Department of Electronic and Computer Engineering
Technical University Of Crete

Forward and Inverse Kinematics for the NAO Humanoid Robot

Nikolaos Kofinas

Thesis Committee
Assistant Professor Michail G. Lagoudakis (ECE)
Professor Minos Garofalakis (ECE)
Assistant Professor Aggelos Bletsas (ECE)

Chania, July 2012

Motivation

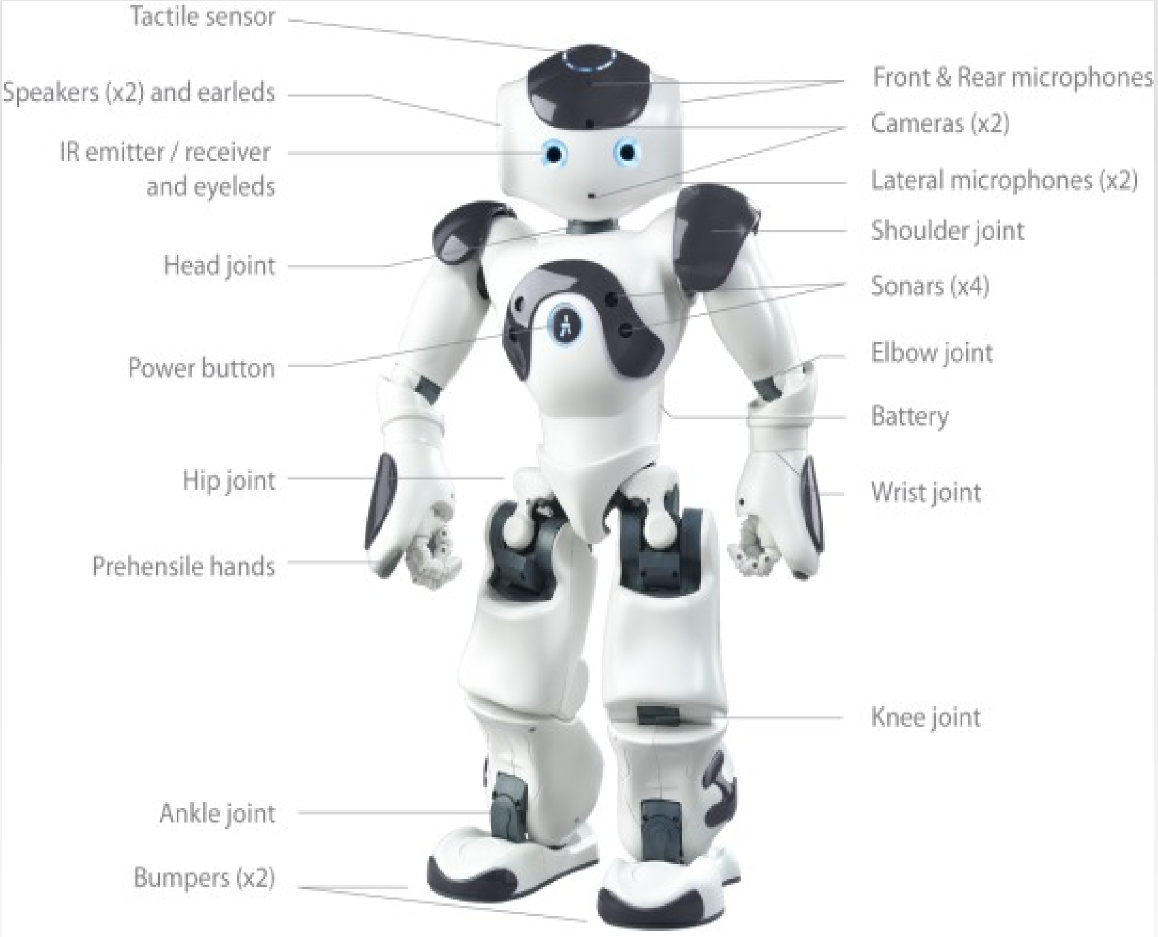
- How do we know where our hands are without seeing them?
- How can we move our hand to a certain point around us?
- Can we make this happen with the robots too?

RoboCup

- **RoboCup**
  - robot soccer competition
  - RoboCup 2012 in Mexico
- **Standard Platform League**
  - identical robots (NAO)
  - focus on software
- **Kouretes**
  - TUC RoboCup SPL team

Aldebaran NAO Humanoid Robot

- **Hardware**
  - 21 independent joints
  - 12-bit encoders
  - cameras, inertial unit
  - FSR, sonars, bumpers
  - 500MHz AMD Geode
- **Software**
  - NaoQi
  - C++, Python
  - Monas



Kinematics and RoboCup

- **Walking**
  - need to follow planned trajectories in the three-dimensional space
- **Kicking**
  - need to follow dynamic trajectories in the three-dimensional space
- **Balancing**
  - need to know the exact position of the center of mass
- **Vision**
  - need to know the exact position and orientation of the camera
- ...

Outline

- Robot Kinematics
- NAO Kinematic Chains
- NAO Forward and Inverse Kinematics
- Results
- Live Demonstration
- Conclusion

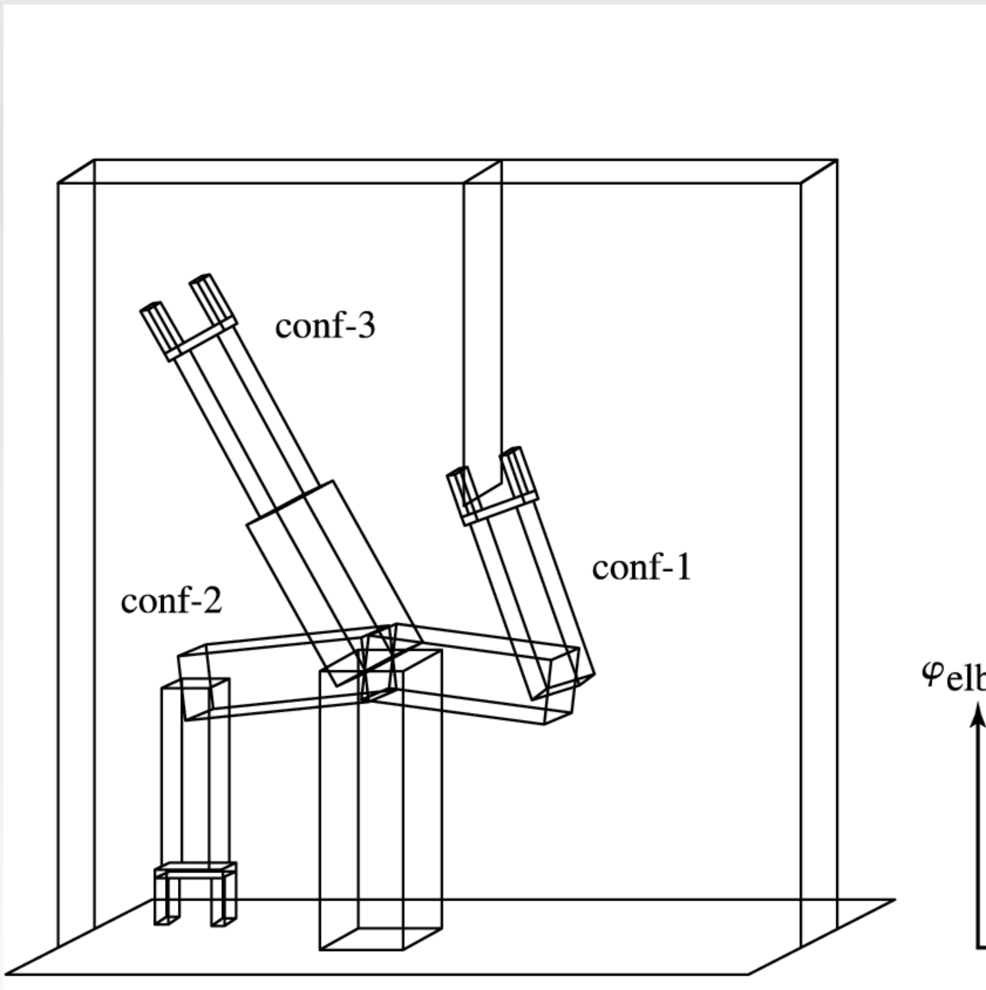
Outline

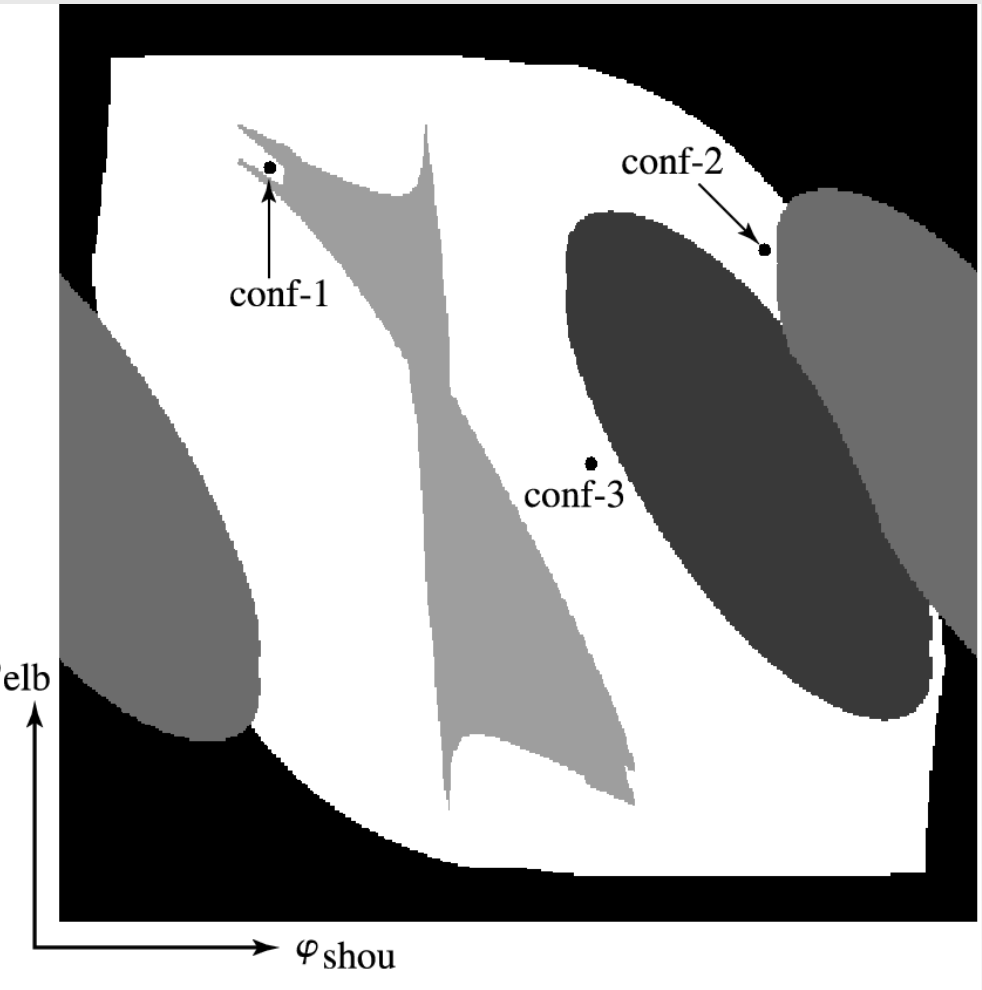
- Robot Kinematics
- NAO Kinematic Chains
- NAO Forward and Inverse Kinematics
- Results
- Live Demonstration
- Conclusion

3D Space and Joint Space

- **3D space**
  - the physical space around the robot
  - a point is described by its position and its orientation
  - position: 3 Cartesian coordinates, orientation: 3 Euler angles
- **Joint space**
  - the space formed by the joints of the robot
  - the dimensions are the degrees of freedom (DOF)
  - a point is described by an n-dimensional vector

3D Space and Joint Space





Robot Kinematics

- **Robot kinematic chain**
  - an articulated manipulator that interacts with the environment
  - a series of links and joints (multiple degrees of freedom)
- **Robot kinematics**
  - the application of geometry to the study of kinematic chains
- **Forward kinematics**
  - a mapping from the joint space to the 3D space
- **Inverse kinematics**
  - a relation between the 3D space and the joint space

Affine Transformations

- **Affine transformation**
  - transformation between spaces that preserves ratios of distances
- **Affine transformation matrix**
  - always invertible, if X invertible
  - product of affine transformations is a new affine transformation

$$T = \begin{bmatrix} X & Y \\ \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} & 1 \end{bmatrix}$$

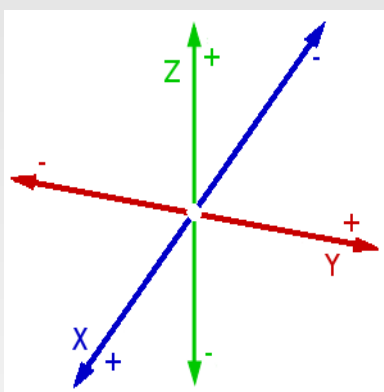
- **Affine rotation and translation matrices**

$$R = \begin{bmatrix} \widehat{R} & \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} & 1 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T = \begin{bmatrix} \widehat{R} & \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & 1 \end{bmatrix}$$

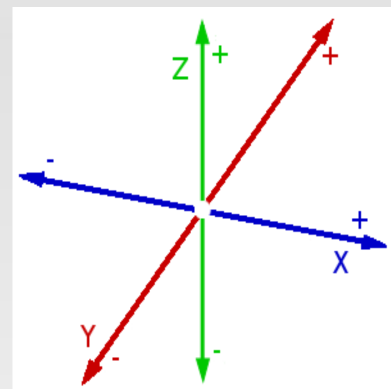
  - any number of translations and rotations can be combined multiplicatively to form a single (affine) transformation

Ordering of Rotations

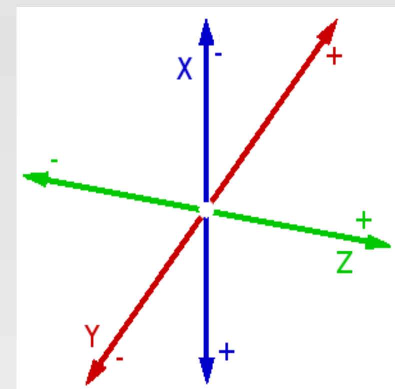
Initial axes



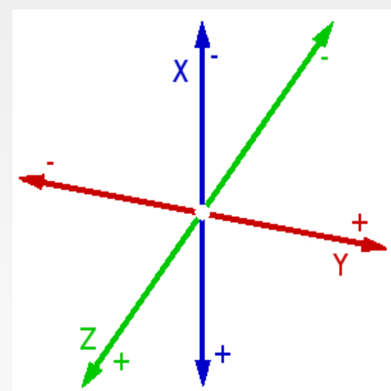
Rotate π/2 about z-axis



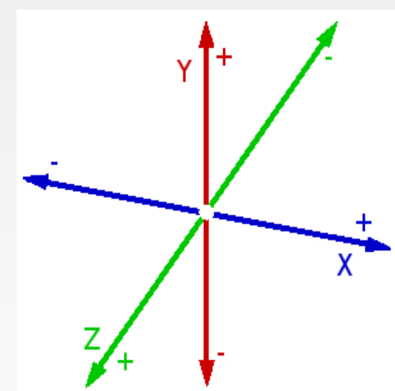
Rotate π/2 about y-axis



Rotate π/2 about y-axis



Rotate π/2 about z-axis

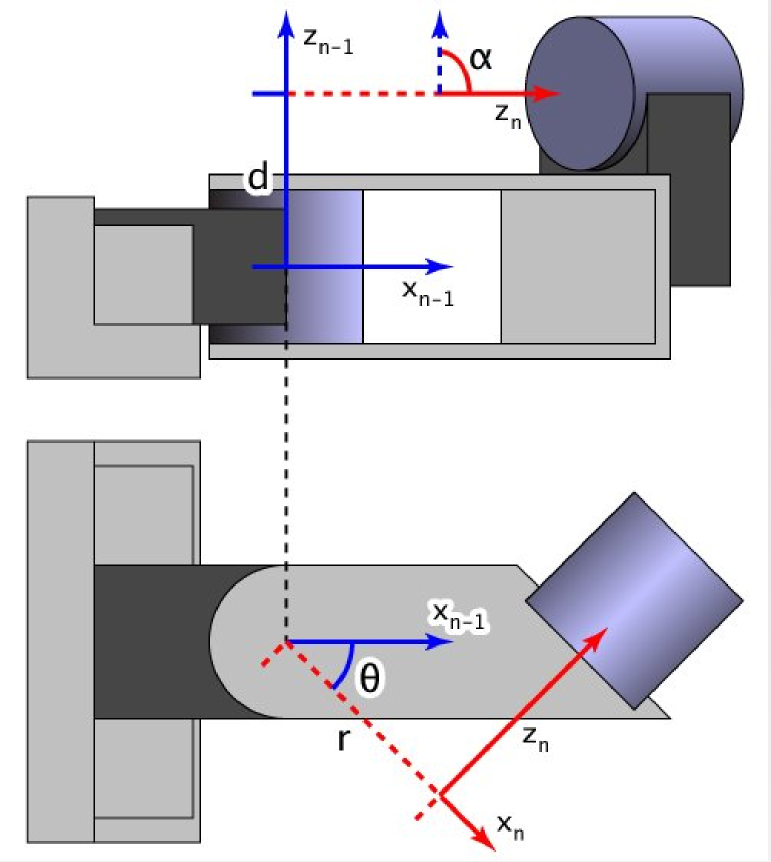


Forward Kinematics Approaches

- **Denavit and Hartenberg (DH)**
  - DH parameters: a,α,d, θ
  - can be defined for any joint in a kinematic chain
- **DH transformation**
  - two translations and two rotations

$$T_{DH} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d)$$

$$T_{DH} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & a \\ \sin\theta\cos\alpha & \cos\theta\cos\alpha & -\sin\alpha & -d\sin\alpha \\ \sin\theta\cos\alpha & \cos\theta\sin\alpha & \cos\alpha & d\cos\alpha \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- **Forward kinematics equation**
  - a product of rotations, translations, and DH transformations

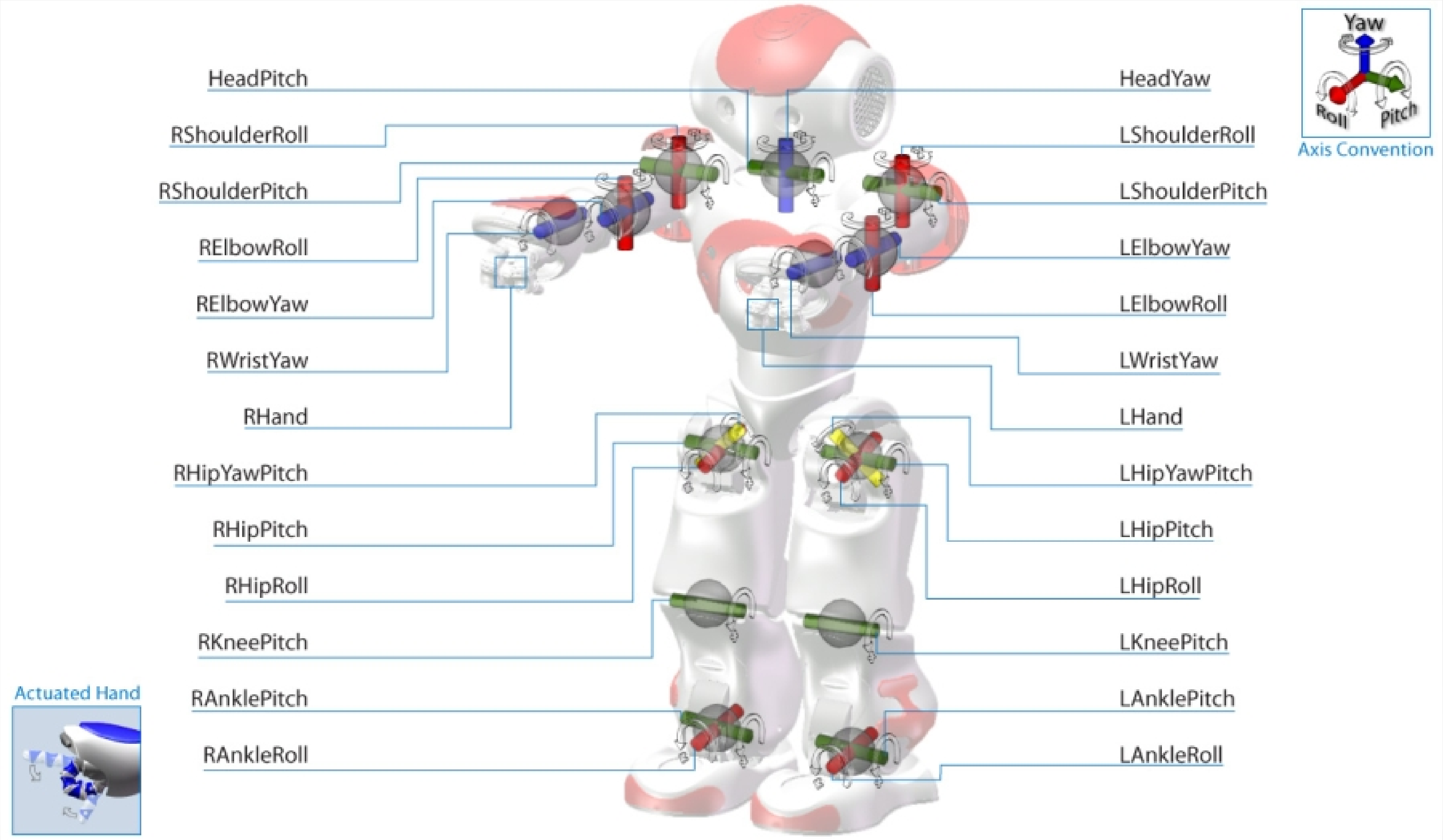
Inverse Kinematics Approaches

- **Numerical approach**
  - iterative approximation methods
  - mostly based on the Jacobian of the transformation matrix
  - may run into singularities
  - accuracy and efficiency depend on the iteration steps
- **Analytical approach**
  - solution of a non-linear system of equations
  - analytical solution can be obtained only for 6 or less DOF
  - for 6 DOF, three consequtive joints must have intersecting axes
  - no singularities
  - accurate

Outline

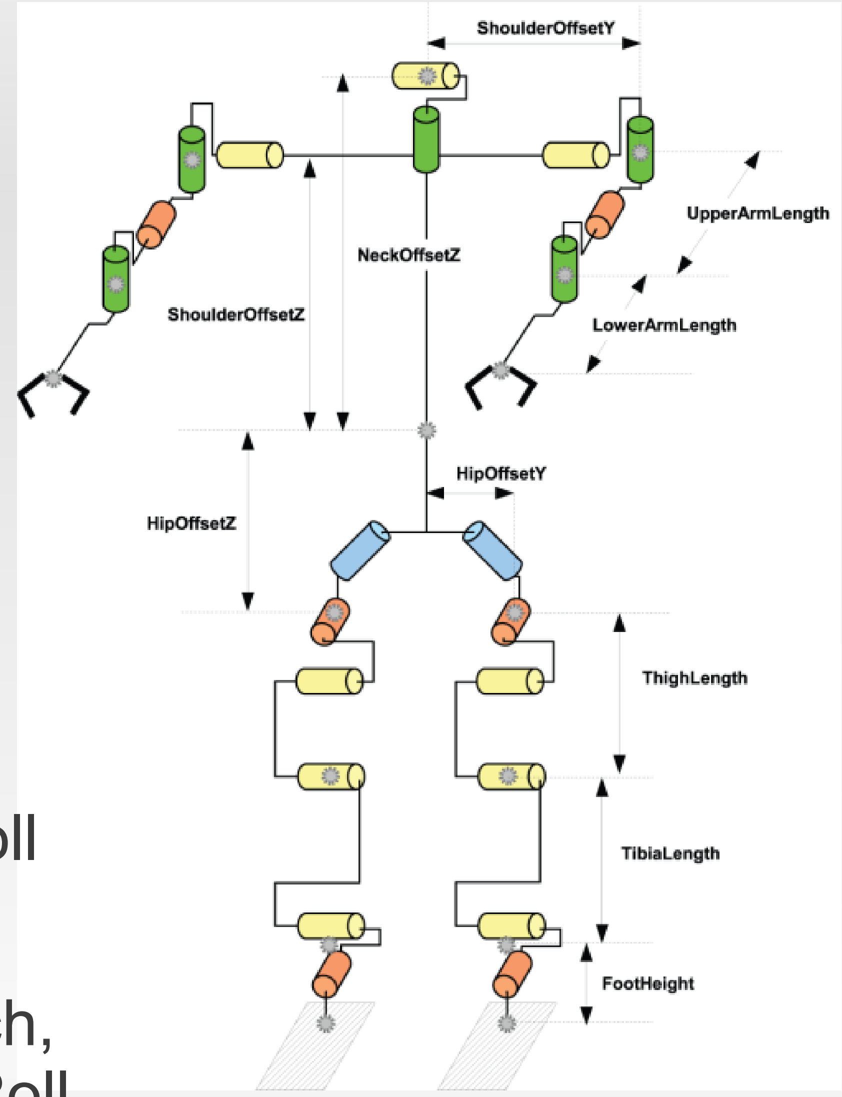
- Robot Kinematics
- NAO Kinematic Chains
- NAO Forward and Inverse Kinematics
- Results
- Live Demonstration
- Conclusion

NAO Degrees of Freedom (DOF)

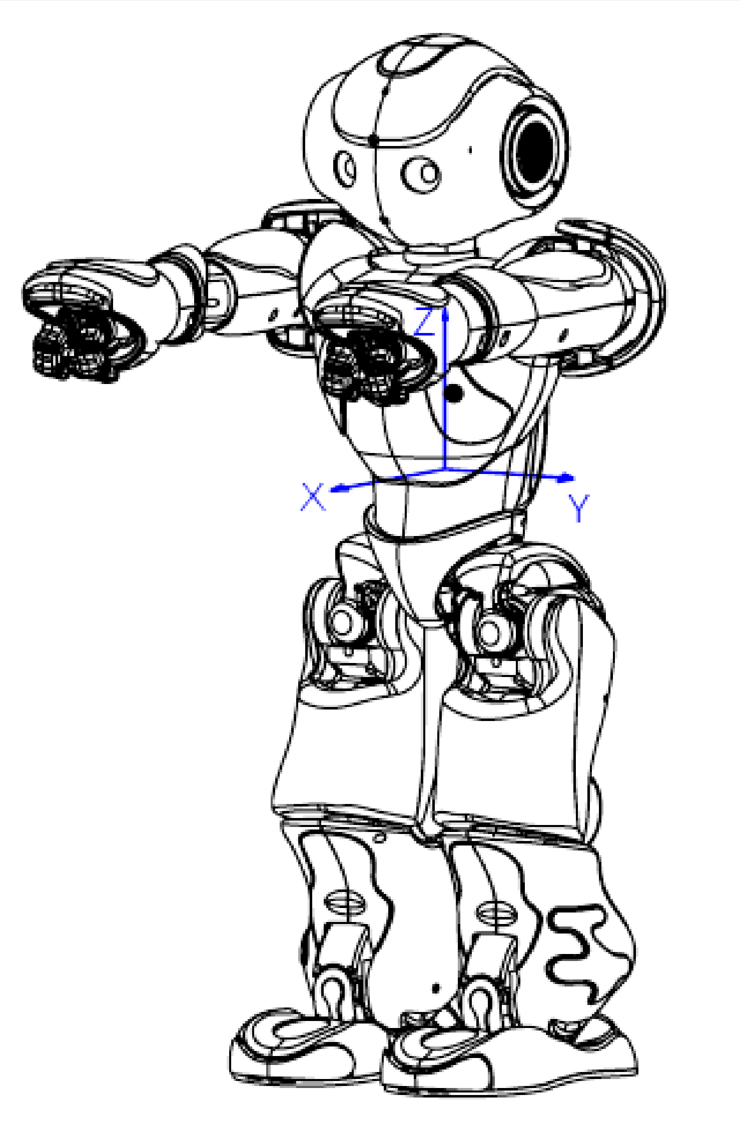


NAO Kinematic Chains

- **Head**
  - HeadYaw, HeadPitch
- **Left Arm**
  - LSHoulderPitch, LShoulderRoll, LElbowYaw, LElbowRoll
- **Right Arm**
  - RSHoulderPitch, RshoulderRoll, RelbowYaw, RElbowRoll
- **Left Leg**
  - LHipYawPitch, LHipRoll, LHipPitch, LKneePitch, LAnklePitch, LAnkleRoll
- **Right Leg**
  - RHipYawPitch, RhipRoll, RLHipPitch, RKneePitch, RAnklePitch, RAnkleRoll



Base (Torso) Frame of NAO



Outline

- Robot Kinematics
- NAO Kinematic Chains
- NAO Forward and Inverse Kinematics
- Results
- Live Demonstration
- Conclusion

NAO Head Forward Kinematics

Frame (Joint)	a	α	d	θ
Base	$A(0, 0, \text{NeckOffsetZ})$			
HeadYaw	0	0	0	θ_1
HeadPitch	0	$-\frac{\pi}{2}$	0	$\theta_2 - \frac{\pi}{2}$
Rotation	$R_x(\frac{\pi}{2})R_y(\frac{\pi}{2})$			
Top Camera	$A(\text{topCameraX}, 0, \text{topCameraZ})$			
Bottom Camera	$A(\text{bottomCameraX}, 0, \text{bottomCameraZ})$			
topCameraX=53.9mm, topCameraZ=67.9mm, bottomCameraX=48.8mm, bottomCameraZ=23.8mm				

$$T_{\text{Base}}^{\text{End}} = A_{\text{Base}}^{0} T_0^1 T_1^2 R_x\left(\frac{\pi}{2}\right) R_y\left(\frac{\pi}{2}\right) A_2^{\text{End}}$$

NAO Left Arm Forward Kinematics

Frame (Joint)	**a**	$\boldsymbol{\alpha}$	**d**	$\boldsymbol{\theta}$
Base	$A(0, \text{ShoulderOffsetY+ElbowOffsetY}, \text{ShoulderOffsetZ})$			
LShoulderPitch	0	$-\frac{\pi}{2}$	0	θ_1
LShoulderRoll	0	$\frac{\pi}{2}$	0	$\theta_2 - \frac{\pi}{2}$
LElbowYaw	0	$-\frac{\pi}{2}$	UpperArmLength	θ_3
LElbowRoll	0	$\frac{\pi}{2}$	0	θ_4
Rotation	$R_z(\frac{\pi}{2})$			
End effector	$A(\text{HandOffsetX+LowerArmLength}, 0, 0)$			

$$T_{\text{Base}}^{\text{End}} = A_{\text{Base}}^{0} T_0^1 T_1^2 T_2^3 T_3^4 R_z(\tfrac{\pi}{2}) A_4^{\text{End}}$$

NAO Left Leg Forward Kinematics

Frame (Joint)	**a**	$\boldsymbol{\alpha}$	**d**	$\boldsymbol{\theta}$
Base	$A(0, \text{HipOffsetY}, -\text{HipOffsetZ})$			
LHipYawPitch	0	$-\frac{3\pi}{4}$	0	$\theta_1 - \frac{\pi}{2}$
LHipRoll	0	$-\frac{\pi}{2}$	0	$\theta_2 + \frac{\pi}{4}$
LHipPitch	0	$\frac{\pi}{2}$	0	θ_3
LKneePitch	$-\text{ThighLength}$	0	0	θ_4
LAnklePitch	$-\text{TibiaLength}$	0	0	θ_5
LAnkleRoll	0	$-\frac{\pi}{2}$	0	θ_6
Rotation	$R_z(\pi)R_y(-\frac{\pi}{2})$			
End effector	$A(0, 0, -\text{FootHeight})$			

$$T_{\text{Base}}^{\text{End}} = A_{\text{Base}}^{0} T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 T_5^6 R_z(\pi) R_y(-\frac{\pi}{2}) A_6^{\text{End}}$$

NAO Inverse Kinematics

- **Analytical approach**
  - accuracy, efficiency, no singularities
- **Solution Methodology**
  1) Construct the (numeric) transformation matrix to the target point
  2) Construct the (symbolic) transformation matrix through the chain
  3) Form a non-linear system by equating the two matrices
  4) Manipulate both sides to make the problem easier
  5) Find values for some joints through geometry and trigonometry
  6) Find values for the remaining joints from the non-linear system
  7) Validate all candidate solutions through forward kinematics
- **Target point**

$$\text{position: } (p_x, p_y, p_z) \qquad \text{orientation: } (a_x, a_y, a_z)$$

NAO Head Inverse Kinematics

$$\theta_1 = a_z$$
$$\theta_2 = a_y$$

$$\theta_2 = \arcsin\left(\frac{-p_z + l_3}{\sqrt{{l_1}^2 + {l_2}^2}}\right) - \arctan\left(\frac{l_1}{l_2}\right) + \frac{\pi}{2}$$

$$\theta_2 = \pi - \arcsin\left(\frac{-p_z + l_3}{\sqrt{{l_1}^2 + {l_2}^2}}\right) - \arctan\left(\frac{l_1}{l_2}\right) + \frac{\pi}{2}$$

$$\theta_1 = \pm\arccos\left(\frac{p_x}{l_2\cos\left(\theta_2 - \frac{\pi}{2}\right) - l_1\sin\left(\theta_2 - \frac{\pi}{2}\right)}\right)$$

NAO Left Arm Inverse Kinematics

$$\theta_4 = -\left(\pi - \arccos\left(\frac{{l_3}^2 + {l_4}^2 - \sqrt{(s_x - p_x)^2 + (s_y - p_y)^2 + (s_z - p_z)^2}^2}{2 l_3 l_4}\right)\right)$$

$$\theta_2 = \pm \arccos\left(\frac{p_y - l_1 - \left(\frac{l_4 \sin\theta_4 T_{(2,2)}}{\cos\theta_4}\right)}{l_3 + l_4 \cos\theta_4 + l_4 \frac{\sin^2\theta_4}{\cos\theta_4}}\right) + \frac{\pi}{2}$$

$$\theta_3 = \arcsin\left(\frac{T_{(2,3)}}{\sin\left(\theta_2 - \frac{\pi}{2}\right)}\right)$$

$$\theta_3 = \pi - \arcsin\left(\frac{T_{(2,3)}}{\sin\left(\theta_2 - \frac{\pi}{2}\right)}\right)$$

$$\theta_1 = \pm \arccos\left(\frac{T_{(3,3)} - \frac{T_{(1,3)} \sin\theta_3 \cos\left(\theta_2 - \frac{\pi}{2}\right)}{\cos\theta_3}}{\cos\theta_3 + \frac{\cos^2\left(\theta_2 - \frac{\pi}{2}\right)\sin^2\theta_3}{\cos\theta_3}}\right) \qquad \text{if } \theta_3 \neq \frac{\pi}{2}$$

$$\theta_1 = \pm \arccos\left(\frac{T_{(1,3)}}{\cos\left(\theta_2 - \frac{\pi}{2}\right)\sin\theta_3}\right) \qquad \text{if } \theta_3 = \frac{\pi}{2}$$

NAO Left Leg Inverse Kinematics

$$T_{\text{llegInv}} = \left(R_x(\tfrac{\pi}{4}) \left(\left(A^0_{\text{Base}}\right)^{-1} T_{\text{lleg}} \left(A^{\text{End}}_6\right)^{-1} \right) \right)^{-1}$$

$$\theta_4 = \pm \left(\pi - \arccos \left(\frac{{l_1}^2 + {l_2}^2 - \sqrt{(s_x - p_x)^2 + (s_y - p_y)^2 + (s_z - p_z)^2}^2}{2 l_1 l_2} \right) \right)$$

$$\theta_6 = \arctan \left(\frac{p_y}{p_z} \right) \qquad \text{if } (l_2 \cos\theta_5 + l_1 \cos(\theta_4 + \theta_5)) \neq 0$$

$$T_{\text{llegInv}'} = \left((T_{\text{llegInv}})^{-1} \left(T^6_5 R_z(\pi) R_y(-\tfrac{\pi}{2}) \right)^{-1} \right)^{-1}$$

$$\theta_5 = \arcsin \left(- \frac{p_y (l_2 + l_1 \cos\theta_4) + l_1 p_x \sin\theta_4}{{l_1}^2 \sin^2\theta_4 + (l_2 + l_1 \cos\theta_4)} \right)$$

$$\theta_5 = \pi - \arcsin \left(- \frac{p_y (l_2 + l_1 \cos\theta_4) + l_1 p_x \sin\theta_4}{{l_1}^2 \sin^2\theta_4 + (l_2 + l_1 \cos\theta_4)} \right)$$

$$T_{\text{llegInv}''} = \left((T_{\text{llegInv}'})^{-1} \left(T^4_3 T^5_4 \right)^{-1} \right)^{-1}$$

$$\theta_2 = \pm \arccos T_{\text{llegInv}''(2,3)} - \frac{\pi}{4}$$

$$\theta_3 = \arcsin \left(\frac{T_{\text{llegInv}''(2,2)}}{\sin\left(\theta_2 + \frac{\pi}{4}\right)} \right)$$

$$\theta_3 = \pi - \arcsin \left(\frac{T_{\text{llegInv}''(2,2)}}{\sin\left(\theta_2 + \frac{\pi}{4}\right)} \right)$$

$$\theta_1 = \pm \arccos \left(\frac{T_{\text{llegInv}''(3,3)}}{\sin\left(\theta_2 + \frac{\pi}{4}\right)} \right) + \frac{\pi}{2}$$

Implementation

- **NAOKinematics**
  - software library written in C++
  - optimized for on-board execution
  - optimized for real-time execution
  - uses KMat for affine matrix transformations
  - implements all functions for forward and inverse kinematics
  - implements center-of-mass calculation

Outline

- Robot Kinematics
- NAO Kinematic Chains
- NAO Forward and Inverse Kinematics
- Results
- Live Demonstration
- Conclusion

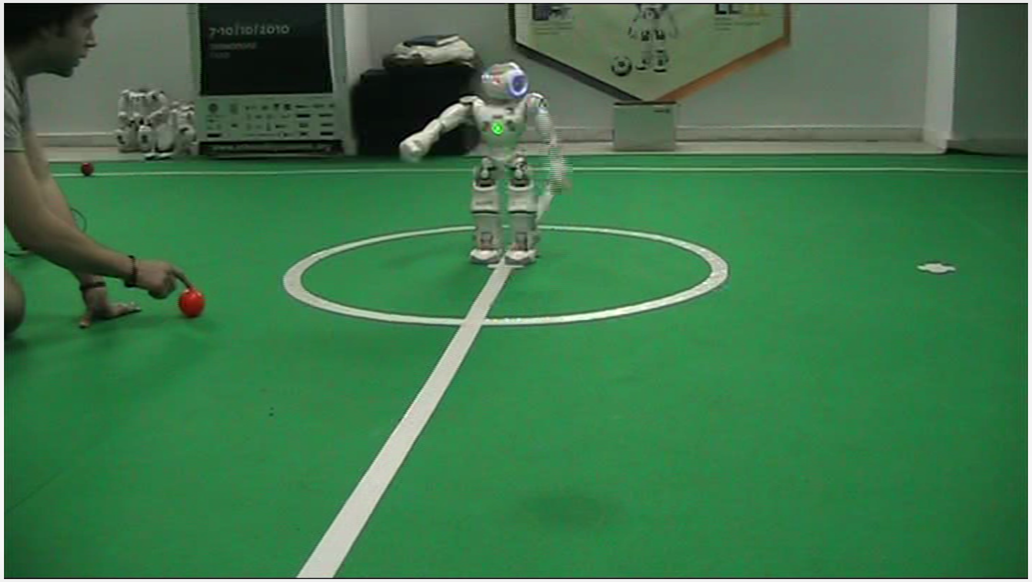
Real-Time Execution

Function	Execution Time in microseconds (us)
Forward Kinematics of the Head	54.28
Forward Kinematics of the Left Arm	66.72
Forward Kinematics of the Right Arm	69.54
Forward Kinematics of the Left Leg	80.88
Forward Kinematics of the Right Leg	80.78
Inverse Kinematics of the Head	70.79
Inverse Kinematics of the Left Arm	170.55
Inverse Kinematics of the Right Arm	200.00
Inverse Kinematics of the Left Leg	185.29
Inverse Kinematics of the Right Leg	184.85
Calculation of the Center of Mass	394.55

Demonstration I: Point to the Ball

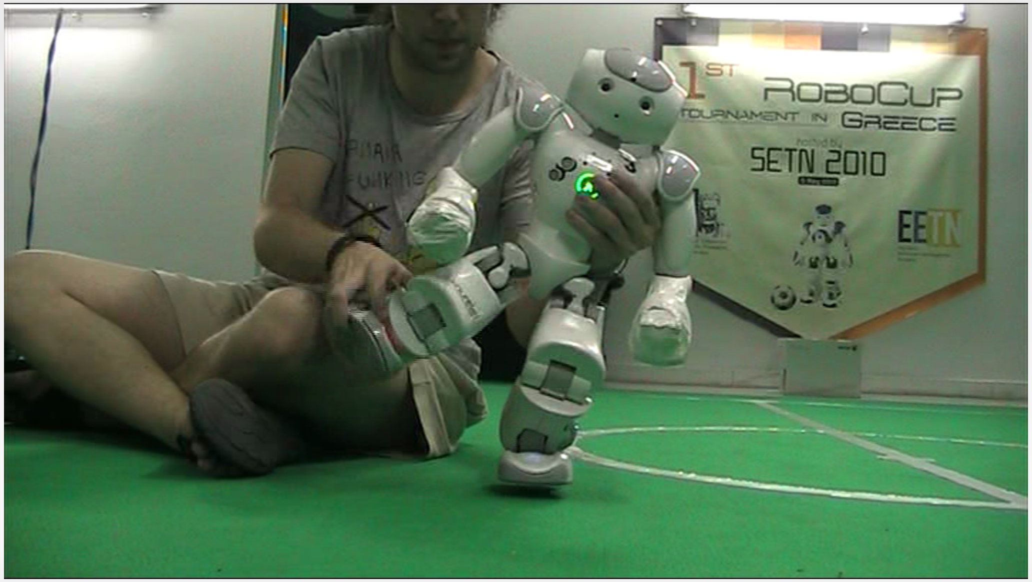
- *goal*: make NAO point to the ball with its stretched arms

- **Approach**
  - Kvision for ball recognition, local world state for ball tracking
  - run forward kinematics to get the height of the torso
  - transform the point of the ball to the work area of the NAO arms
  - run inverse kinematics for the arms for the resulting target point



Demonstration II: Basic Balancing

  - *goal*: make NAO move its foot to the projection of the CoM
- **Approach**
  - run the CoM calculation to obtain the current position of the CoM
  - obtain the torso plane orientation from the inertial unit
  - project the CoM to the floor plane, transform back to torso plane
  - run inverse kinematics for the leg for the resulting target point



Outline

- Robot Kinematics
- NAO Kinematic Chains
- NAO Forward and Inverse Kinematics
- Results
- Live Demonstration
- Conclusion

Live Demonstration

Outline

- Robot Kinematics
- NAO Kinematic Chains
- NAO Forward and Inverse Kinematics
- Results
- Live Demonstration
- Conclusion

Comparison

- **Aldebaran**
  - numerical iterative approximation based on Jacobian
  - suffers from singularities
- **B-Human**
  - analytical solution for the legs only
  - assumption: the foot is always parallel to the *x*-*y* plane
- **QIAU**
  - analytical solution for the legs only
  - problem: we could not reproduce their results
- **Our work**
  - analytical solution for the entire robot
  - accuracy, efficiency, no singularities

Conclusion

- **Future Work**
  - kick engine
  - dynamic balancing
  - omni-directional walk
  - visual signals for communication
- **Lessons learned**
  - kinematics is a hard problem, especially the inverse ones
  - math is magic!
  - it is fun to apply math on robots

Thank you

010101000110100001100001011011100110101101110011

Questions

Questions?