Computer Systems in Engineering

Singularities

Tomasz Karolak, Eric Schubert, Chris Taggeselle

Structure

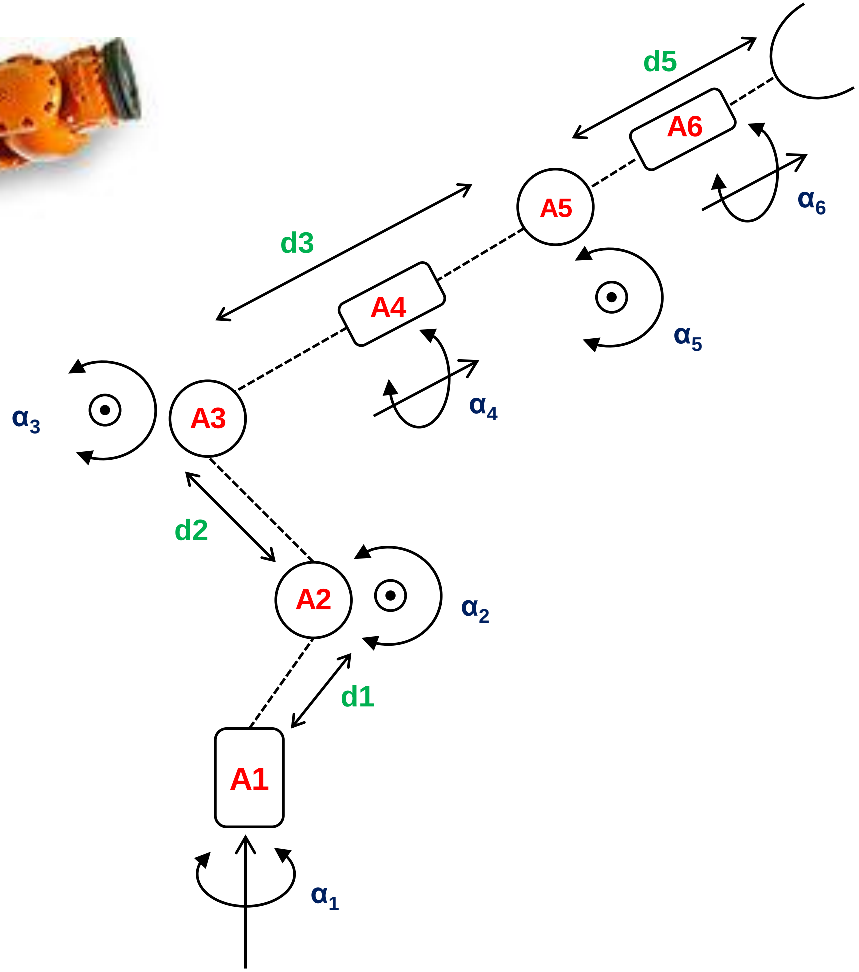
1. Explanation
2. Wrist singularity
3. Overhead singularity

Singularities for 6 DOF

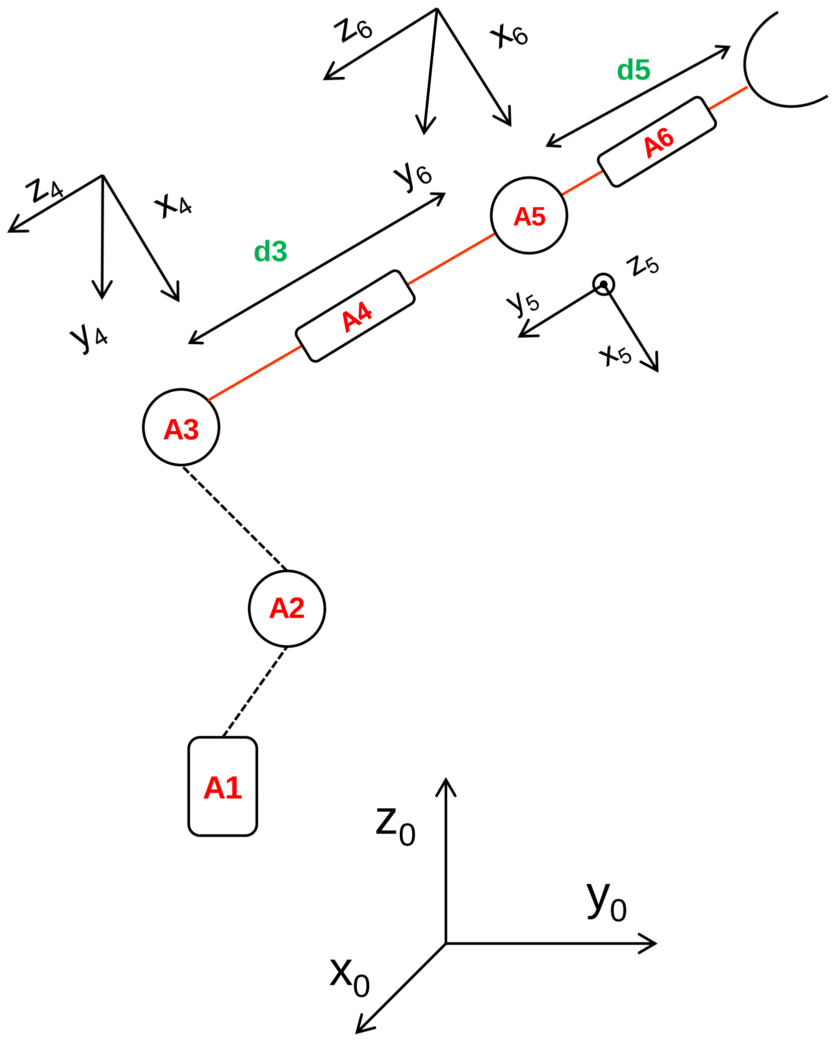
- Can appear on several points
- Arrangements that have infinite solutions
- 2 Types of singularities
  - Wrist
  - Overhead

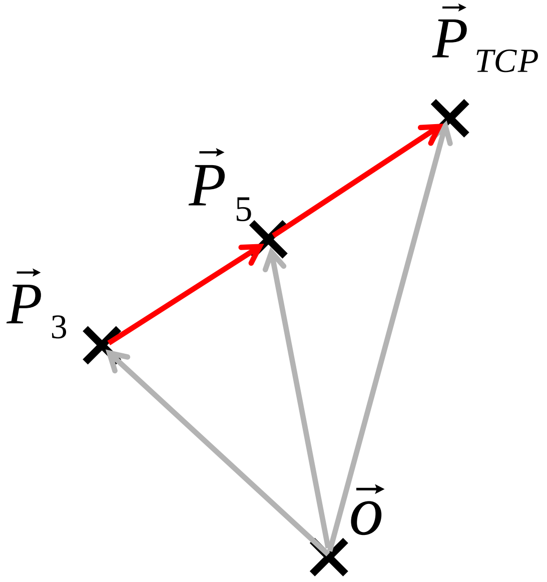
- Explanation with example of KR30J16

6 DOF robot arm and sketch



Wrist singularity





Wrist singularity

Wrist singularity

- Appearance:
  - Calculate $T_0 \bullet T_1 \bullet T_2 \bullet T_3$ and $T_{TCP} \bullet T_6^{-1}$ to get the z-Axis in spherical wrist joints (A4 and A6)
  - If A4-z-Axis equals A6-z-Axis then wrist singularity
- Solution:
  - Calculate next step
  - Take average between angle values of previous step and one step forward
  - Alternative : „jump“ over singularity with PTP

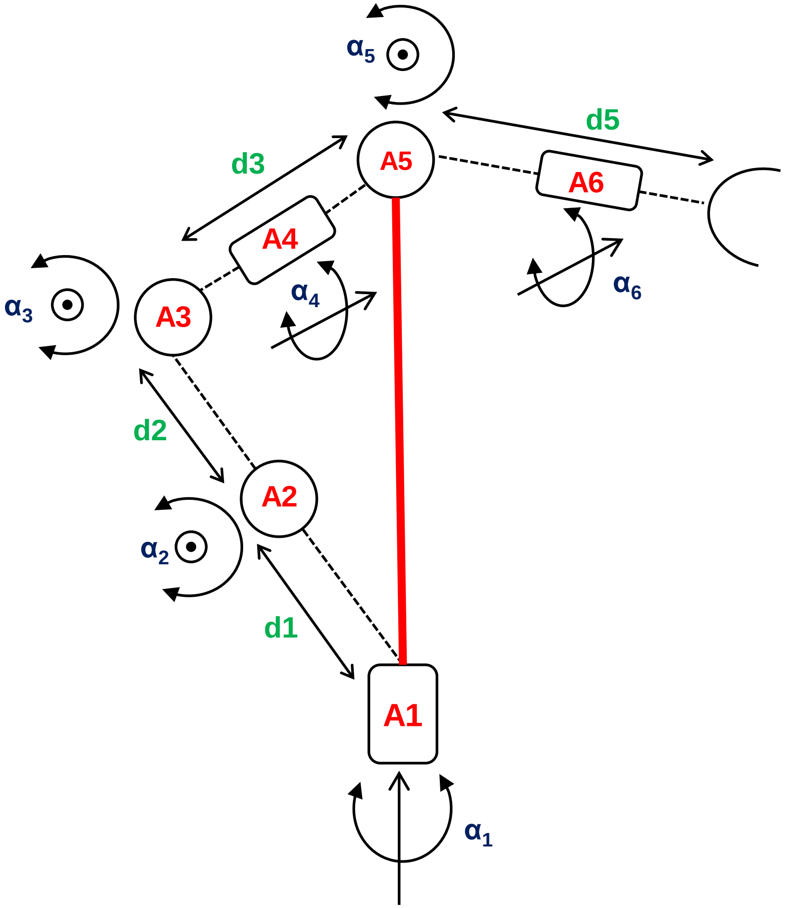
Wrist singularity

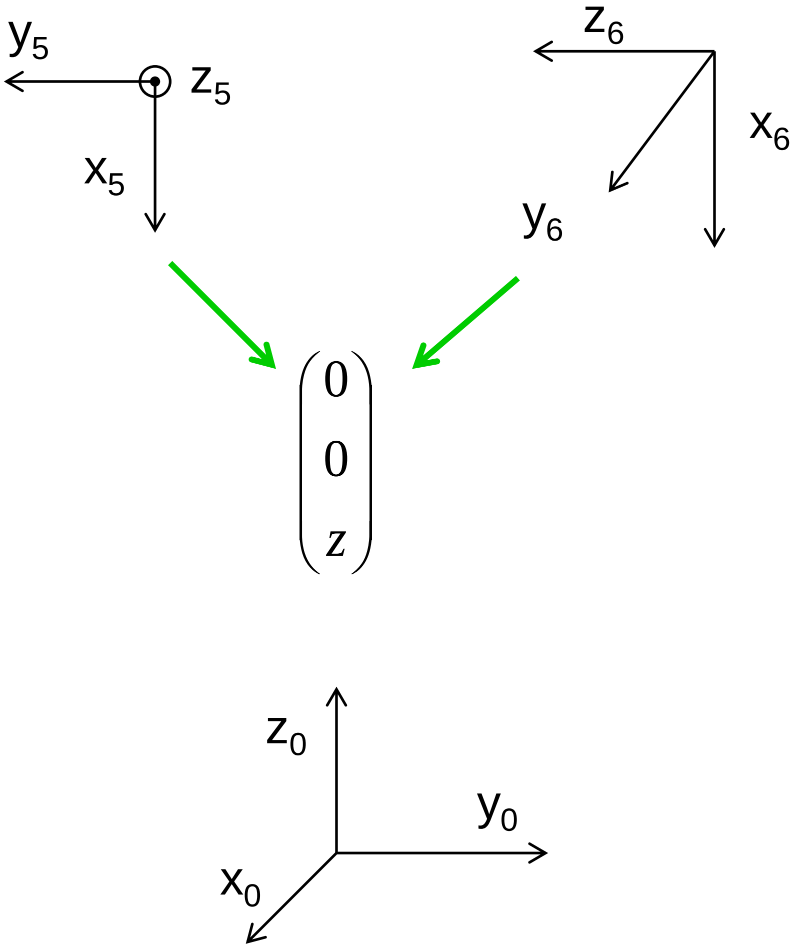
- Pseudocode:

```
  isWristSingularity
{
  m_wrist4 = multiplyMatrix(
      multiplyMatrix(m_0,m_1),
      multiplyMatix(m_2,m_3));
  m_wrist6 =
      multiplyMatrix(m_TCP,getInverse(m_Tool));

  if (m_wrist4[*][2] == m_wrist6[*][2])
      then return true;
  return false;
}
```

Overhead singularity





$$\begin{pmatrix} 0 \\ 0 \\ z \end{pmatrix}$$

Overhead singularity

Overhead singularity

- Appearance:
  - Calculate wrist position in base coordinates without DH
  - $m\_wrist = T_0 \bullet T_1 \bullet T_2 \bullet T_3 \bullet T_4 \bullet T_5 = T_{TCP} \bullet T_6^{-1}$
  - $= \begin{pmatrix} & p_1 \\ R_{1-6} & p_2 \\ & p_3 \\ 0 & 1 \end{pmatrix}$, if $p_1 = p_2 = 0$ then Overhead singularity

- Solution:
  - Calculate next step
  - Take average between angle values of previous step and one step forward
  - Alternative : „jump" over singularity with PTP

Overhead singularity

- Pseudocode:

$$\begin{pmatrix} & p_1 \\ R_{1-6} & p_2 \\ & p_3 \\ 0 & 1 \end{pmatrix} \begin{array}{l} \longrightarrow \texttt{m\_wrist[0][3]} \\ \longrightarrow \texttt{m\_wrist[1][3]} \\ \\ \\ \end{array}$$

```
  isOverheadSingularity
{
  m_wrist = multiplyMatrix(m_TCP,getInverse(m_Tool);

  if (m_wrist[0][3] == 0 and m_wrist[1][3] == 0)
      then return true;
  else return false;
}
```

Thanks for your attention !!!!

Sources

Theory:

- http://coro.etsmtl.ca/blog/?p=107
- http://wwwhomes.uni-bielefeld.de/ggoetze/Robotik/robotikWS0607.pdf
- http://www.ohio.edu/people/williar4/html/PDF/IASTED.pdf

Images:

- http://www.robotize.com.au/images/robot-finder/9e020e.jpg

Videos:

- http://www.youtube.com/watch?v=zlGCurgsqg8

Appendix: Solution

$$\begin{tikzcd}[column sep=small] \alpha(Pos1) \arrow[r] & \alpha(Pos2) \arrow[r] & \alpha(Pos3) \arrow[r] \arrow[d] & \alpha(Pos4)\ldots\alpha(PosN) \\ & & \boxed{\text{Singularity}} & \end{tikzcd}$$

$$\alpha(Pos3) = \frac{\alpha(Pos2)+\alpha(Pos4)}{2}$$

- Pseudocode:

```
  if(isOverheadSingularity() == true )
{
rotationAngle(Pos3) = (rotationAngle(Pos2)+
      rotationAngle(Pos4)) * 0.5;
}
```

Appendix: alternative Solution

- Pseudocode:

```
  if(isOverheadSingularity() == true )
{
MakePTPMovement(Pos2,Pos4);
}
```