(\frac{\phi}{2}) + (\mathbf{e_3} \wedge \mathbf{e_2})\sin(\frac{\phi}{2}) \end{aligned}$$

represents a rotation around the x-axis. The angle between two lines or two planes is defined according to section 6.1 as follows:

$$\cos(\theta) = \frac{o_1^* \cdot o_2^*}{|o_1^*|\,|o_2^*|}, \tag{7.5}$$

We already know the cosine of the angle. This is why we are able to compute the quaternion in a more direct way using the following two properties of the trigonometric functions

$$\cos(\frac{\phi}{2}) = \pm\sqrt{\frac{1+\cos(\phi)}{2}} \tag{7.6}$$

and

$$\sin(\frac{\phi}{2}) = \pm\sqrt{\frac{1-\cos(\phi)}{2}}, \tag{7.7}$$

Efficient computation of quaternions

leading to the formulas

$$\cos(\frac{\phi}{2}) = \pm\sqrt{\frac{1 + \frac{o_1^* \cdot o_2^*}{|o_1^*||o_2^*|}}{2}} \tag{7.8}$$

and

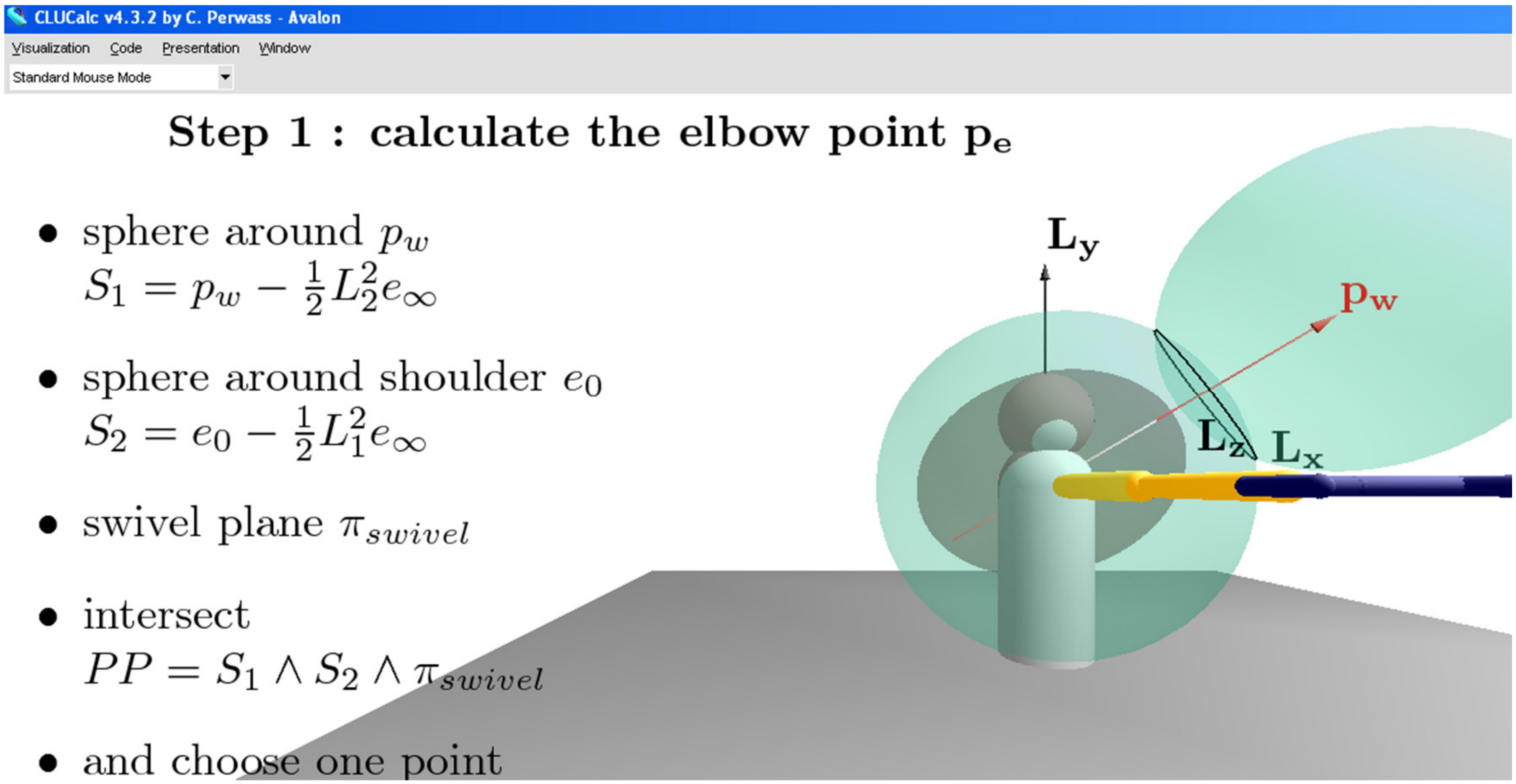
$$\sin(\frac{\phi}{2}) = \pm\sqrt{\frac{1 - \frac{o_1^* \cdot o_2^*}{|o_1^*||o_2^*|}}{2}}. \tag{7.9}$$

The signs of these formulas depend on the application.

- Note: no need of computing trigonometric functions in case of the computation of a quaternion describing the transformation of a joint angle

Inverse kinematics algorithm

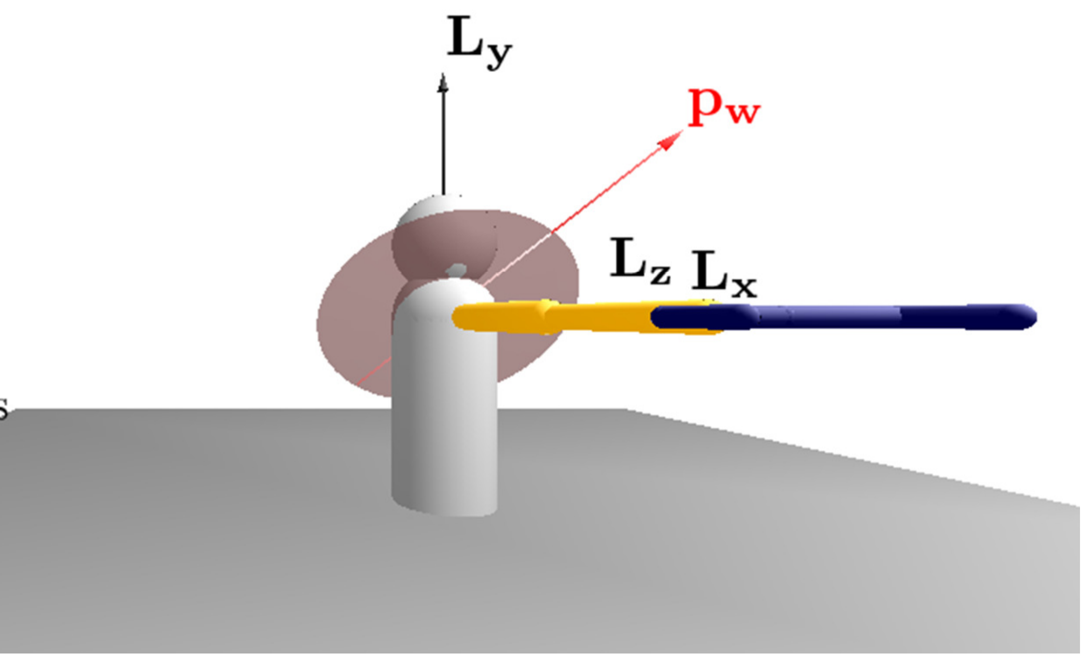
- Chapter 9 of the book „Foundations of Geometric Algebra Computing“ (Avalon.clu)



Inverse kinematics algorithm

7DOF-arm-like kinematic chain

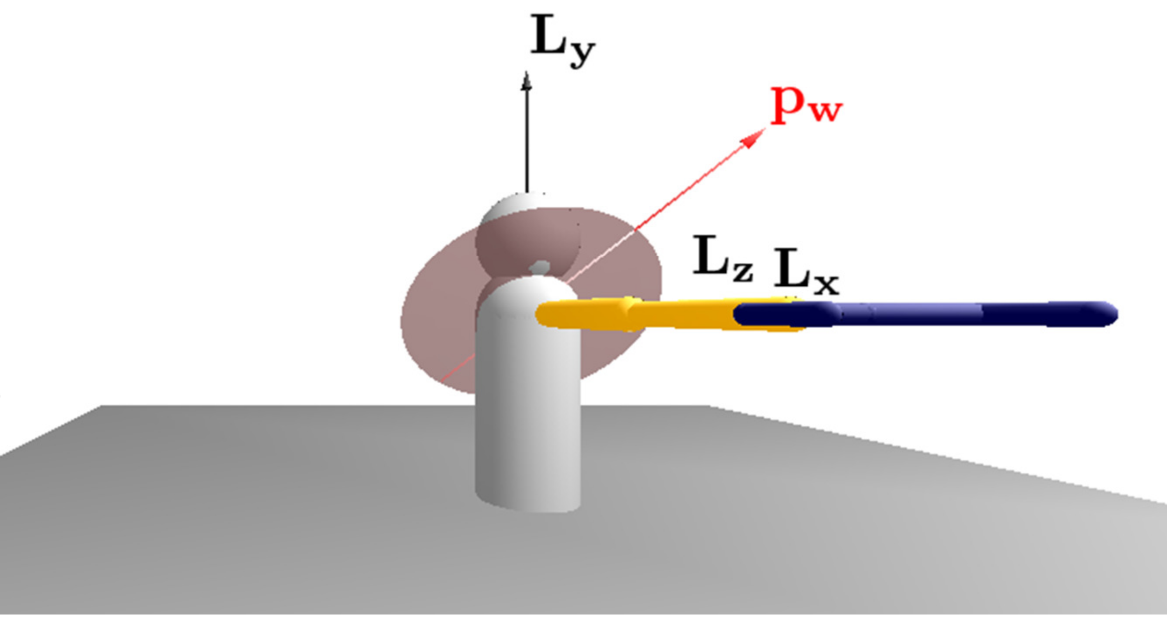
- 7 degrees of freedom
  3 : shoulder
  1 : elbow
  3 : wrist
- swivel plane constraint
  - reduce the number of solutions
  - based on the approach of the IKAN library



Inverse kinematics algorithm

Inverse Kinematics of a human-arm-like robot

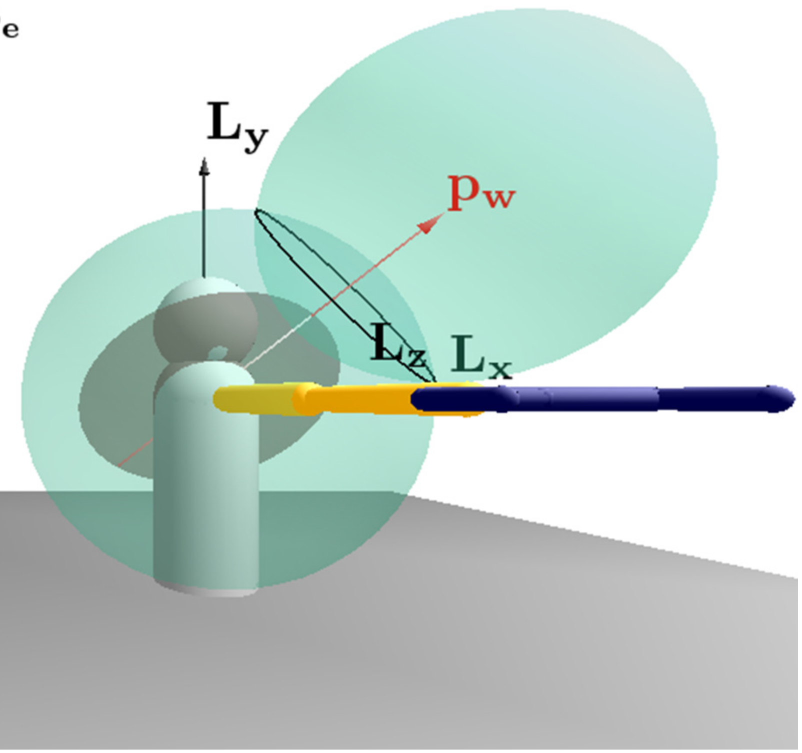
- choose target position p_w
- define the swivel plane through shoulder and p_w
- calculate the joint angles and/or quaternions accordingly
- calculate the quaternions q_s (shoulder) and q_e (elbow)



Inverse kinematics algorithm

Step 1 : calculate the elbow point $\mathbf{p_e}$

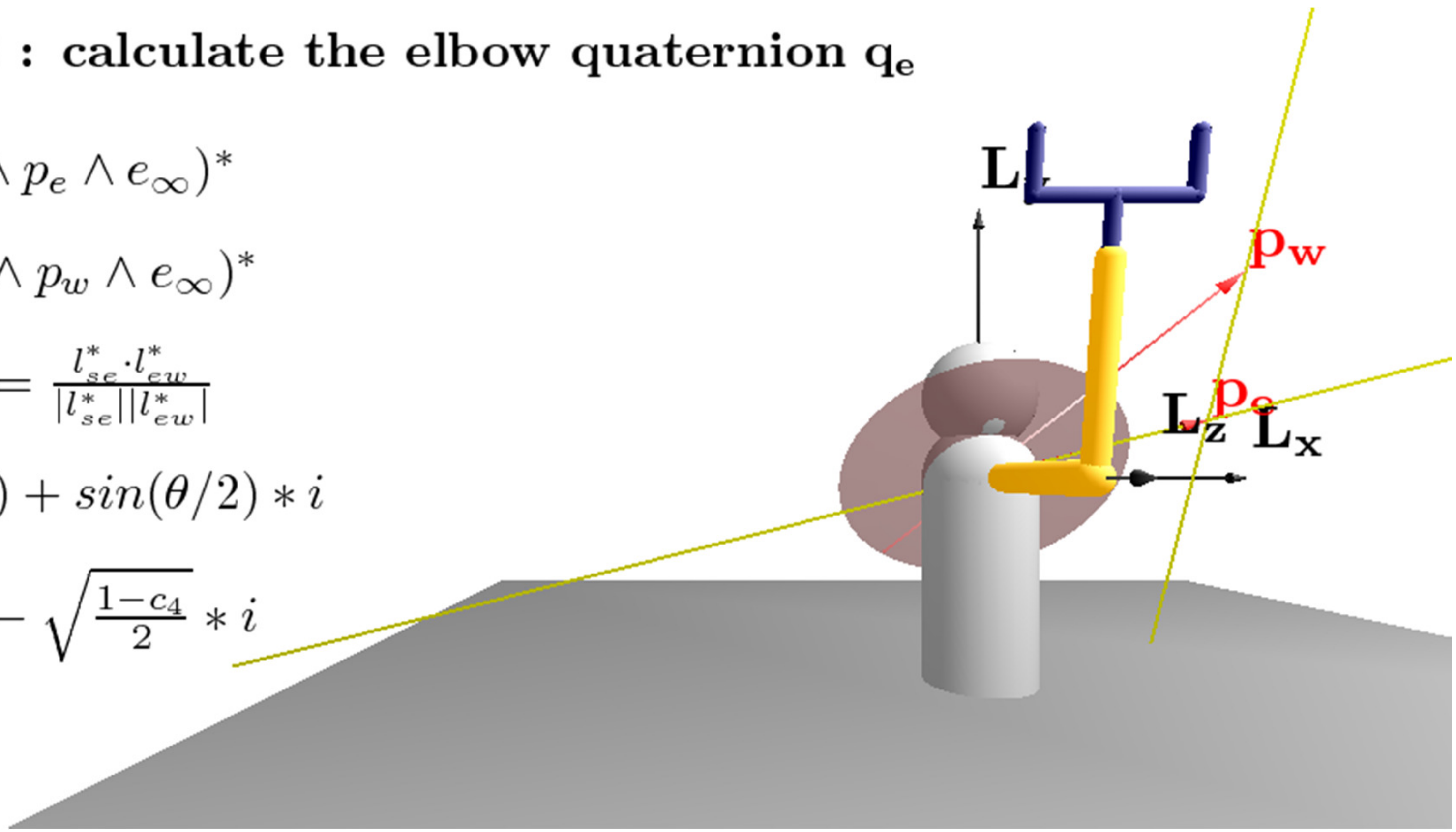
- sphere around p_w
  $S_1 = p_w - \frac{1}{2}L_2^2 e_\infty$
- sphere around shoulder e_0
  $S_2 = e_0 - \frac{1}{2}L_1^2 e_\infty$
- swivel plane π_{swivel}
- intersect
  $PP = S_1 \wedge S_2 \wedge \pi_{swivel}$
- and choose one point
  of the resulting point pair



Inverse kinematics algorithm

Step 2 : calculate the elbow quaternion $\mathbf{q_e}$

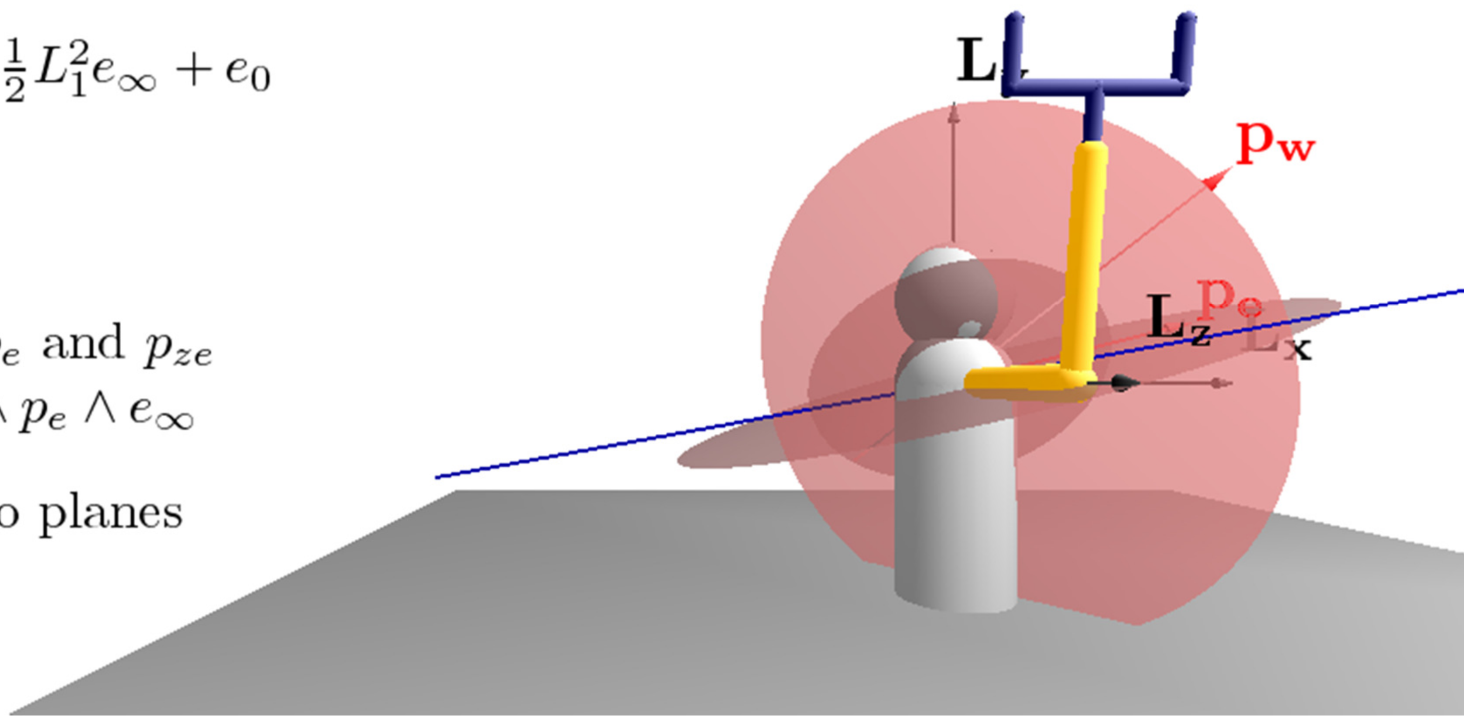
- $\text{line}_{se} = (e_0 \wedge p_e \wedge e_\infty)^*$
- $\text{line}_{ew} = (p_e \wedge p_w \wedge e_\infty)^*$
- $c_4 = \cos(\theta_4) = \frac{l^*_{se} \cdot l^*_{ew}}{|l^*_{se}||l^*_{ew}|}$
- $q_e = cos(\theta/2) + sin(\theta/2) * i$
- $q_e = \sqrt{\frac{1+c_4}{2}} - \sqrt{\frac{1-c_4}{2}} * i$



Inverse kinematics algorithm

Step 3 : compute middle line between p_e and p_{ze}

- $p_{ze} = L_1 * e_3 + \frac{1}{2}L_1^2 e_\infty + e_0$
- middle plane
  $\pi_M = p_{ze} - p_e$
- plane through p_e and p_{ze}
  $\pi_e * = e_0 \wedge p_{ze} \wedge p_e \wedge e_\infty$
- intersect the two planes
  $l_M = \pi_e \wedge \pi_M$

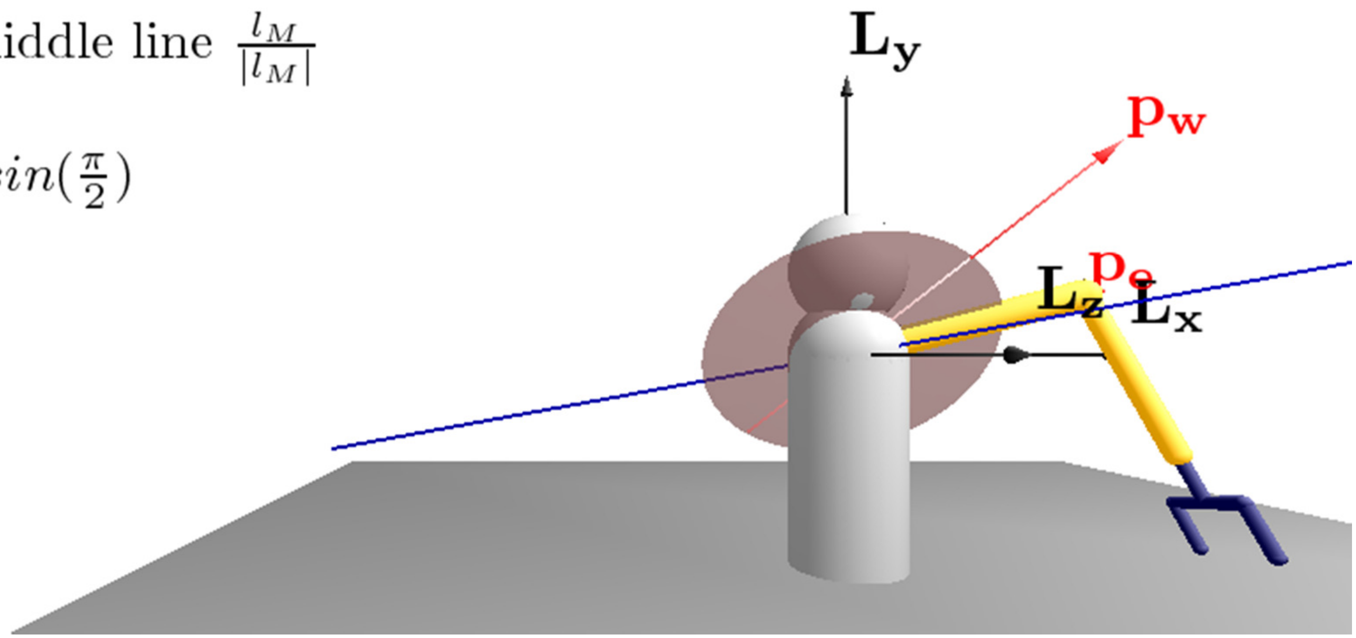


Inverse kinematics algorithm

TECHNISCHE UNIVERSITÄT DARMSTADT

Step 4 : rotate to the elbow position

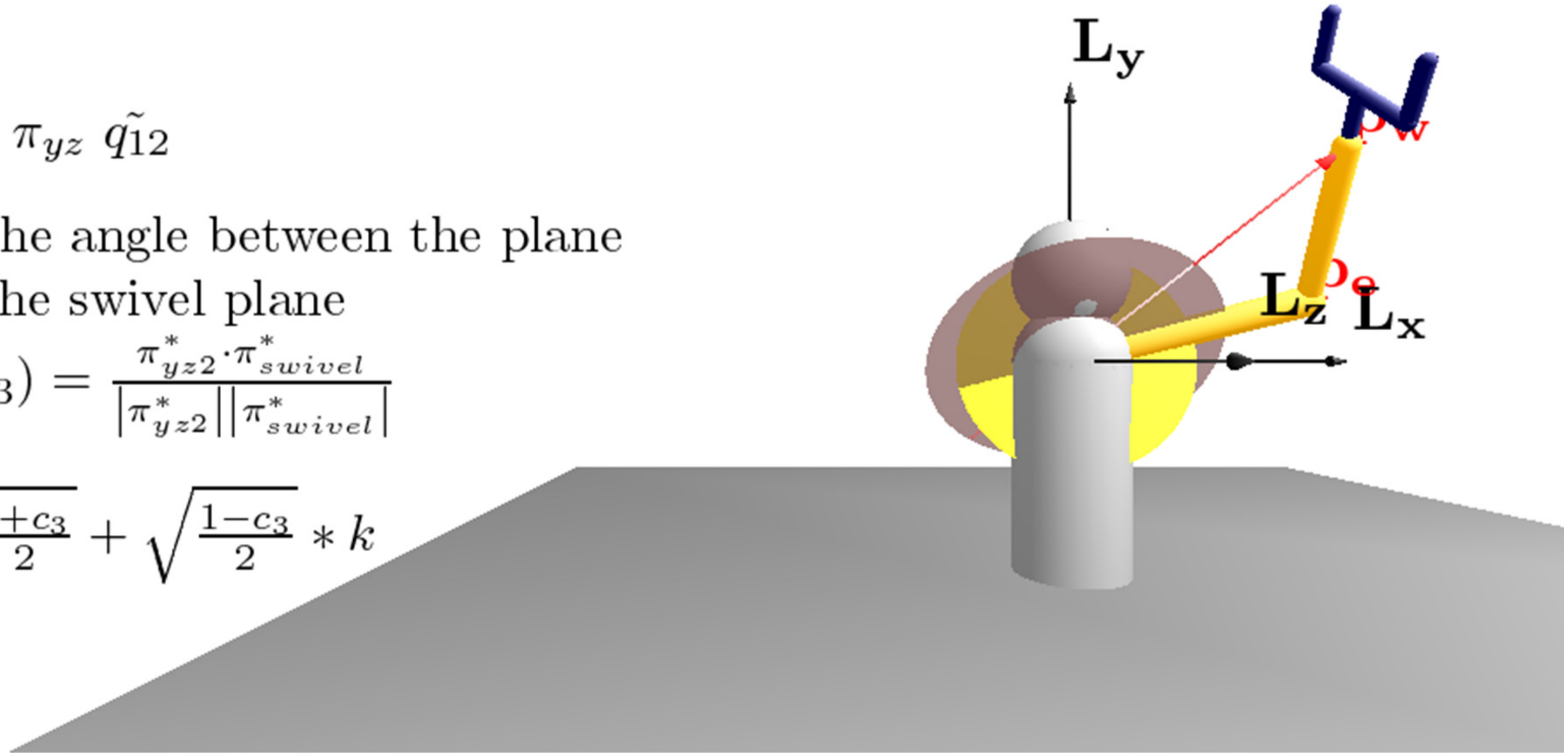
- rotate around the middle line $\frac{l_M}{|l_M|}$ with angle π
  $q_{12} = cos(\frac{\pi}{2}) + \frac{l_M}{|l_M|} sin(\frac{\pi}{2})$
- $q_{12} = \frac{l_M}{|l_M|}$



Inverse kinematics algorithm

Step 5 : rotate to the wrist location

- $\pi_{yz} = e_1$
- $\pi_{yz2} = q_{12}\ \pi_{yz}\ \tilde{q_{12}}$
- calculate the angle between the plane π_{yz2} and the swivel plane
  $c_3 = \cos(\theta_3) = \frac{\pi^*_{yz2} \cdot \pi^*_{swivel}}{|\pi^*_{yz2}||\pi^*_{swivel}|}$
- $q_3 = \pm\sqrt{\frac{1+c_3}{2}} + \sqrt{\frac{1-c_3}{2}} * k$
- $q_s = q_{12} q_3$



Thanks a lot …