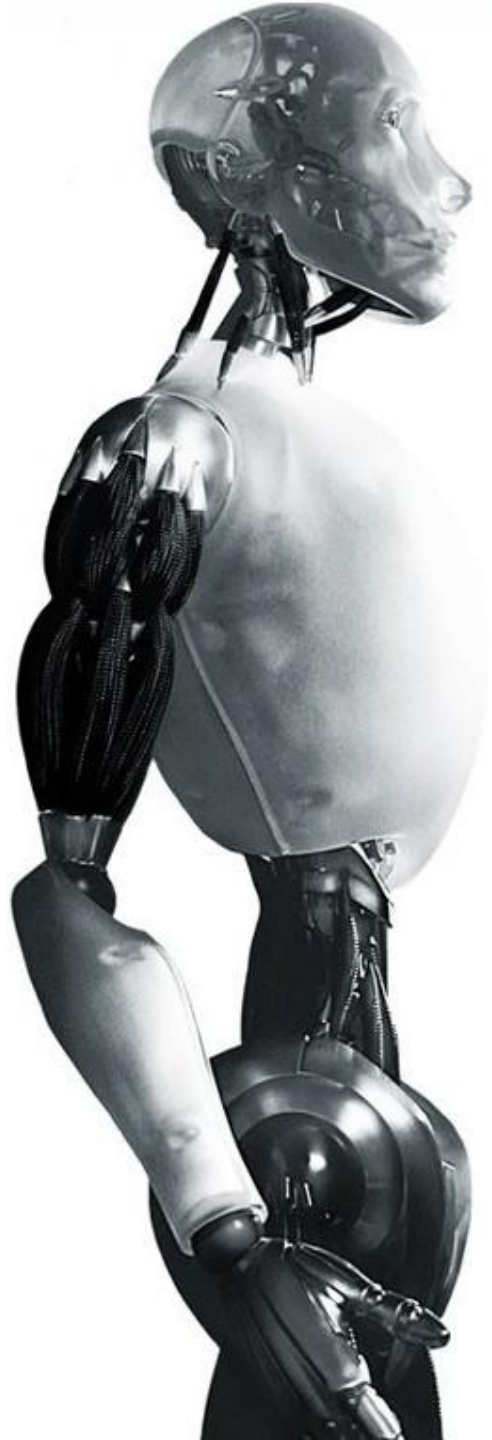
Disclaimer

These slides are intended as presentation aids for the lecture. They contain information that would otherwise be to difficult or time-consuming to reproduce on the board. But they are incomplete, not self-explanatory, and are not always used in the order they appear in this presentation. As a result, these slides should not be used as a script for this course. I recommend you take notes during class, maybe on the slides themselves. It has been shown that taking notes improves learning success.

Reading for this set of slides

- Craig – Intro to Robotics (3rd Edition)
  - Chapter 2, Appendix B

Please note that this set of slides is intended as support for the lecture, not as a stand-alone script. If you want to study for this course, please use these slides in conjunction with the indicated chapters in the text books. The textbooks are available online or in the TUB library (many copies that can be checked out for the entire semester. There are also some aspects of the lectures that will not be covered in the text books but can still be part of the homework or exam. For those It is important that you attend class or ask somebody about what was covered in class.



Robotics

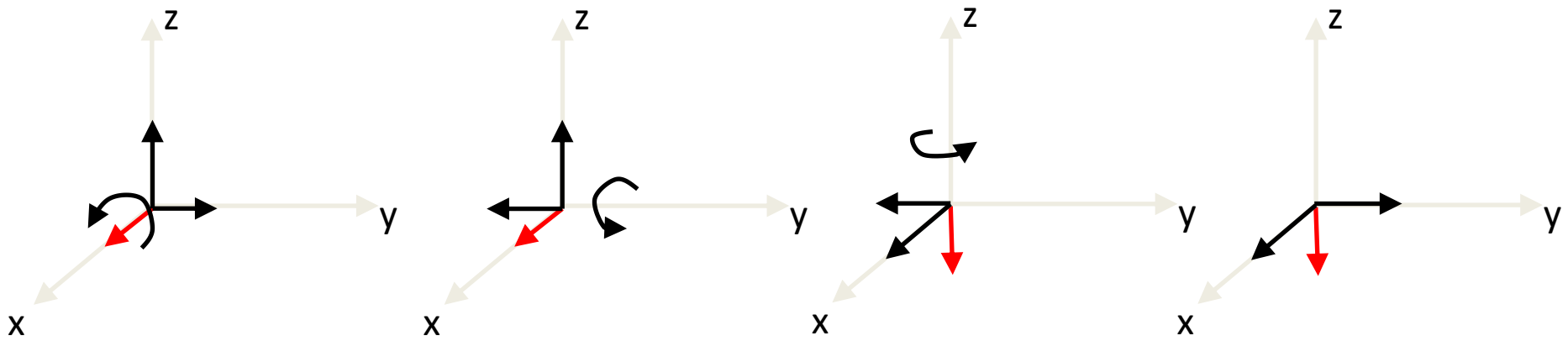
Representations of Orientation

TU Berlin

Oliver Brock

X-Y-Z Fixed Angles (roll, pitch, yaw)

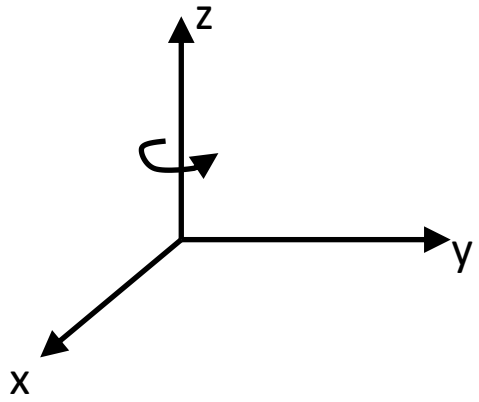
$(\pi/2, \pi/2, \pi/2)$

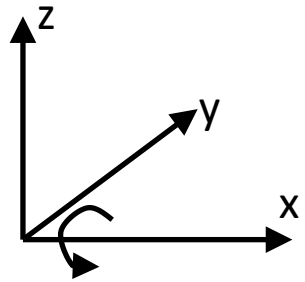


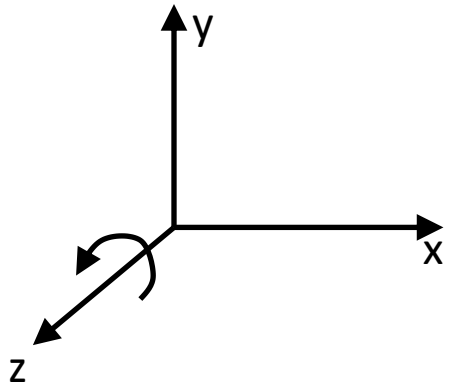
Other combination of axes exist.

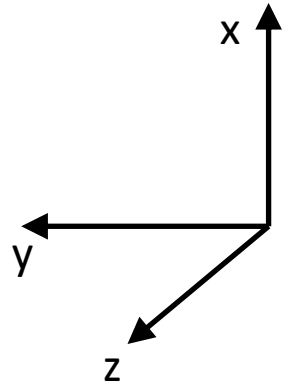
Z-X-Z Euler Angles

$(\pi/2, \pi/2, \pi/2)$





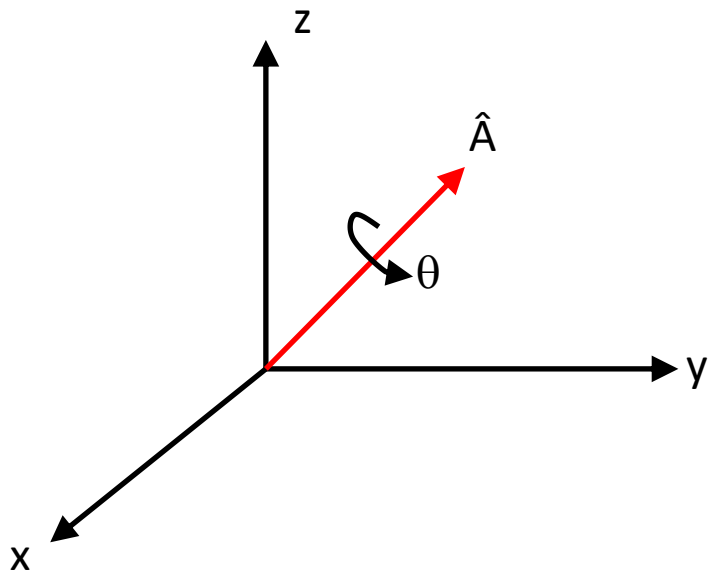




Other combination of axes exist.

Arbitrary Axis

- Direction of vector indicates axis $\hat{A}$
- $|\hat{A}| = 1$, so only two numbers are required
- Magnitude given by angle θ



Conversions

- Exist between all representations (see Appendix B of Craig)
- For example, XYZ fixed angle to rotation matrix:

$$R_{XYZ}(\alpha,\beta,\gamma) = \begin{bmatrix} c\gamma c\beta & c\gamma s\beta s\alpha - s\gamma c\alpha & c\gamma s\beta c\alpha + s\gamma s\alpha \\ s\gamma c\beta & s\gamma s\beta s\alpha + c\gamma c\alpha & s\gamma s\beta c\alpha - c\gamma s\alpha \\ -s\beta & c\beta s\alpha & c\beta c\alpha \end{bmatrix}$$

Problems with Rotation

- Rotation matrices are used most often, but
  - numerical error buildup
  - interpolation
  - complexity of multiplication
- Angle representations
  - address numerical error buildup
  - but have degeneracies (Gimbal problem)
- Arbitrary axis representation introduces
  - zero rotation problem

Quaternions

- Address:
  - numerical error buildup
  - interpolation
  - complexity of multiplication (among themselves)
- But:
  - eliminate homogeneity
  - are inefficient when many points are rotated
    (then we convert them into a matrix before performing the rotation)