Locomotion Control and Environment Interactions Handling for Multi-legged Walking Robots

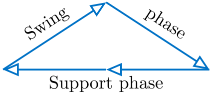
Petr Čížek

Computational Robotics Laboratory (CRL)
Department of Computer Science
Faculty of Electrical Engineering
Czech Technical University in Prague

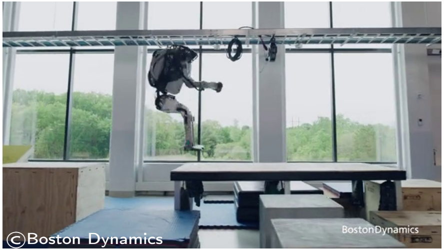
B4M36UIR - Artificial Intelligence in Robotics

Legged locomotion – mimicking natural way of transport

- **Discrete nature** – Separate footholds – does not need a continuous path of support.
- Motion is achieved using legs which alternate between:
  - Support (stance) phase - supporting the body,
  - Swing phase - moving leg to the new foothold.
- Gait prescribes the repetitive pattern in which the legs move; i.e, alternation of swing and stance phases of legs.



- T_{stance} .. stance duration, T_{swing} .. swing duration
- Duty factor: $\beta = T_{\text{stance}} / (T_{\text{stance}} + T_{\text{swing}})$.
  - $\beta > 0.5$ .. walking.
  - $\beta < 0.5$ .. running (there is a flight phase where no leg is touching ground).

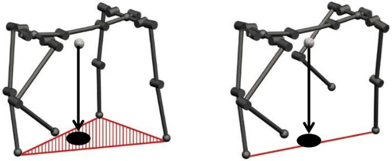


©Boston Dynamics

Legged locomotion - gait examples and stability

- Gait examples – biologically inspired – **pentapod, tetrapod, tripod**, walk, amble, gallop, ...

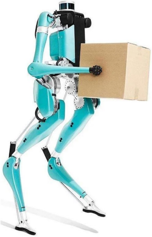
- Stability requirement: projection of Center of Gravity (CoG) is inside of the Support polygon.
- Support polygon – horizontal region; vertical projection in the direction of gravity vector G of the convex hull of the contact points.
- **Static stability** – robot will not fall when all joints freeze.
- **Dynamic stability** – requires actively maintaining balance.



Legged robots – overview – 1 and 2-legged platforms

- **Complex morphology** with many Degrees of Freedom (DoF).
  - (Recall) Controllable DoF (CDoF) vs. Total DoF (TDoF).
- **1-legged** – Requires continuous hopping.
- **2-legged (Humanoid)** – Requires continuous dynamic balancing.




Salto	Hopper	Cassie	Digit	Atlas	Talos	i-cub
4 CDoF	3 CDoF	10 CDoF	19 CDoF	28 CDoF	32 CDoF	53 CDoF
10 TDoF	9 TDoF	20 TDoF (10 CDoF + 4 passive joints + 6 DoF)				

Legged robots – overview – 4-legged platforms

- Compromise between 2 and 6 legs.
- Capable of statically stable locomotion using 3:1 gait.

Anymal-C
12 CDoF

Cheetah mini
12 CDoF

Spot
12 CDoF

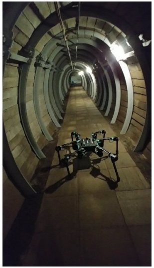
Magneto
12 CDoF

Pleurobot
27 CDoF

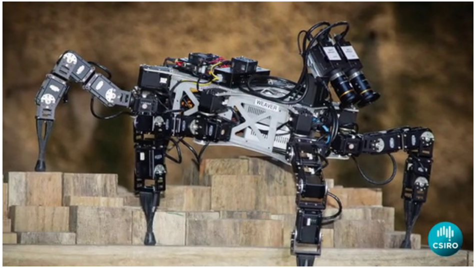
Legged robots – overview – 6-legged platforms

- Minimum number of legs for two stride ($\beta = 0.5$) statically stable locomotion.

RHex
6 CDoF

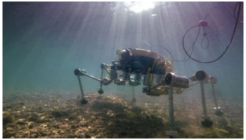


HEBI Lily
18 CDoF



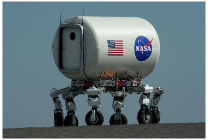
Weaver
10 CDoF

Tractor Timberjack
Harvester



Silver2
Underwater inspection

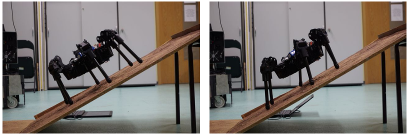
Crabster
Underwater inspection

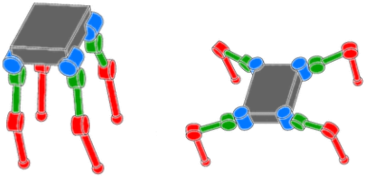


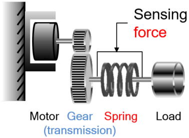
Athlete
NASA rover

Legged robots – overview – other characteristics

- **Number of DoF per leg** – affects leg reachability and platform maneuverability.
  - 1,2,3,4,5+
  - Passive joints that adds up to TDoF are sometimes used.
- **Leg morphology** – affects stability and locomotion speed.
  - Mammalian – lower joint torques, better leg dynamics – less inertia during swing.
  - Reptilian/Insect – wide posture, more stable, slower.
- **Actuation principles:**
  - Large reduction gearbox drive – stiff, strong, power efficient, but can't withstand impacts.
  - Direct drive – compliant (single reduction element between shaft and axle), but power inefficient.
  - Series Elastics Actuator – actuator with added compliance.
  - Hydraulics/Pneumatics – complicated, large.







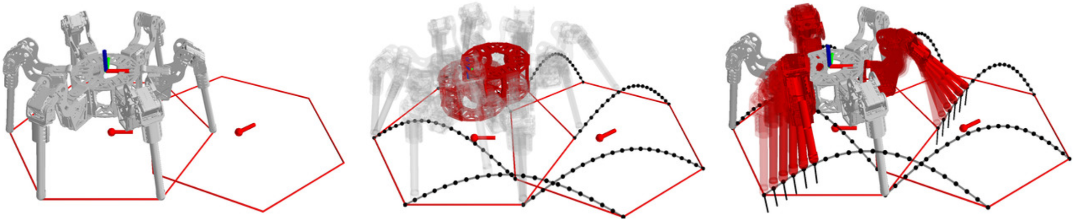
Multi-legged robot locomotion control

- Kinematic control output the reference joint angles according to the desired trajectory of the limbs.
- Dynamic control output the reference joint torques to satisfy the desired trajectory of the limbs.
- Kinematic control problem decomposition:
  - Body motion
  - Leg motion

  Can be decoupled and executed consecutively.

  - Contact sensing
  - Contact resolution

  For terrain-aware locomotion.



Sidenote – Forward Kinematics Task (FKT)

- **Forward Kinematics Task** – calculate position of the end-effector from known joint angles.

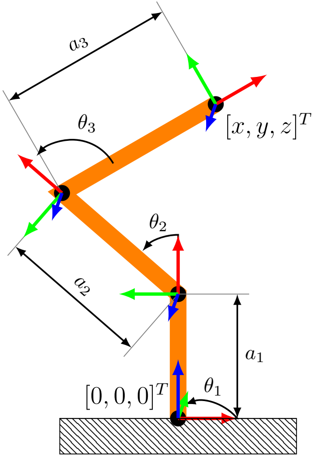
- a_1, a_2, a_3 .. lengths of the links
- $\theta_1, \theta_2, \theta_3$ .. joint angles
- $[x, y, z]^T$ .. end-effector position in global reference frame

- **FKT** (using Denavit-Hartenberg (DH) notation)

$$[x, y, z, 1]^T = \mathbf{M}_1^0\,\mathbf{M}_2^1\,\mathbf{M}_3^2\,[0, 0, 0, 1]^T, \tag{1}$$

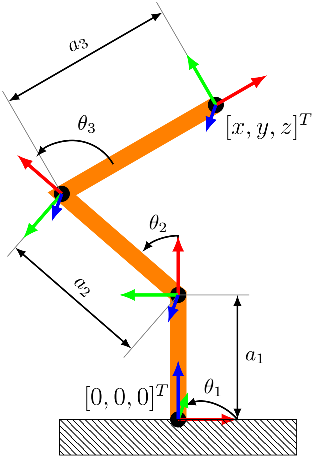
$$\mathbf{M}_i^{i-1} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}\,c_{\alpha_i} & s_{\theta_i}\,s_{\theta_i} & a_i\,c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}\,c_{\alpha_i} & -c_{\theta_i}\,s_{\alpha_i} & a_i\,s_{\psi_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

with DH parameters $\theta_i, \alpha_i, a_i, d_i$. And $c_{\theta_i}, s_{\theta_i}$ denoting $\cos\theta_i, \sin\theta_i$.



Sidenote – Inverse Kinematics Task (IKT)

- **Inverse Kinematics Task** – calculate joint angles to reach the given end-effector position.



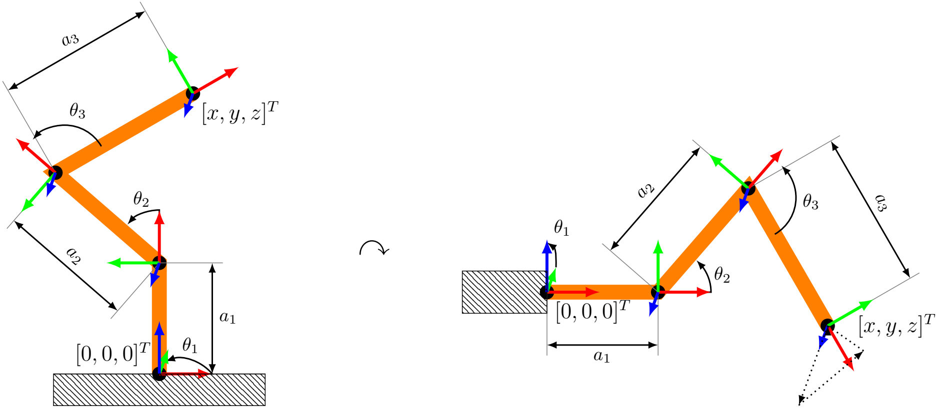
- Usually hard to find analytical solution by direct inversion of the FKT.
- Can be solved as iterative optimization problem, e.g., using FABRIK (Forward And Backward Reaching Inverse Kinematics)

A. Aristidou, Graphical models, 2011

- Usually there are multiple solutions, singular points have ∞ solutions.
- For general 3 DoF arm analytical solution exists. First calculate θ_1 to orientate the robot motion plane and then θ_2, θ_3 using a cosine law from link lengths constraints. For any $[x, y, z]^T$ there will be $0, 1, 2, 3, 4$, or ∞ possible solutions.

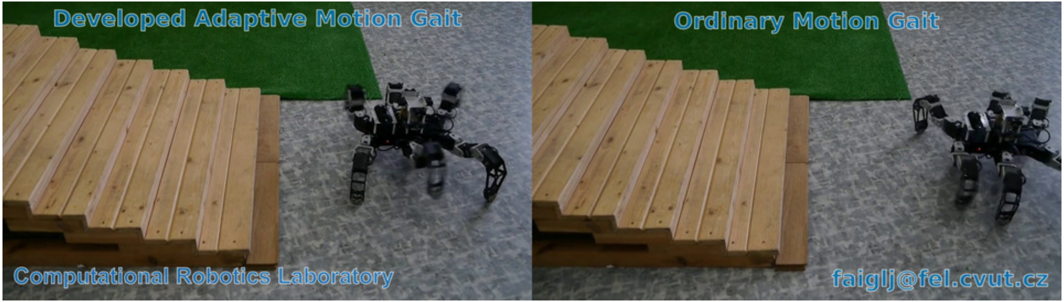
Multi-legged robot locomotion contol – contd.

- Manipulator kinematics is similar to leg kinematics.
- Using IKT to calculate the joint angles to make the leg follow the prescribed trajectory.



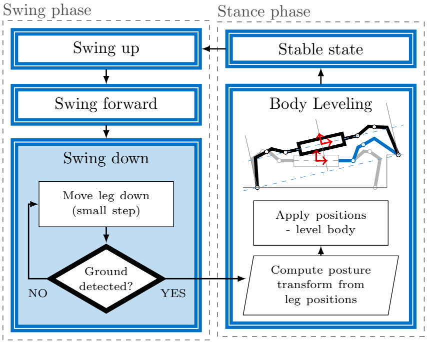
Blind locomotion vs. Adaptive locomotion

- **Blind locomotion** is fast but only usable on flat terrain or with compliant platform (e.g., RHex).
- **Adaptive locomotion** takes into account environment interactions (Reactive paradigm).
- **Environment interaction** appears anywhere along the robot's morphology (footstrikes, bumping into obstacles, human-machine interaction, collaborative manipulation, etc.)
- Environment interaction handling:
  - Implicit – contact occurs emergently. Reaction is part of the control rule.
  - Explicit – event is detected, identified, and explicit reaction is triggered. interaction.



Adaptive locomotion control with explicit interaction handling

- Handling foot-strike events – Using proprioception (e.g., tactile sensors, position feedback, joint force feedback, etc.) to stop the leg motion when it touches the ground.
- Discrete body motion



Swing phase

Swing up

Swing forward

Swing down

Move leg down
(small step)

Ground
detected?

NO

YES

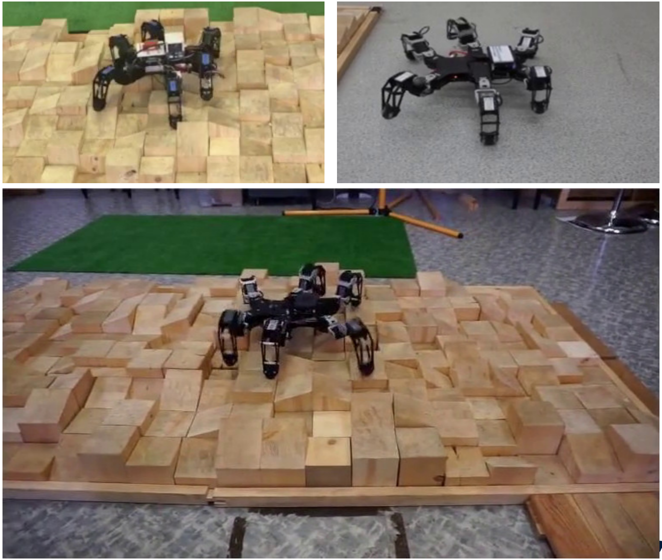
Stance phase

Stable state

Body Leveling

Apply positions
- level body

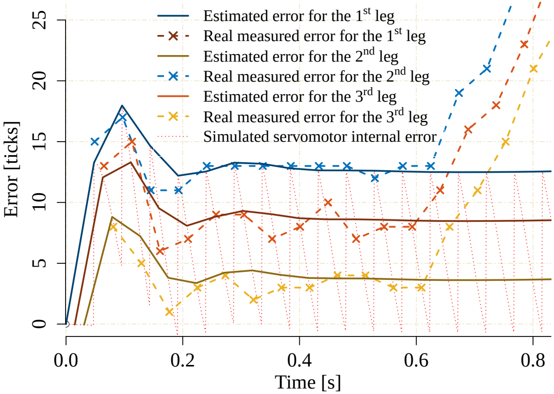
Compute posture
transform from
leg positions



Example – foot contact sensing using proprioceptive modalities

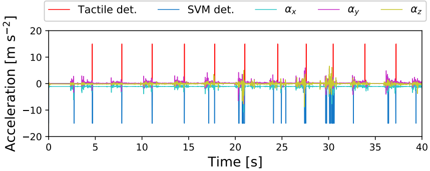
- Detection using joint position error – Detect deviation of real joint position from the commanded joint position taking into account leg and controller dynamics.

Faigl et al., RAS 2018

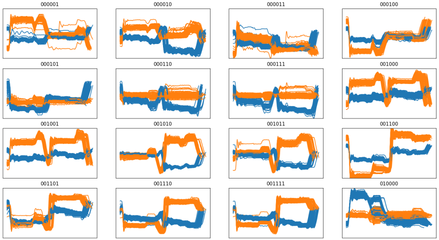


- Detection using accelerometric data – NN and SVN detection of tactile events from stream of acc data.

Čížek et al., IROS 2018

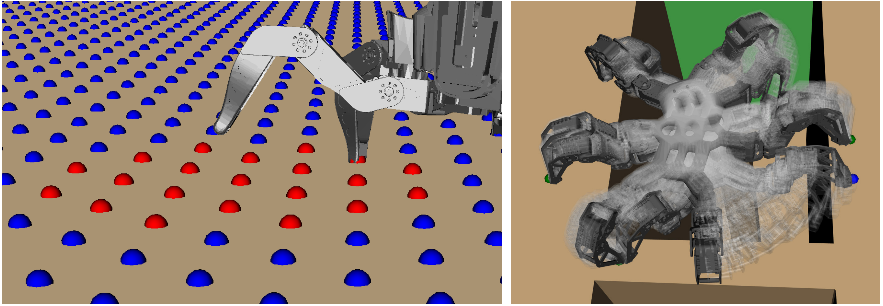


- Leg contact detection failure detection.



Kinematic control – further improvement ideas – examples

- **Continuous body motion** – improves locomotion speed, reduces frame stress, (looks better).
- **Deliberative control** – integration of exteroceptive sensing – foothold selection and precise motion planning.
- **Full-body tactile event detection** – external wrench estimation for interaction handling.
- **Robustness to morphology and parameters changes** – robustness to, e.g., change of weight of the robot or losing/damaging leg.

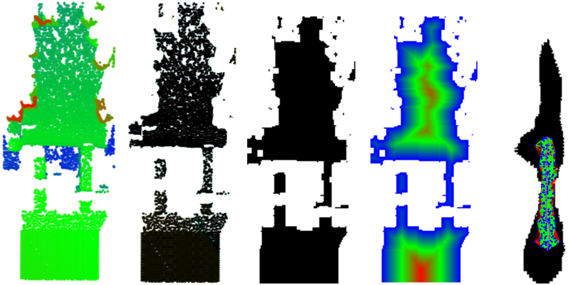


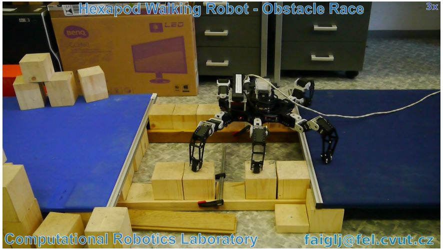
Left: candidate footholds for leg motion. Right: phased motion over obstacle.

Deliberative control – precise motion planning – example

- Using **hierarchical paradigm** – Sense, Plan, Act.
- Sensing using depth camera, fusion into elevation map, thresholding untraversable terrain by height, binary closing to fill in holes in map, distance transform to score the footholds, convolution with body pattern to get body-pose map, planning using PRM over body-pose map, acting using kinematic control.

Čížek et al., IROS 2017





From left to right: Elevation map, Threshold map, Map after binary closing, Distance map, Body pose map with PRM.

Model-based External Wrench Estimation – example

- Estimation of external wrench acting on the robot – requires dynamic modelling of the robot.

Thank you for your attention!