Legged Robots

Lecture 3

Overview of control approaches

Auke Jan Ijspeert

Different control approaches

- There are many different control approaches for legged robots.
- Here I mean “control” in the large sense, i.e. control + trajectory planning

- There are three broad categories:
    - **Model-based methods**, strongly influenced by traditional control engineering
    - **Learning-based methods**, strongly influenced by machine learning
    - **Bio-inspired approaches**, strongly influenced by computational neuroscience and biomechanics

Model-based methods

- Most extensively used
- Extensive use of models (mainly dynamic, sometimes only kinematic models)
- Often use of **simple models**: LIP or SLIP
- Sometimes use of **full models**, e.g. full dynamics and inverse dynamics.
- Sometimes use of two types of models, simple and full models together (different control layers)
- Increasing use of **optimization** (e.g. optimal control and model predictive control)

Examples of model-based approaches

Model-based control:

1. **trajectory based methods (ZMP)**
2. Virtual leg control (Raibert)
3. Virtual model control (Pratt et al)
4. Hybrid Zero Dynamics control
5. Planning methods (Little dog project)
6. Inverse dynamics and model predictive control (MPC)

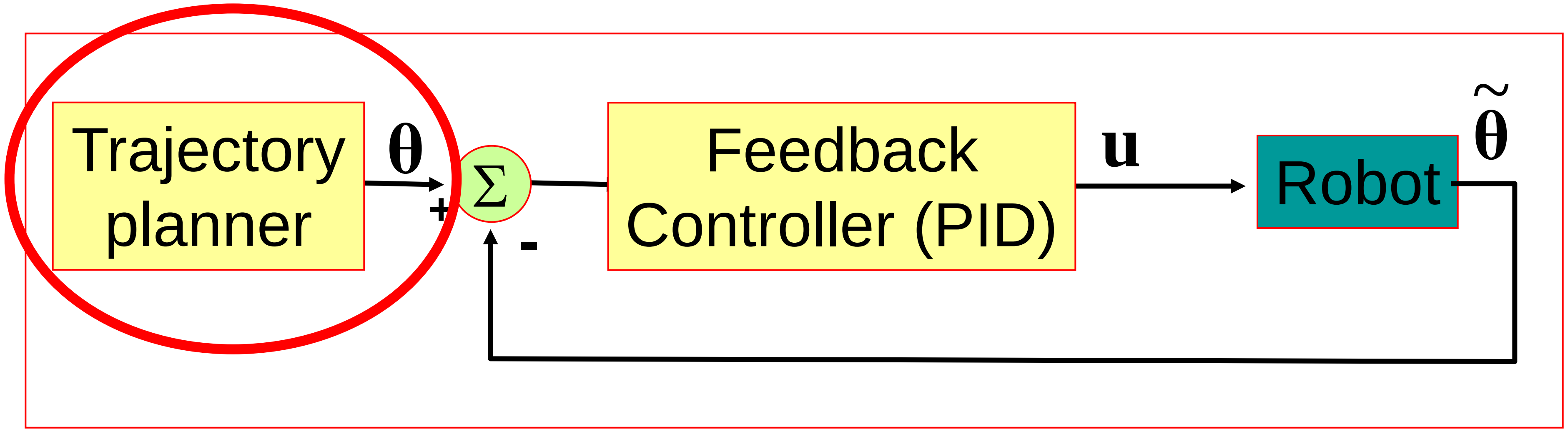
Trajectory based methods

- Main idea: design **walking kinematic trajectories** (i.e. joint angles over time), and use the dynamic equations to test and prove that locomotion is stable

- Trajectories were initially designed by trial-and-error, from human recordings, and/or based on simple models like LIP (now most people use optimization)

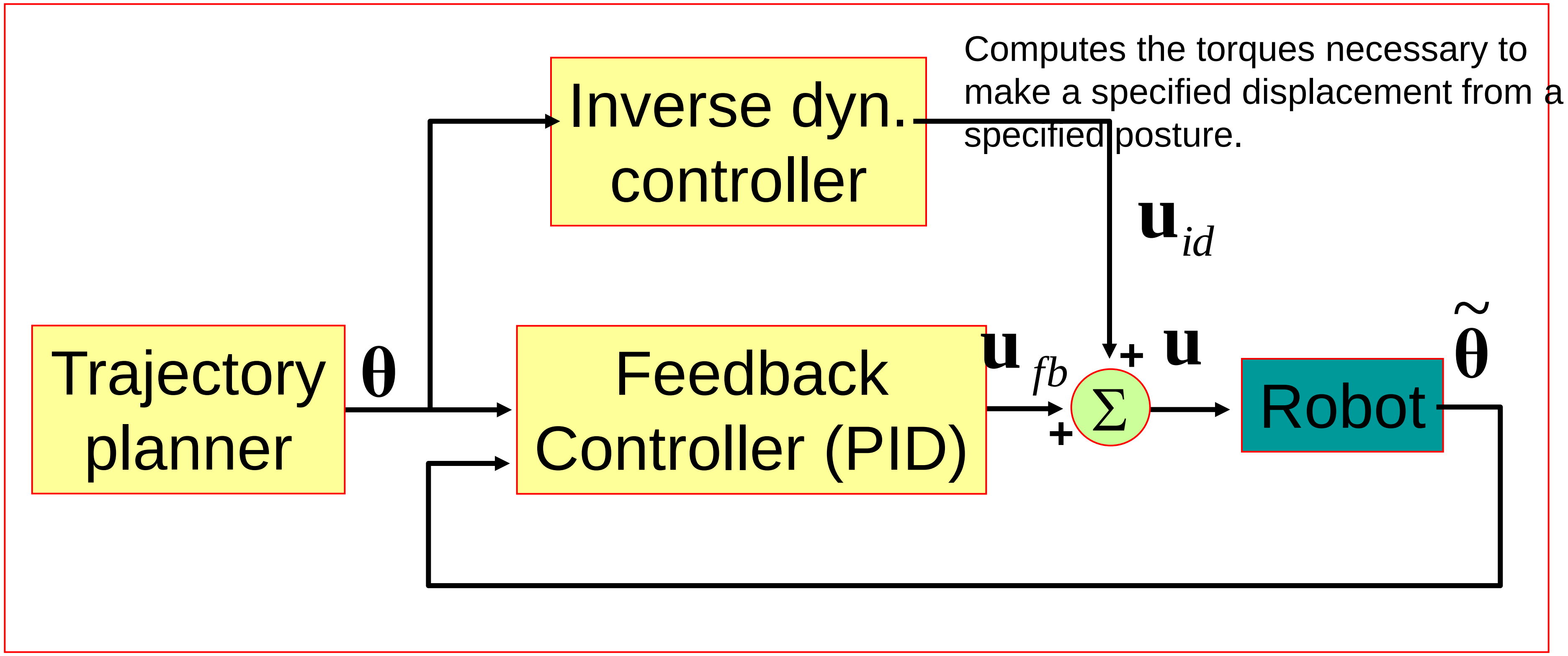
- Most used stability criterion: Zero Moment Point (ZMP) (Vukobratovic 1990)

Minimalistic control diagram



$\boldsymbol{\theta}$	Desired robot posture
$\tilde{\boldsymbol{\theta}}$	Actual robot posture
$\mathbf{u}$	Command (torque)

More sophisticated: Inverse dynamics



θ Desired robot posture

$\tilde{\theta}$ Actual robot posture

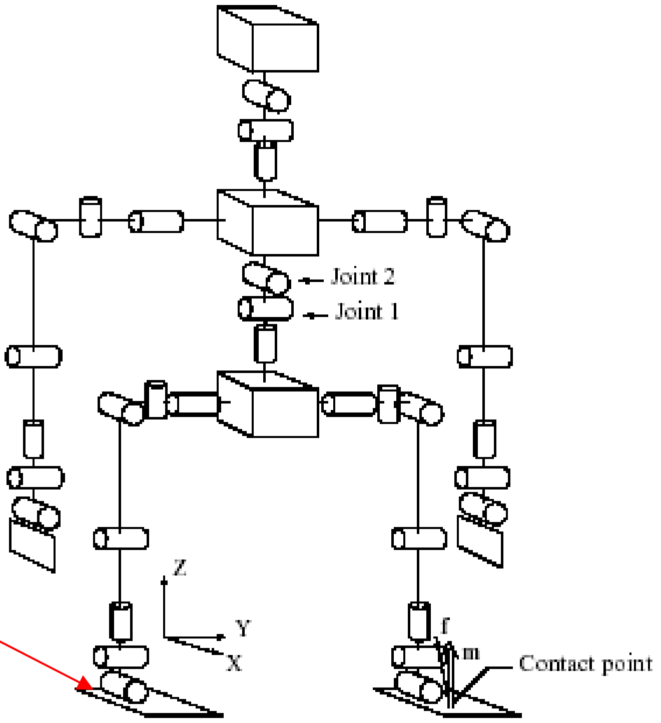
$\mathbf{u}$ Command (torque)

A PID does not know anything about the physics of the body (e.g. gravity, inertias).

The inverse dynamics uses knowledge of the configuration and characteristics of the robot to compute the exact torques necessary to make a displacement

Trajectory-based with ZMP

Foot-print polygon



Locomotion is stable if the ZMP remains within the foot-print polygons over time

Trajectory-based with ZMP

- Example: (early) Honda robot, Asimo

Note the crouched gait, with the CoM staying almost horizontal, similarly to the LIP model

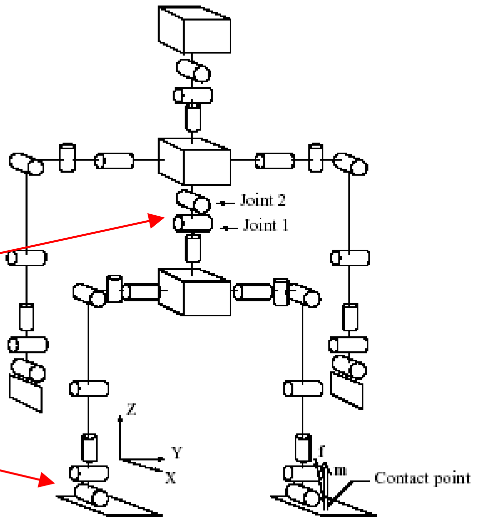
Trajectory-based with ZMP

Most used method in earlier papers:

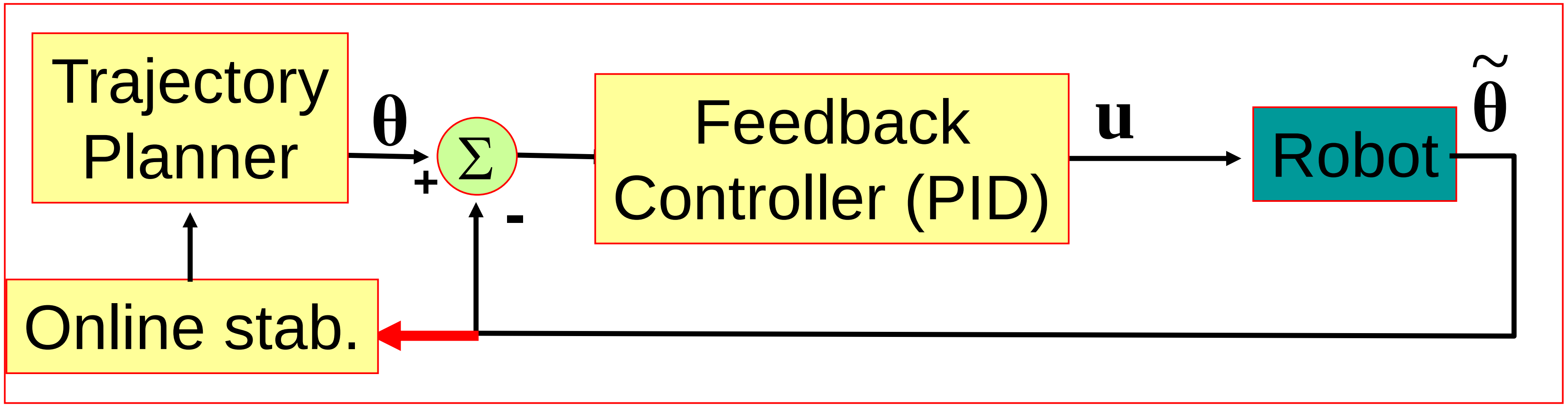
1. Human motion capture, educated guesses for getting trajectories, and/or use simple models like LIP for online footstep planning.
2. Modify trajectories offline such that locomotion is stable according to the ZMP criterion
3. Add online stabilization to deal with perturbations

Example of **online stabilization**:

- Use of two hip actuators to manipulate the ZMP
- Alternatively: use of ankle actuators



Control diagram:
ZMP + online stabilization



θ	Desired robot posture
$\tilde{\theta}$	Actual robot posture
u	Command (torque)

Trajectory-based with ZMP: conclusions

Pros:

- Well-defined methodology for proving dynamic balance
- Well-suited for expensive robots that should never fall

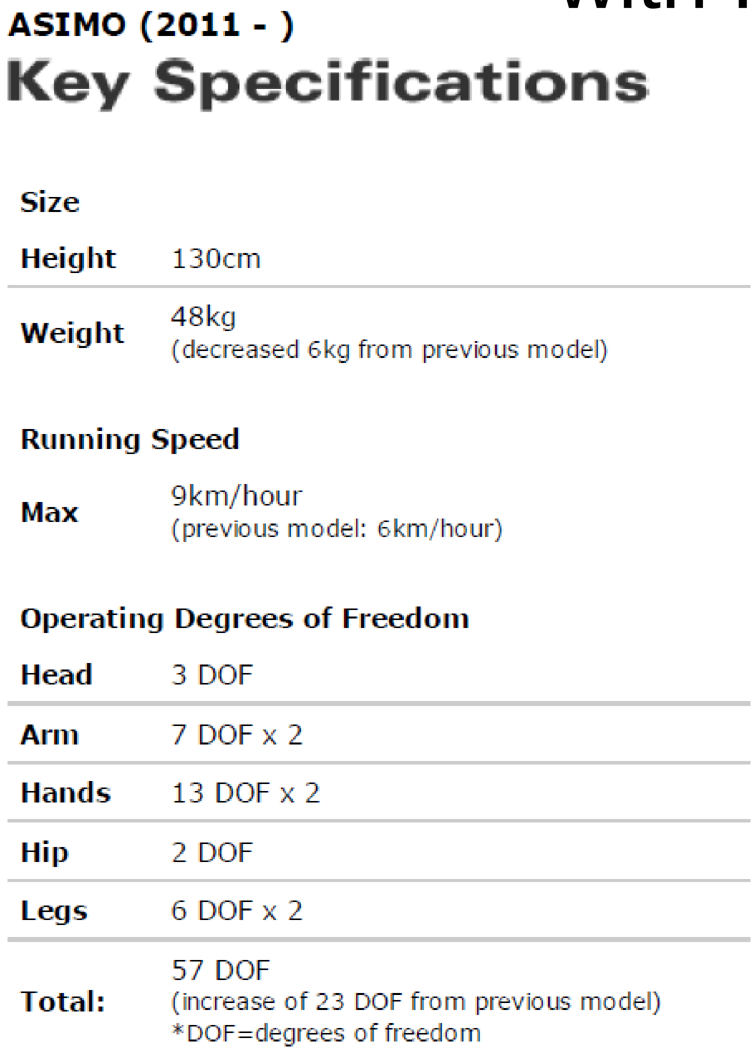
Cons:

- Requires a perfect knowledge of the robot's dynamics and of the environment
- Defining good trajectories can be time-consuming
- Energetically inefficient (requires stiff actuation, and often used with crouched-knee walking)

Reference: Vukobratovic, M. and Borovac, B. (2004). Zero-moment point - thirty five years of life. International Journal of Humanoid Robotics, 1(1):157–173.

Kajita and Espiau. 2008. "Legged Robots." In *Springer Handbook of Robotics*, edited by Bruno Siciliano and Oussama Khatib, 361–89. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-30301-5_17.

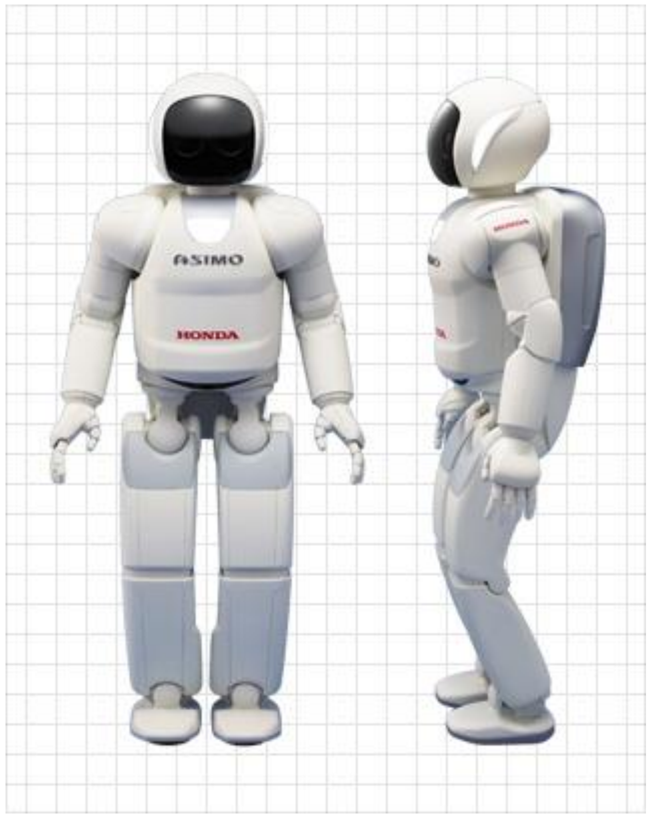
Note the huge progress with Honda’s Asimo



ASIMO (2011 -)

Key Specifications

Size	
Height	130cm
Weight	48kg (decreased 6kg from previous model)
Running Speed	
Max	9km/hour (previous model: 6km/hour)
Operating Degrees of Freedom	
Head	3 DOF
Arm	7 DOF x 2
Hands	13 DOF x 2
Hip	2 DOF
Legs	6 DOF x 2
Total:	57 DOF (increase of 23 DOF from previous model) *DOF=degrees of freedom



- http://world.honda.com/ASIMO/
- http://world.honda.com/ASIMO/video/index.html

More recent ASIMO's controllers use the DCM

See the practical and Milad Shafiee's presentation in a few weeks

Takenaka, Toru, Takashi Matsumoto, and Takahide Yoshiike. "Real time motion generation and control for biped robot-1 st report: Walking gait pattern generation." 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009.(Citation:299)

Examples of model-based approaches

Model-based control:

1. Trajectory based methods (ZMP)
2. **Virtual leg control (Raibert)**
3. Virtual model control (Pratt et al)
4. Hybrid Zero Dynamics control
5. Planning methods (Little dog project)
6. Inverse dynamics and model predictive control (MPC)

Virtual Leg Control

- Developed by Marc Raibert and colleagues (CMU, MIT, Boston Dynamics) for **hopping/running robots** (i.e. with short flight phases). **Closely related to the SLIP model**.

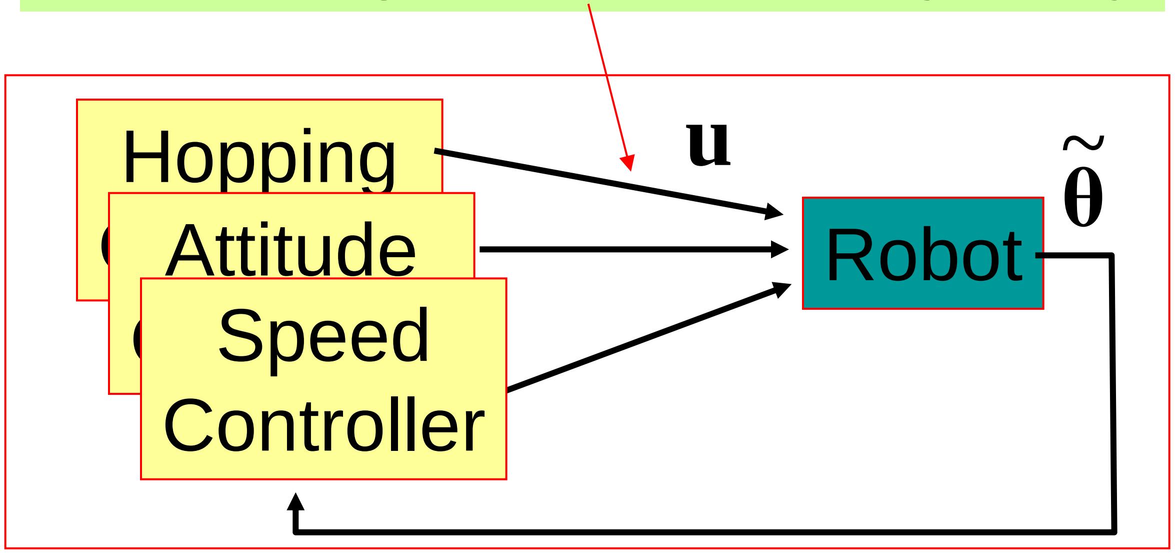
- One- two- and four-legged robots controlled by a similar approach

- Key idea: to **decompose the problem into three (independent) control loops**:

1. *Hopping control*: Supporting the body with a vertical bouncing motion
2. *Attitude control*: Controlling the attitude of the body by servoing the body through hip torques during stance
3. *Speed control:* Placing the feet in key locations on each step using symmetry principles (a.k.a. *Raibert's heuristics*)

Control diagram: Virtual Leg Control

Directly produces torques, no tracking of a desired trajectory



$\tilde{\theta}$ Actual robot posture

$\mathbf{u}$ Command (torque and force)

Robots at MIT LegLab

Similarities between 1, 2, and 4 legs

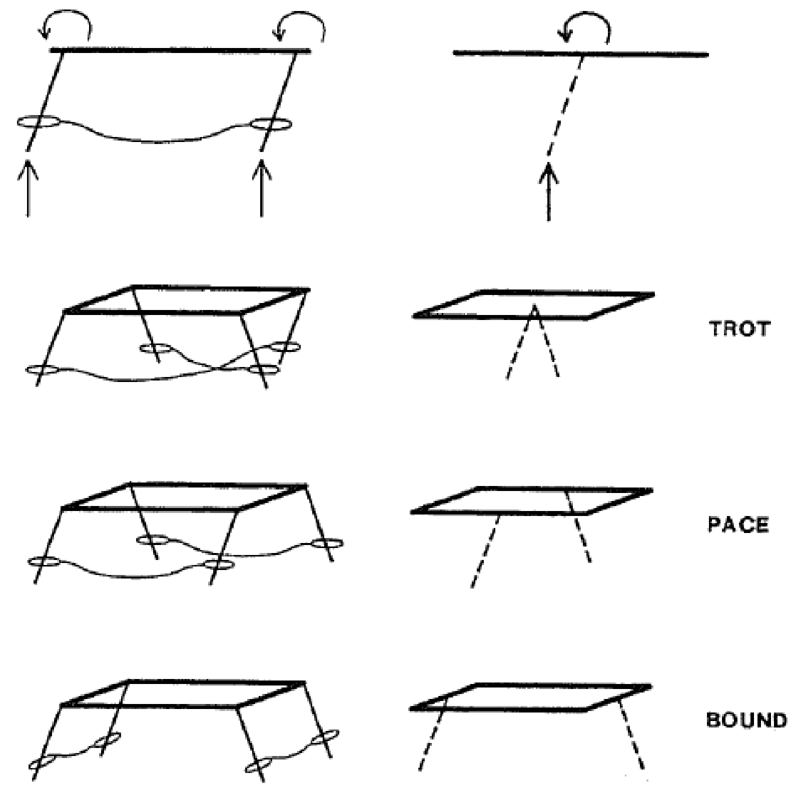


Fig. 3. Virtual legs. When two legs are coordinated to act in unison, they can be represented by a functionally equivalent *virtual leg*. The virtual leg and the original pair of physical legs both exert the same forces and moments on the body, so they both result in the same behavior. When each pair of legs is replaced by a virtual leg, the trot, the pace, and the bound are transformed into equivalent virtual biped gaits. One virtual leg is used for support at a time. Sutherland first introduced the concept of the virtual leg to simplify the design of a six-legged walking machine (Sutherland and Ullner, 1984).

Key idea to extend to more legs:

When two legs are coordinated to act in unison, they can be represented by a functionally equivalent **virtual leg**

Same forces and moments as the pair of legs

Note: this is not strictly a model-based approach (no explicit model used in the control loop), but the SLIP model has been important in designing the heuristic control loops.

Raibert, 1990, Trotting, pacing and bounding by a quadruped robot, Journal of Biomechanics Volume 23, Supplement 1, 1990, Pages 79–81, 83–98

Virtual Leg Control: summary

Pros:

- The most impressive locomotion skills in current robots (e.g. BigDog)
- Quite simple to implement (e.g. no complex models needed)

Cons:

- Needs very powerful actuators (hydraulic)
- No (analytical) proof of stability
- Only applicable to hopping/running robots (no walking)

References:

- Raibert, M. H. and Hodgins, J. K. (1993). Legged robots. In Beer, R. D., Ritzmann, R. E., and McKenna, T. M., editors, Biological Neural Networks in Invertebrate Neuroethology and Robotics, pages 319–354. Academic Press.
- M.H. Raibert, M. Chepponis, and H. Benjamin Brown, "Running on Four Legs As Though They Were One," IEEE Journal of Robotics and Automation, Vol. RA-2, No. 2, June, 1986, pp. 70 - 82.

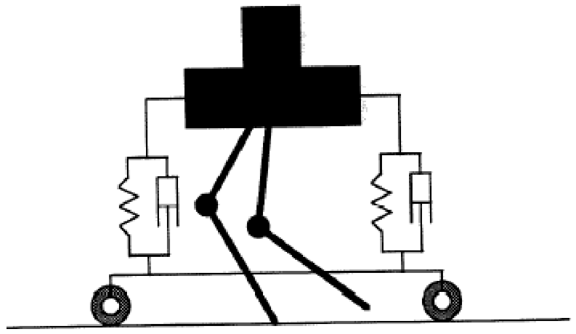
Examples of model-based approaches

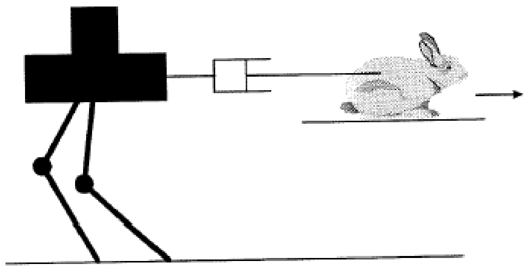
Model-based control:

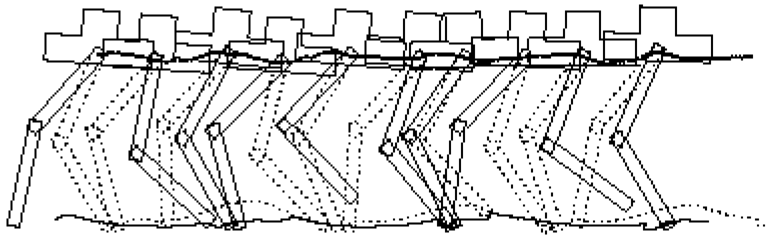
1. Trajectory based methods (ZMP)
2. Virtual leg control (Raibert)
3. **Virtual model control (Pratt et al)**
4. Hybrid Zero Dynamics control
5. Planning methods (Little dog project)
6. Inverse dynamics and model predictive control (MPC)

Virtual Model Control

- Nice example of model-based control: Virtual Model Control (G.Pratt)
- Idea: create **virtual elements** to keep the robot upright and have it move forward
- Then compute the necessary torques such that the robot motors replicate the effect of those virtual elements







Virtual granny walker for balance control

Virtual bunny for velocity control

Virtual Model Control

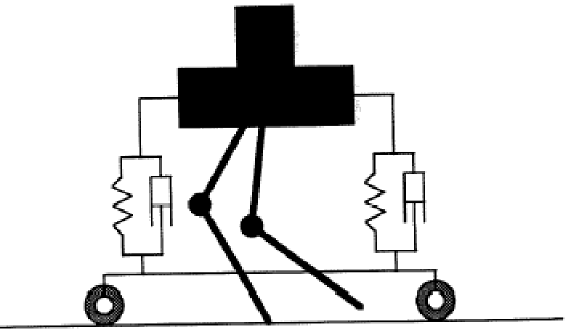
- For each virtual element producing a force F, the joint torque needed to produce that virtual force can be computed with:

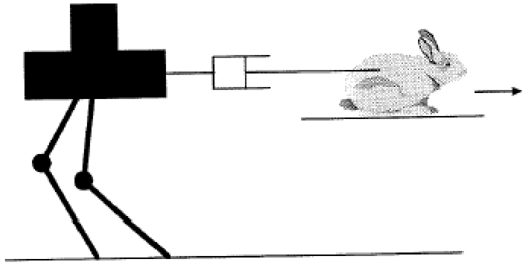
$$\vec{T} = \mathbf{J}^T \vec{F}$$

- **J** is the *Jacobian* relating the reference frame of the virtual element to the robot

$$\vec{x} = f(\vec{\theta})$$

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$





Example

The forward kinematic map from frame $\{A\}$ to frame $\{B\}$ of this example is as follows:

$$ {}^A_B\vec{X} = \begin{bmatrix} x \\ z \\ \theta \end{bmatrix} = \begin{bmatrix} -L_1 s_a - L_2 s_{a+k} \\ L_1 c_a + L_2 c_{a+k} \\ -\theta_h - \theta_k - \theta_a \end{bmatrix}, \qquad (1)$$

where s_a, s_{a+k}, c_a, and c_{a+k} denote $\sin(\theta_a)$, $\sin(\theta_{a+k})$, $\cos(\theta_a)$, and $\cos(\theta_a + \theta_k)$, respectively.

Partial differentiation produces the Jacobian,

$$ {}^A_B J = \begin{bmatrix} -L_1 c_a - L_2 c_{a+k} & -L_2 c_{a+k} & 0 \\ -L_1 s_a - L_2 s_{a+k} & -L_2 s_{a+k} & 0 \\ -1 & -1 & -1 \end{bmatrix}. \qquad (2)$$

The Jacobian relates the virtual velocity ${}^A_B\dot{\vec{X}}$ between frames A and B with the joint velocities $\dot{\vec{\Theta}} = [\theta_a\ \theta_k\ \theta_h]^T$

$$ {}^A_B\dot{\vec{X}} = {}^A_B J\ \dot{\vec{\Theta}} \qquad (3)$$

and the virtual force $\vec{F} = [f_x\ f_z\ f_\theta]^T$ to joint torque $\vec{\tau} = [\tau_a\ \tau_k\ \tau_h]^T$

$$ \vec{\tau} = ({}^A_B J)^T\ ({}^A_B\vec{F}). \qquad (4)$$

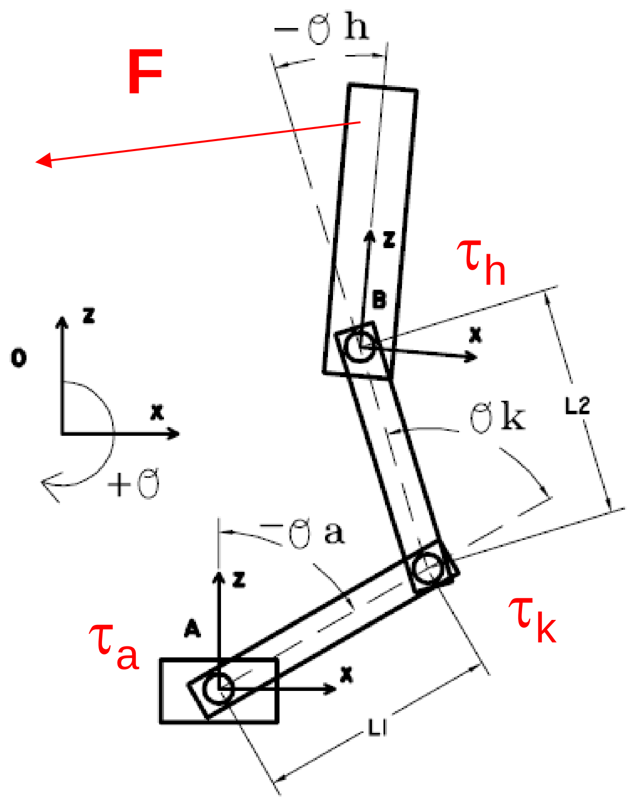
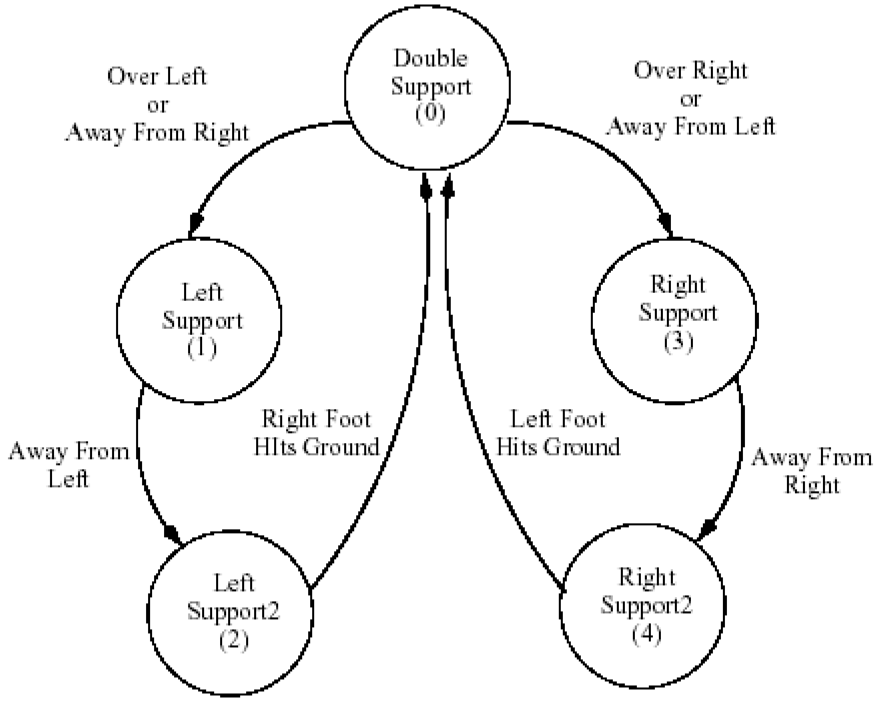


Fig. 3. Single-leg implementation. Reaction frame $\{A\}$ is assumed to be in the same orientation as reference frame $\{O\}$ so that ${}^O_A R = I$.

Mapping the forces of the virtual elements to torques in the motors

Virtual Model Control

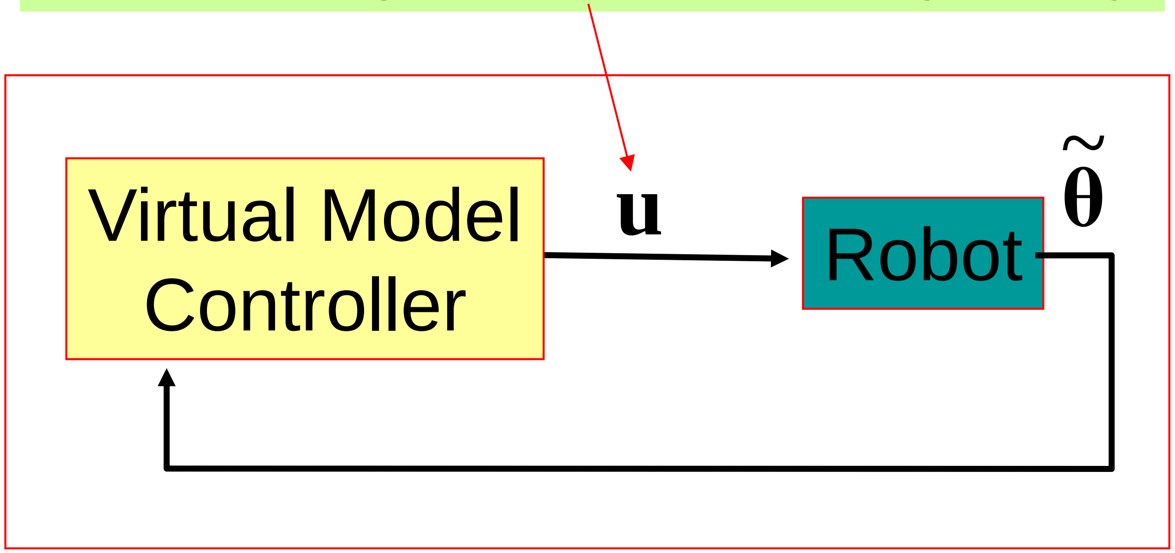
Only some motors should be activated at particular phases in the locomotor cycle



Finite state machine (set of if-then rules) for cycling through different actuation phases

Control diagram: Virtual Model Control

Directly produces torques, no tracking of a desired trajectory



$\tilde{\theta}$ Actual robot posture

$\mathbf{u}$ Command (torque)

Virtual Model Control

- Example: Flamingo robot at MIT Leg LAB

Virtual Model Control: summary

Pros:

- Intuitive way of designing a controller
- Does not need an accurate model of the environment
- Robust against pertubations
- No need of a dynamic model (only a kinematic model)

Cons:

- Need to make sure that the virtual forces can actually be generated by the robot’s motors
- Cannot be used for running gaits??

Reference: Pratt et al, Virtual Model Control: An intuitive approach for bipedal locomotion, The International Journal of Robotics Research, Vol. 20, No. 2, 129-143 2001 .

Examples of model-based approaches

Model-based control:

1. Trajectory based methods (ZMP)
2. Virtual leg control (Raibert)
3. Virtual model control (Pratt et al)
4. **Hybrid Zero Dynamics control**
5. Planning methods (Little dog project)
6. Inverse dynamics and model predictive control (MPC)

Hybrid Zero Dynamics control

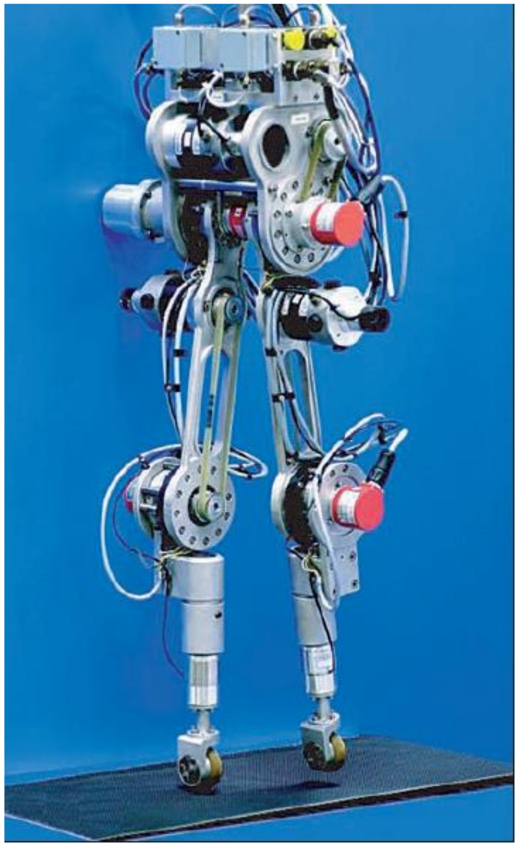
Developed by Grizzle, Chevallereau and others for the Rabbit and MABEL robots

Specific property: no feet

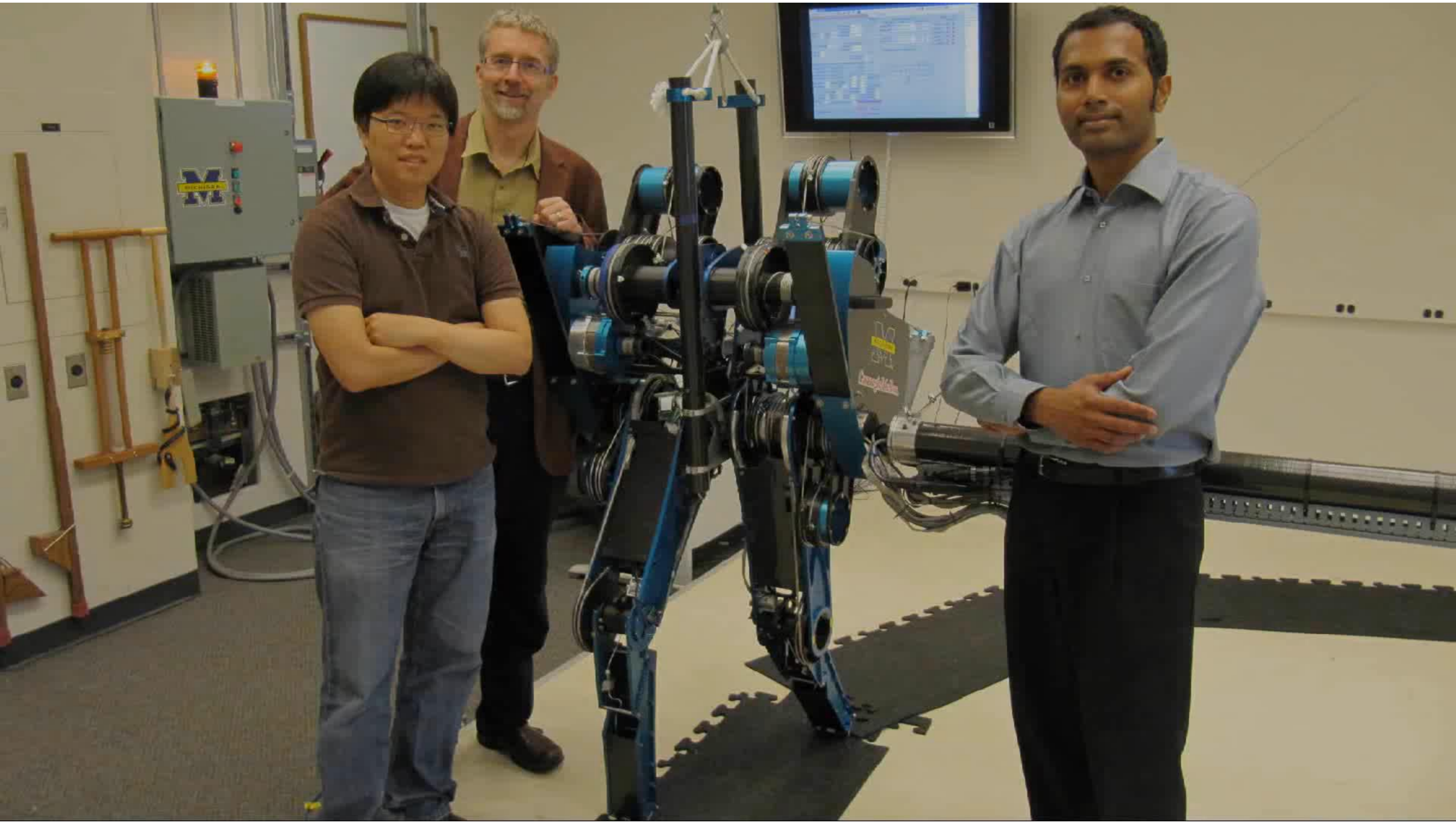
Very nice theoretical framework to obtain **provably asymptotically stable walking and running gaits**

Hybrid zero dynamics: method to reduce the number of degrees of freedom to control. This is done using **virtual constraints** to link different DOFs.

Existence and stability of stable gaits can be determined on the basis of a **scalar Poincaré return map**


Rabbit robot

Hybrid Zero Dynamics control, MABEL robot



https://www.youtube.com/watch?v=xlOwk6_xpWo
http://www.youtube.com/user/DynamicLegLocomotion

Hybrid Zero Dynamics: summary

Pros:

- The most complete theoretical foundation
- Analytical proof of stability

Cons:

- Not so easy to understand when applied to system with a high number of DOFs.

References:

- C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E.R. Westervelt, C. Canudas-de-Wit, and J.W. Grizzle RABBIT: A Testbed for Advanced Control Theory, IEEE Control Systems Magazine, Vol. 23, No. 5, October, 2003, pp. 57-79
- C. Chevallereau, E.R. Westervelt, and J.W. Grizzle, Asymptotically Stable Running for a Five-Link, Four-Actuator, Planar Bipedal Robot, International Journal of Robotics Research, Volume 24, Issue 6, June 2005, pp. 431 - 464.
- Sreenath, K., Park, H.-W., Poulakakis, I., & Grizzle, J. W. (2011). A Compliant Hybrid Zero Dynamics Controller for Stable, Efficient and Fast Bipedal Walking on MABEL. *The International Journal of Robotics Research*, *30*(9), 1170–1193. https://doi.org/10.1177/0278364910379882
- Westervelt, E. R., Grizzle, J. W., Chevallereau, C., Choi, J. H., & Morris, B. (2007). Feedback Control of Dynamic Bipedal Robot Locomotion. CRC Press. https://doi.org/10.1201/9781420053739

Examples of model-based approaches

Model-based control:

1. Trajectory based methods (ZMP)
2. Virtual leg control (Raibert)
3. Virtual model control (Pratt et al)
4. Hybrid Zero Dynamics control
5. **Planning methods (Little dog project)**
6. Inverse dynamics and model predictive control (MPC)

Planning methods

- DARPA’s Little Dog project

- Main idea: control locomotion on very rough terrain by **providing very accurate 3D information about the ground** and the **robot absolute position and orientation**

- Competition with 5 US teams

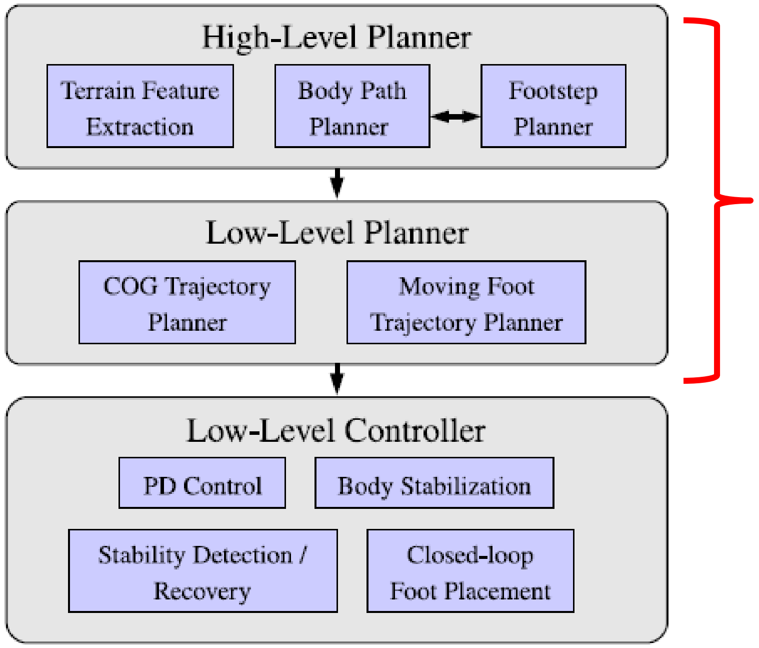
- Most teams highly depend on **planning methods**

- Several use learning, e.g. for foot placement

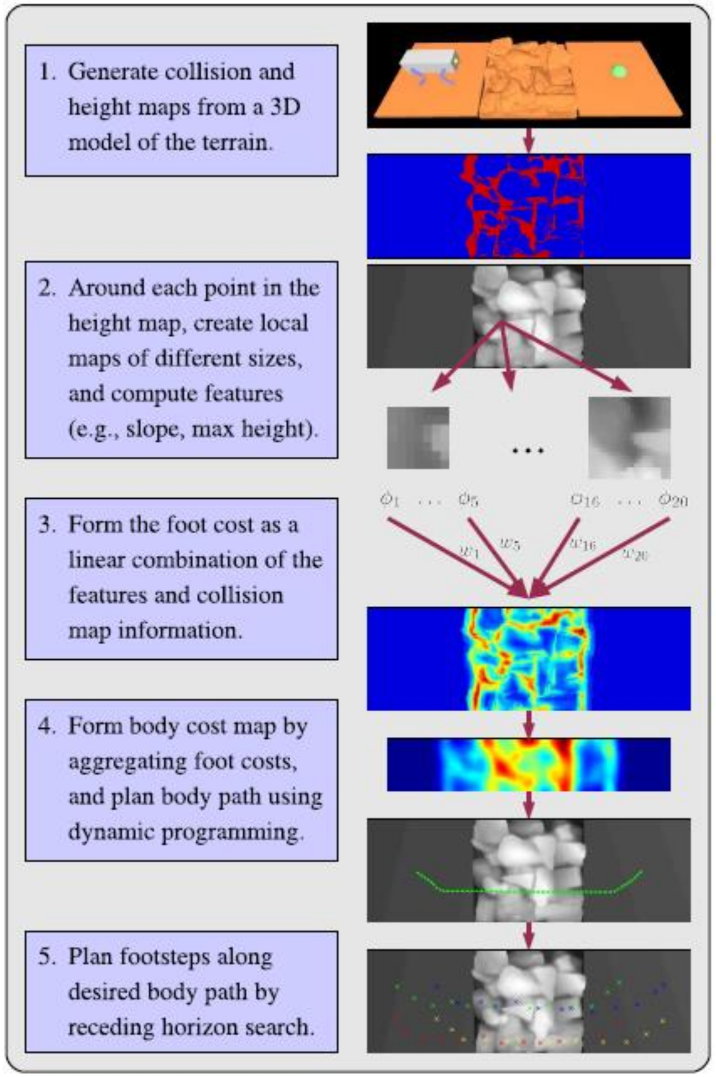
Buchli et al 2009

Planning methods

- Example: Stanford's team (Ng and colleagues)



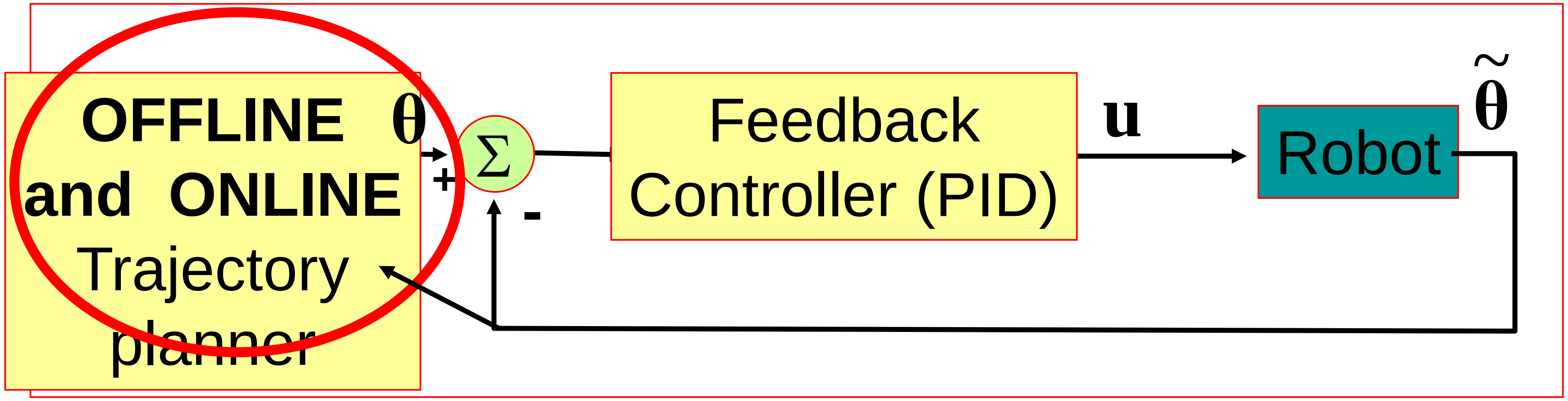
Kolter et al 2008



1. Generate collision and height maps from a 3D model of the terrain.
2. Around each point in the height map, create local maps of different sizes, and compute features (e.g., slope, max height).
3. Form the foot cost as a linear combination of the features and collision map information.
4. Form body cost map by aggregating foot costs, and plan body path using dynamic programming.
5. Plan footsteps along desired body path by receding horizon search.

Different types of planning done thanks to good models of the environment and the robot

Minimalistic control diagram



OFFLINE and ONLINE Trajectory planner
θ
+
-
Σ
Feedback Controller (PID)
u
Robot
$\tilde{\theta}$

θ Desired robot posture

$\tilde{\theta}$ Actual robot posture

$\mathbf{u}$ Command (torque)

Planning methods

- Example: USC’s team (Schaal, Buchli and colleagues)

Learning Locomotion with LittleDog

http://www-clmc.usc.edu

Mrinal Kalakrishnan, Jonas Buchli, Peter Pastor, and Stefan Schaal

Planning methods: summary

- Pros:
  - Ability to handle very complex terrain that requires careful foot holds.
- Cons:
  - Requires very accurate 3D maps of the ground.
  - It is not clear how performance degrades with less good sensory input
  - Not well suited for biped locomotion (except slow statically stable locomotion)

References:

Buchli, J.;Kalakrishnan, M.;Mistry, M.;Pastor, P.;Schaal, S. (2009). Compliant quadruped locomotion over rough terrain, Proceedings of IROS 2009, pp.814-820.

Kalakrishnan, M.;Buchli, J.;Pastor, P.;Schaal, S. (2009). Learning locomotion over rough terrain using terrain templates, Proceedings of IROS 2009 pp.167-172.

J. Zico Kolter, Mike P. Rodgers, and Andrew Y. Ng. A Control Architecture for Quadruped Locomotion over Rough Terrain. In Proceedings of ICRA2008, 2008.

Examples of model-based approaches

Model-based control:

1. Trajectory based methods (ZMP)
2. Virtual leg control (Raibert)
3. Virtual model control (Pratt et al)
4. Hybrid Zero Dynamics control
5. Planning methods (Little dog project)
6. **Inverse dynamics and model predictive control (MPC)**

Inverse dynamics and model predictive control

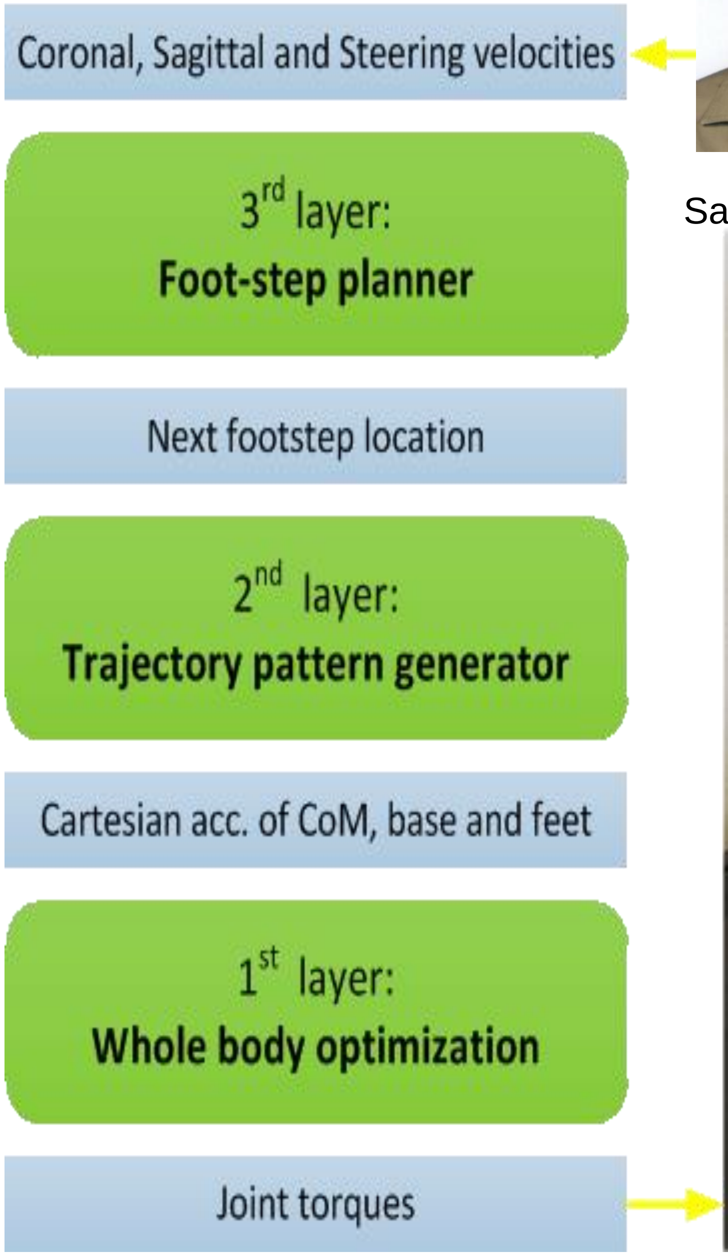
- For **torque-controlled** robots for which an **accurate dynamical model** exists:

- Possibility:
  - To **compute the inverse-dynamics of the robot**, i.e. finding the torques needed to perform specific movements
  - To **run optimizations** to find torques that optimize some objective functions and that respect some constraints.
  - And perform foot step planning using **model predictive control**

  - And therefore to obtain highly versatile gaits and whole body control.

  - Probably used on the latest videos of Boston Dynamics

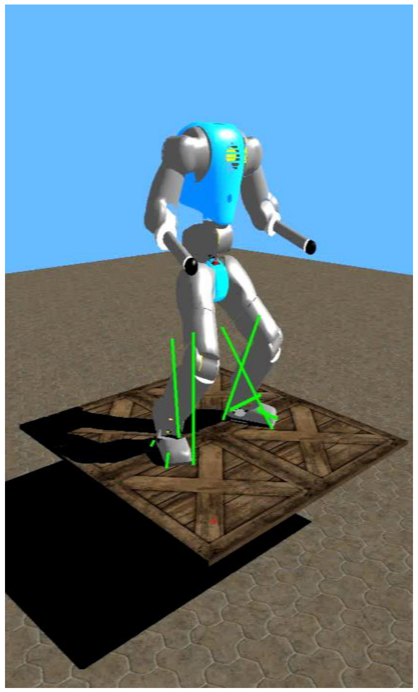
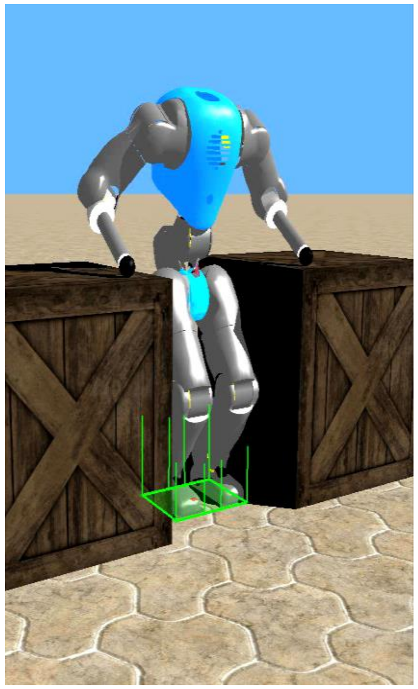
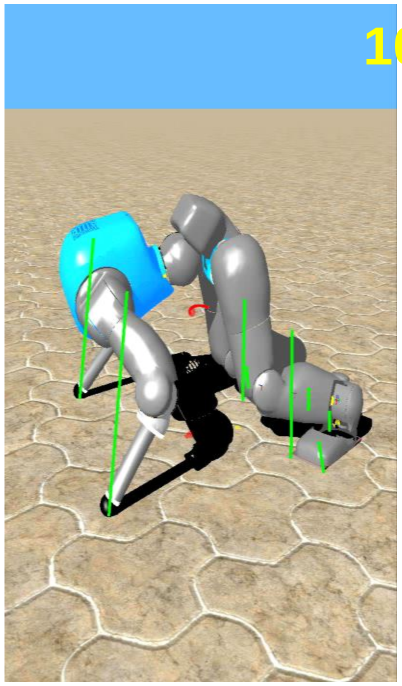
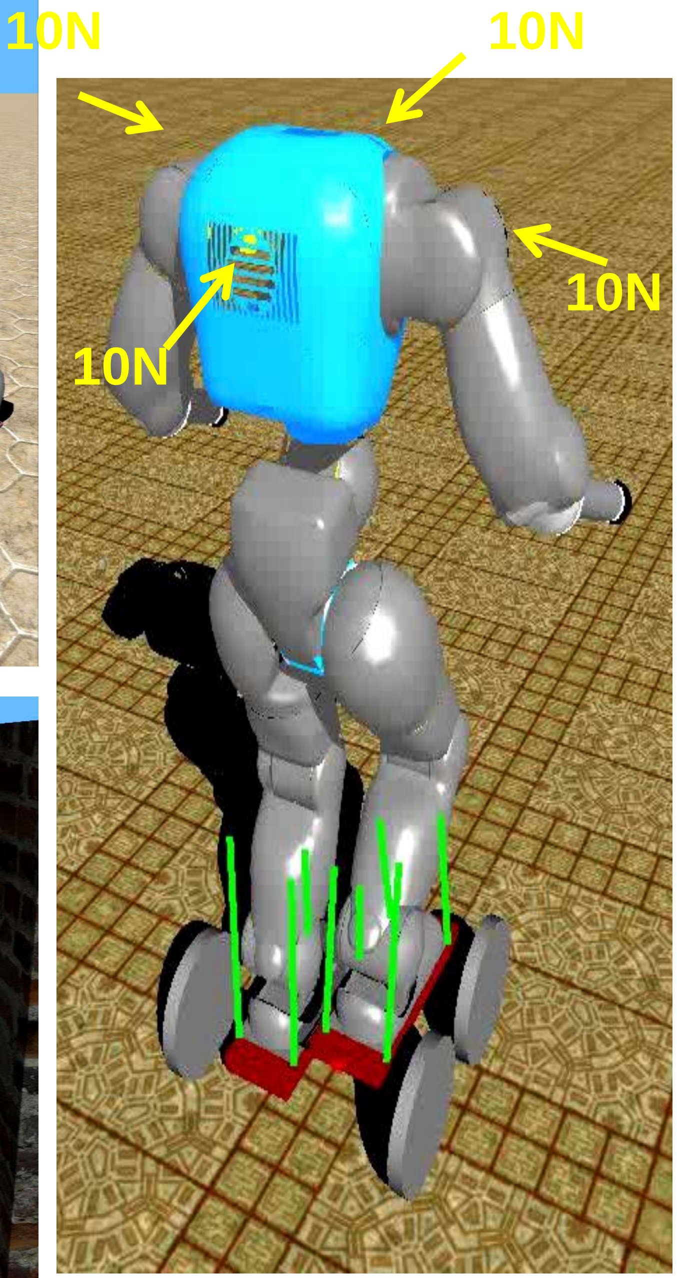
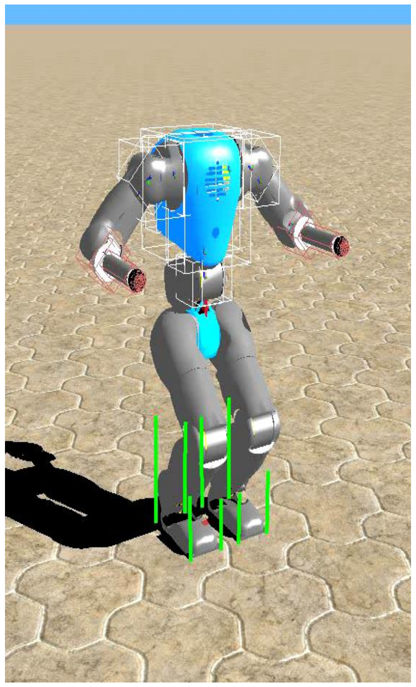
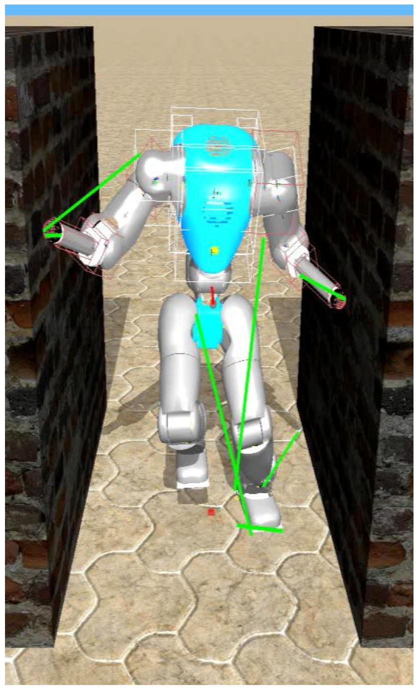
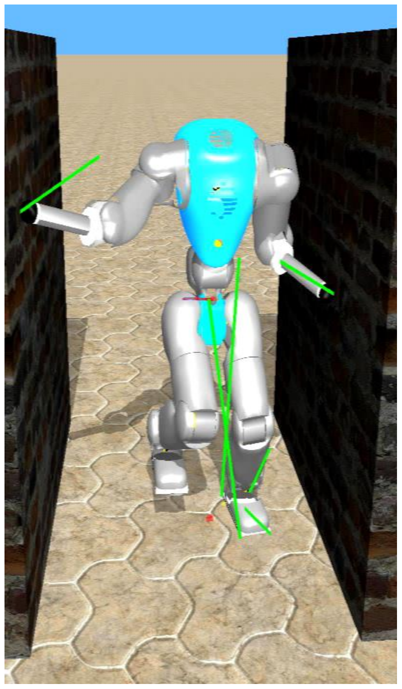
Optimization of **footsteps**, **Model predictive control** Linear Inverted Pendulum model


Coronal, Sagittal and Steering velocities
3rd layer:
Foot-step planner
Next footstep location
2nd layer:
Trajectory pattern generator
Cartesian acc. of CoM, base and feet
1st layer:
Whole body optimization
Joint torques

Salman Faraji

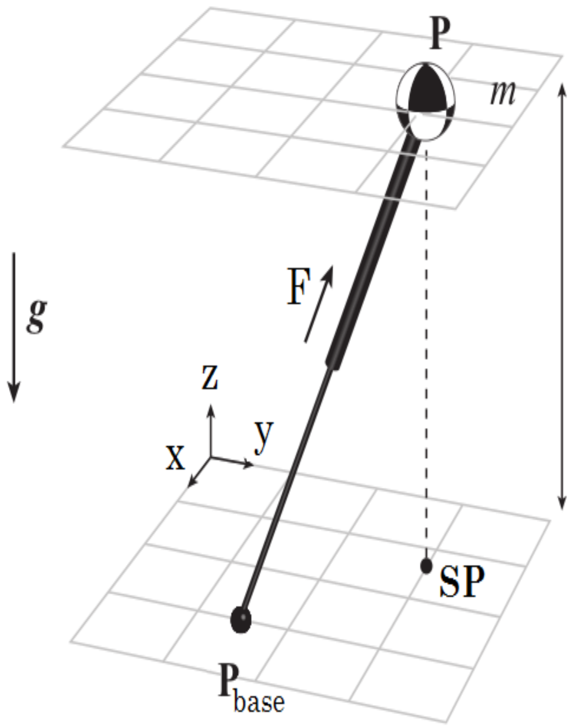
Inverse dynamics, Online optimization of **torques** for all DOFs

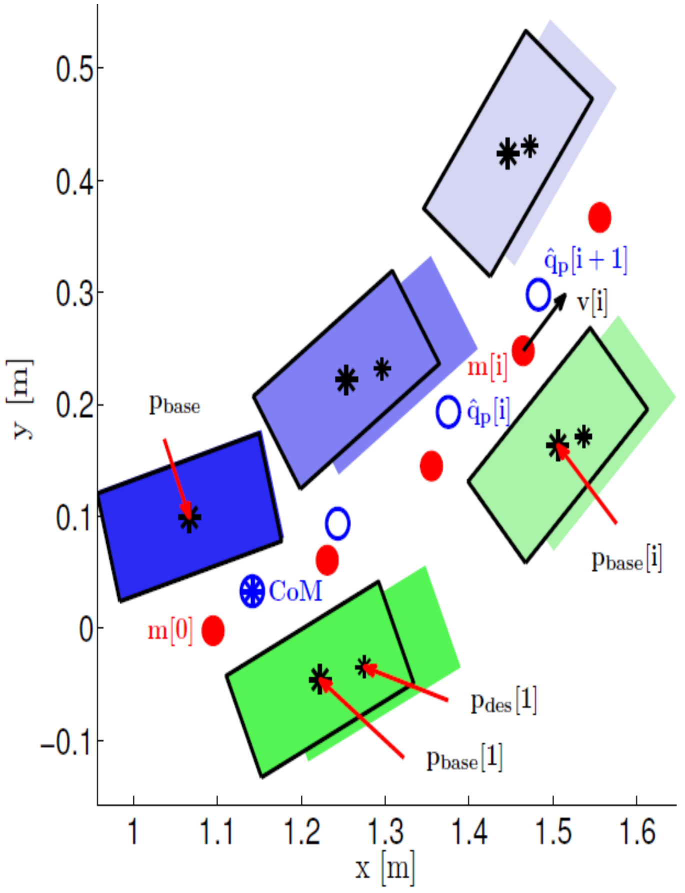
Faraji, S., Pouya, S., & Ijspeert, A. (RSS 2014). *Robust and Agile 3D Biped Walking With Steering Capability Using a Footstep Predictive Approach*. https://doi.org/10.15607/RSS.2014.X.028

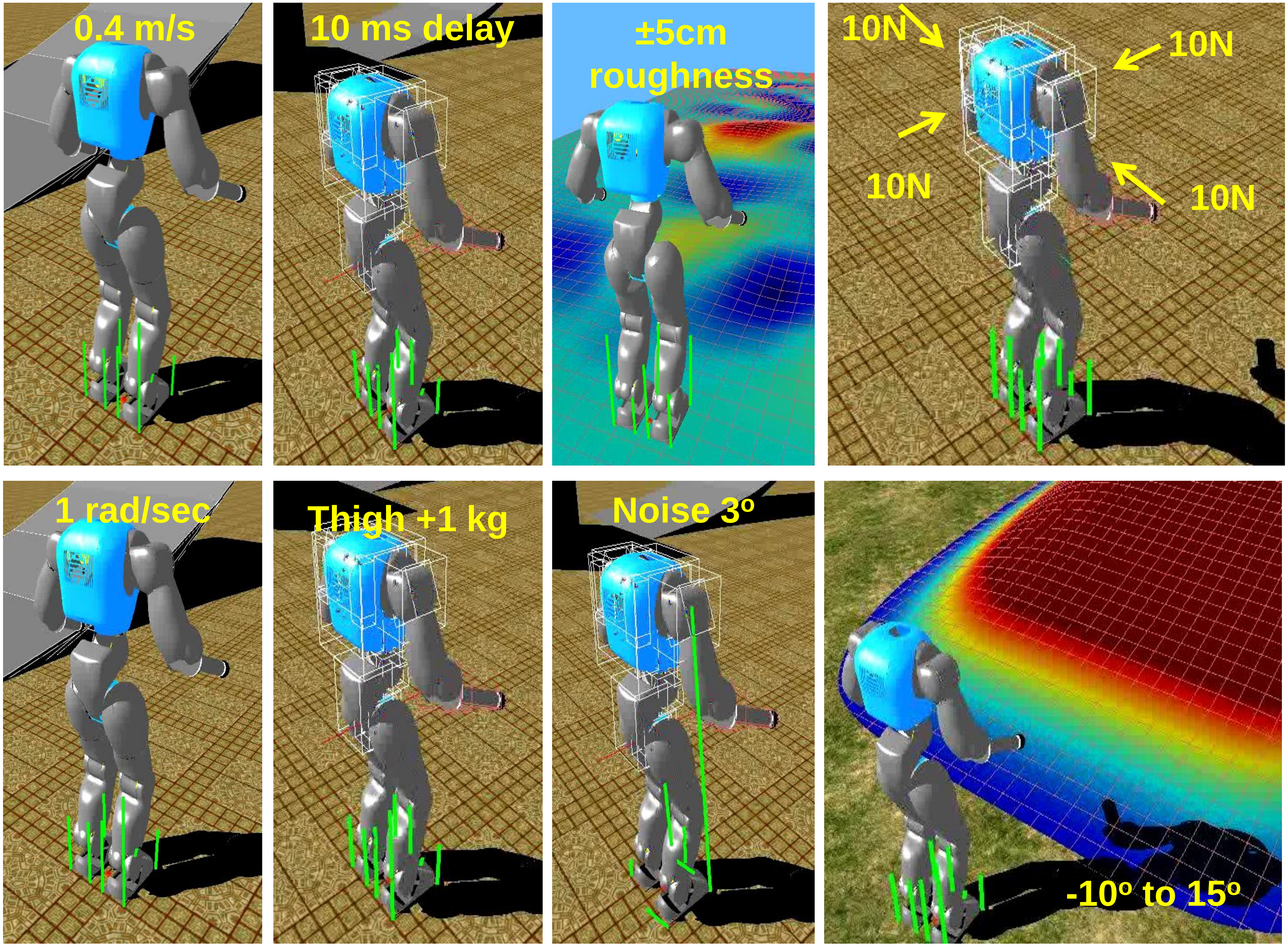




10N
10N
10N
10N




Linear Inverted Pendulum and MPC for step planning

$$\ddot{x} = \frac{g}{z_0}(x - x_{base})$$






0.4 m/s
10 ms delay
±5cm roughness
10N
10N
10N
10N
1 rad/sec
Thigh +1 kg
Noise 3^o
-10^o to 15^o

Inverse dynamics and model predictive control

- Pros:
  - Ability to generate a large class of movements: walking + many others
  - Allows one to design controllers in task space, as opposed to joint space

See more about optimal control and MPC in guest lectures in a few weeks

- Cons:
  - Requires accurate dynamic models
  - Requires (very) good torque control
  - Heavy computation, but many approaches can now do this online

References:

Herzog, A., Rotella, N., Mason, S., Grimminger, F., Schaal, S., & Righetti, L. (2016). Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. *Autonomous Robots*, *40*(3), 473–491. https://doi.org/10.1007/s10514-015-9476-6

Salman Faraji, Soha Pouya, Christopher G. Atkeson, and Auke Jan Ijspeert (2014) Versatile and Robust 3D Walking with the Humanoid Robot Atlas: a Model Predictive Control Approach. ICRA 2014.

Neunert, M., Stäuble, M., Giftthaler, M., Bellicoso, C. D., Carius, J., Gehring, C., Hutter, M., & Buchli, J. (2018). Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds. *IEEE Robotics and Automation Letters*, *3*(3), 1458–1465. https://doi.org/10.1109/LRA.2018.2800124

Di Carlo, J., Wensing, P. M., Katz, B., Bledt, G., & Kim, S. (2018). Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–9.

Learning-based methods

- Learning-based methods are recently becoming popular thanks to progress in machine learning, increased computation power, and faster simulators.

- Most people use **Reinforcement learning (RL)**, which can be seen as a generalization of optimal control.
- Different approaches exist: e.g. hierarchical reinforcement learning (Peng et al., 2017) or direct end-to-end learning (Jain et al., 2019).

- Some perform learning directly on the hardware platform in the real world (Choi & Kim, 2019; Ha et al., 2020, 2018), but **most approaches rely on simulation** and then tackle the **sim-to-real transfer** challenge (Hwangbo et al., 2019; Tan et al., 2018, Peng et al 2020).

Learning with ANYmal

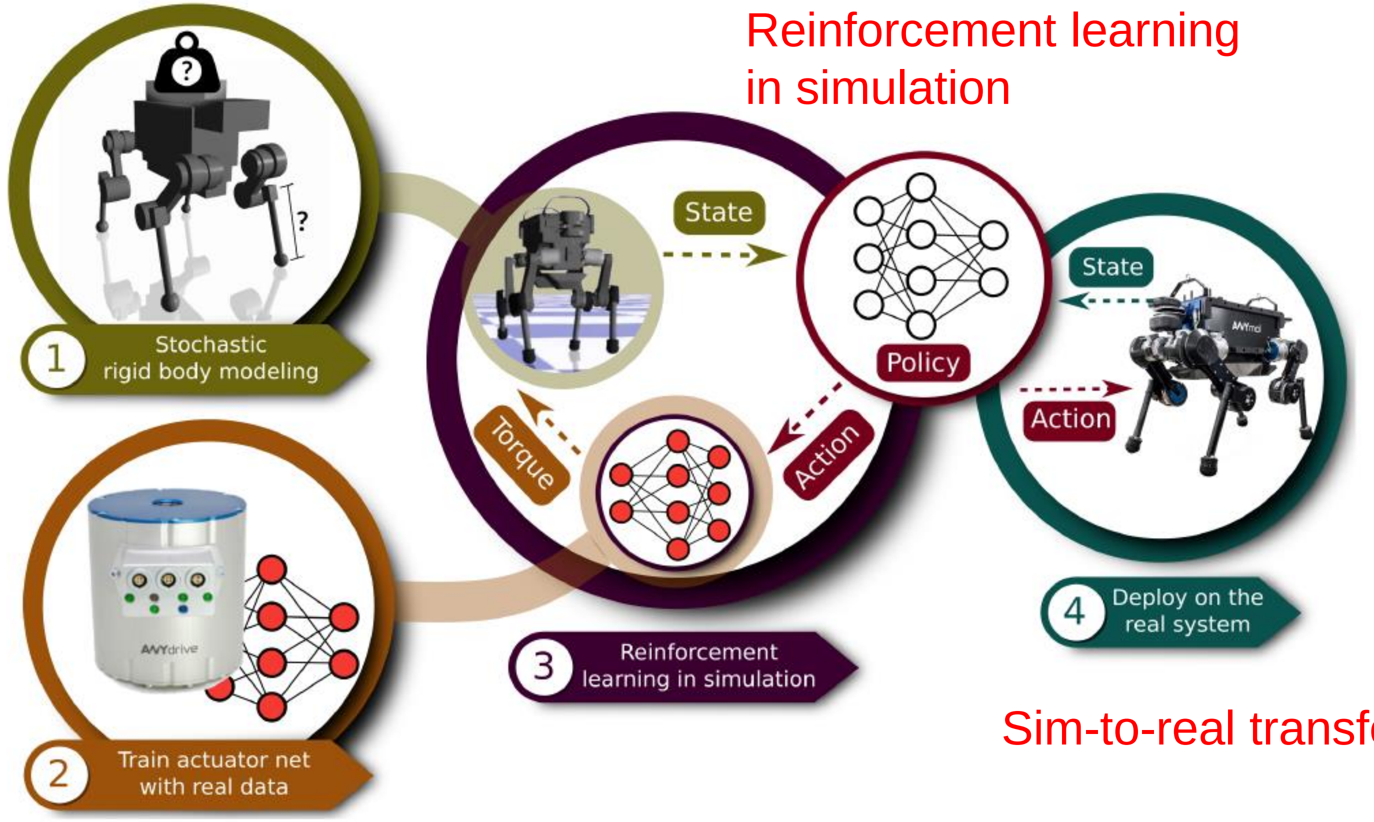
- Example: (Hwangbo et al., 2019) nice combination of reinforcement learning and supervised learning.

Good simulator

Reinforcement learning in simulation

Supervised learning of actuator dynamics

Sim-to-real transfer

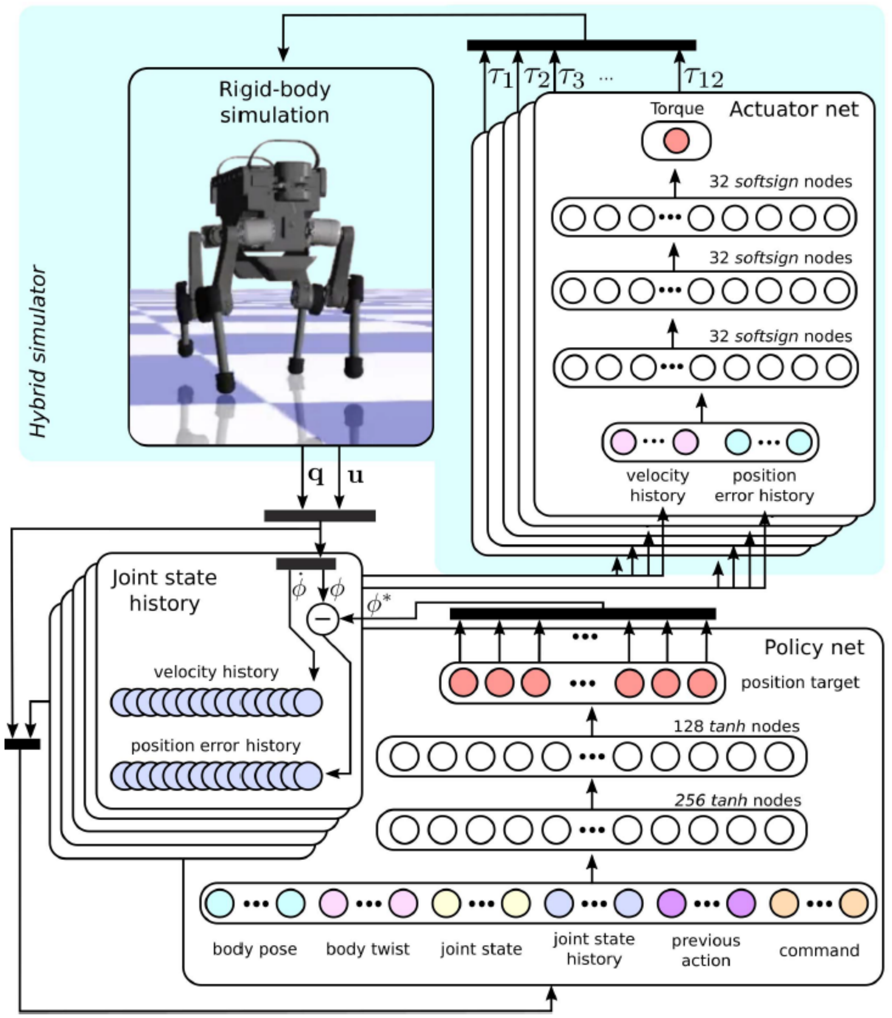


1 Stochastic rigid body modeling
2 Train actuator net with real data
State
Policy
Torque
Action
3 Reinforcement learning in simulation
State
Action
4 Deploy on the real system

Fig. 1. Creating a control policy. In the first step, we identify the physical parameters of the robot and estimate uncertainties in the identification. In the second step, we train an actuator net that models complex actuator/software dynamics. In the third step, we train a control policy using the models produced in the first two steps. In the fourth step, we deploy the trained policy directly on the physical system.

Hwangbo, J., Lee, J., Dosovitskiy, A., Bellicoso, D., Tsounis, V., Koltun, V., & Hutter, M. (2019). Learning agile and dynamic motor skills for legged robots. *Science Robotics*, *4*(26), eaau5872. https://doi.org/10.1126/scirobotics.aau5872 https://www.science.org/doi/10.1126/scirobotics.aau5872

Learning with ANYmal

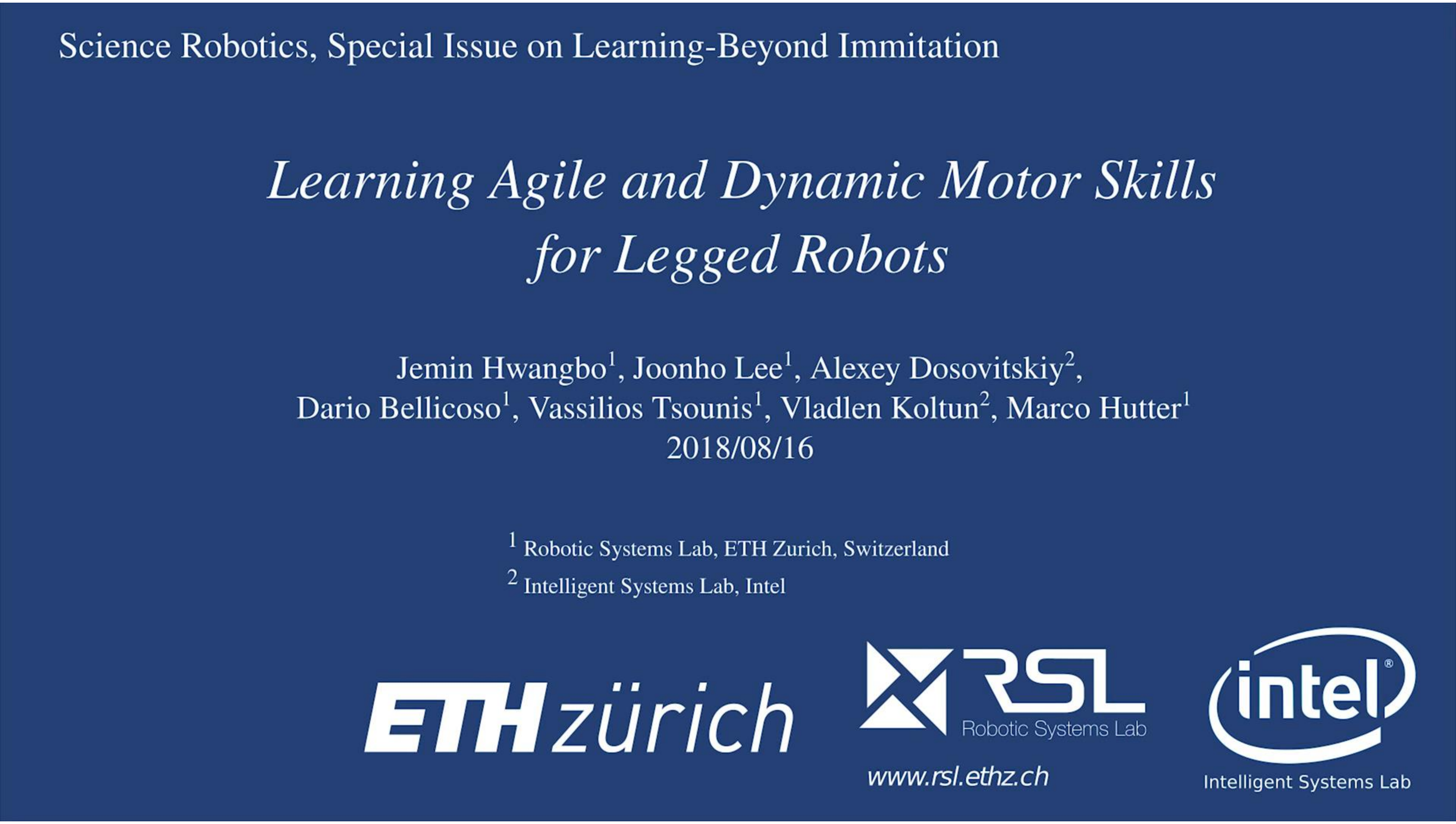


ig. 5. **Training control policies in simulation.** The policy network maps the current observation and the joint state istory to the joint position targets. The actuator network maps the joint state history to the joint torque, which is used n rigid-body simulation. The state of the robot consists of the generalized coordinate *q* and the generalized velocity *u*. The tate of a joint consists of the joint velocity $\dot{\phi}$ and the joint position error, which is the current position ϕ subtracted from he joint position target ϕ^*.

See Movie 1: https://www.science.org/doi/10.1126/scirobotics.aau5872

See the guest lecture of Marco Hutter in a few weeks

Learning with ANYmal



Science Robotics, Special Issue on Learning-Beyond Immitation

Learning Agile and Dynamic Motor Skills for Legged Robots

Jemin Hwangbo[1], Joonho Lee[1], Alexey Dosovitskiy[2], Dario Bellicoso[1], Vassilios Tsounis[1], Vladlen Koltun[2], Marco Hutter[1]

2018/08/16

[1] Robotic Systems Lab, ETH Zurich, Switzerland

[2] Intelligent Systems Lab, Intel

ETH zürich
RSL
Robotic Systems Lab
www.rsl.ethz.ch
intel
Intelligent Systems Lab

https://www.youtube.com/watch?v=aTDkYFZFWug

Learning-based methods

Pros:

- They offer **generic design methods** that require less expertise than model-based approaches.
- Their performance **can beat human-designed model-based controllers**.
- They can combine supervised and unsupervised learning.
- They can **generate new movements** that would be difficult/impossible to hand-design.

See the practical and Guillaume Bellegarda's presentation in a few weeks

Cons:

- need for **very long training sequences**
- a **strong reliance on simulation and sim-to-real transfer**
- need of expertise in designing cost functions and training scenarios,
- Black-box controller, **lack of proof of stability/performance**
- possibly poor generalization to handle situations (e.g. terrains) that have not been explored during training.

Learning-based methods

References:

Choi, S., & Kim, J. (2019). Trajectory-based Probabilistic Policy Gradient for Learning Locomotion Behaviors. *2019 International Conference on Robotics and Automation (ICRA)*, 1–7. https://doi.org/10.1109/ICRA.2019.8794207

Ha, S., Kim, J., & Yamane, K. (2018). Automated Deep Reinforcement Learning Environment for Hardware of a Modular Legged Robot. *2018 15th International Conference on Ubiquitous Robots (UR)*, 348–354. https://doi.org/10.1109/URAI.2018.8442201

Ha, S., Xu, P., Tan, Z., Levine, S., & Tan, J. (2020). Learning to Walk in the Real World with Minimal Human Effort. *ArXiv:2002.08550 [Cs]*. http://arxiv.org/abs/2002.08550

Hwangbo, J., Lee, J., Dosovitskiy, A., Bellicoso, D., Tsounis, V., Koltun, V., & Hutter, M. (2019). Learning agile and dynamic motor skills for legged robots. *Science Robotics*, *4*(26), eaau5872. https://doi.org/10.1126/scirobotics.aau5872

Jain, D., Iscen, A., & Caluwaerts, K. (2019). Hierarchical Reinforcement Learning for Quadruped Locomotion. *ArXiv:1905.08926 [Cs]*. http://arxiv.org/abs/1905.08926

Peng, X. B., Berseth, G., Yin, K., & Van De Panne, M. (2017). DeepLoco: Dynamic Locomotion Skills Using Hierarchical Deep Reinforcement Learning. *ACM Trans. Graph.*, *36*(4), 41:1–41:13. https://doi.org/10.1145/3072959.3073602

Peng, X. B., Coumans, E., Zhang, T., Lee, T.-W., Tan, J., & Levine, S. (2020). Learning Agile Robotic Locomotion Skills by Imitating Animals. RSS 2020. *ArXiv:2004.00784 [Cs]*. http://arxiv.org/abs/2004.00784

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. *International Conference on Machine Learning*, 1889–1897.

Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., & Hutter, M. (2020). Learning quadrupedal locomotion over challenging terrain. Science Robotics, 5(47). https://doi.org/10.1126/scirobotics.abc5986

Siekmann, J., Godse, Y., Fern, A., & Hurst, J. (2021). Sim-to-Real Learning of All Common Bipedal Gaits via Periodic Reward Composition. ArXiv:2011.01387 [Cs]. http://arxiv.org/abs/2011.01387

Bio-inspired approaches

- Replicate some of the control principles identified in vertebrate and invertebrate animals
- They are strongly influenced by the idea of **embodied intelligence** proposed by Rolf Pfeifer and colleagues, and Rodney Brooks' observation that "the world is its own best model".
- Bioinspired locomotion controllers typically combine numerical models of **central pattern generators** implemented as recurrent neural networks and coupled nonlinear oscillators and **reflexes,** implemented as feedback laws.
- Some (sensory-driven) approaches only rely on reflexes.
- And a few approaches use reinforcement learning for setting control parameters.

R. Pfeifer and J. Bongard, *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.

R. Pfeifer, M. Lungarella, and F. Iida, "Self-Organization, Embodiment, and Biologically Inspired Robotics," *Science*, vol. 318, no. 5853, pp. 1088–1093, Nov. 2007, doi: 10.1126/science.1145803.

R. A. Brooks, "How to Build Complete Creatures Rather than Isolated Cognitive Simulators," *Architectures for Intelligence*, Jan. 14, 2014. https://www.taylorfrancis.com/ (accessed Apr. 07, 2020).

Examples of bio-inspired approaches

1. **Passive and dynamic walkers**

2. Sensory-driven methods,

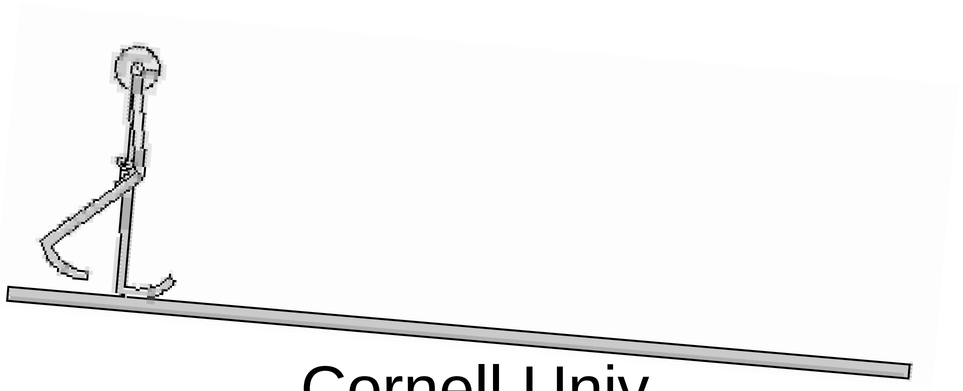
3. CPG-and-reflex based methods

Robotics: passive walkers

- The laws of physics should be exploited: passive walkers

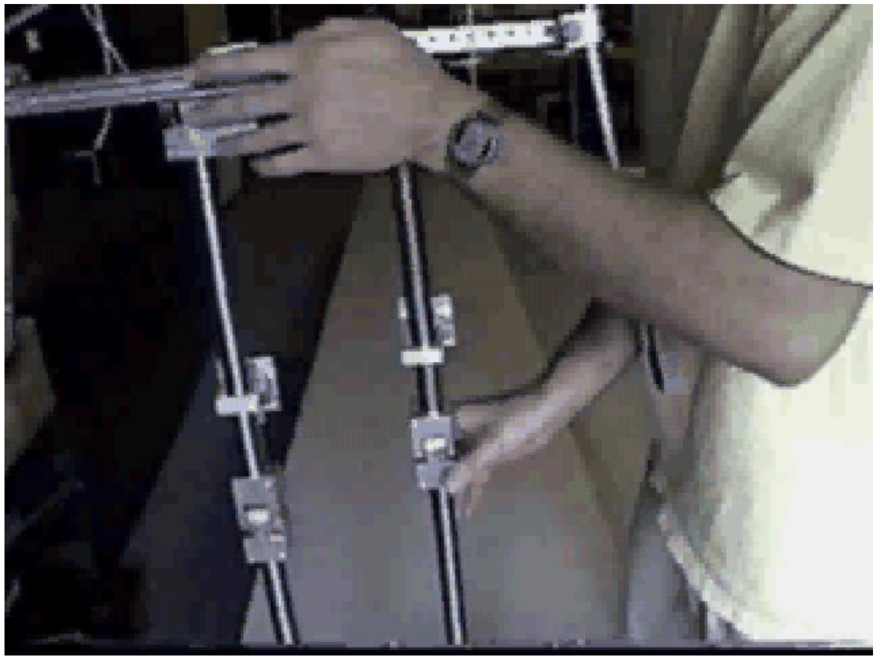
Movie by Jun Nakanishi

Nice example of <u>mechanical</u> limit cycle behavior



Cornell Univ.

Passive walkers (ct'd)



A Three-Dimensional Passive-Dynamic Walking Robot with Two Legs and Knees, Collins, S. H., Wisse, M., Ruina, A. International Journal of Robotics Research, Vol. 20, No. 2, Pages 607-615, 2001

Passive and dynamic walkers

- The **laws of physics can be exploited** to produce relatively **robust control-less walking**
- Instead of cancelling-out the natural dynamics of the robot (by using high-power electric motors), takes advantage of the natural frequencies of the robot

- **Self-stabilizing phenomenon**
- **Dynamic walkers** are passive walkers + actuation
- **Require little energy** when actuated E.g. robot Mike at Delft Univ. with McKnibben muscles

Robot Mike

- See the Cornel Ranger that can walk 65 km on one battery. Bhounsule et al (IJRR_2014)

Examples of bio-inspired approaches

1. Passive and dynamic walkers
2. **Sensory-driven methods,**
3. CPG-and-reflex based methods

Runbot project

Exploitation of natural dynamics

Sensor driven controller implemented with a neural network, **locomotion as a chain of reflexes**

Policy gradient reinforcement learning algorithm to tune the parameters in real time.

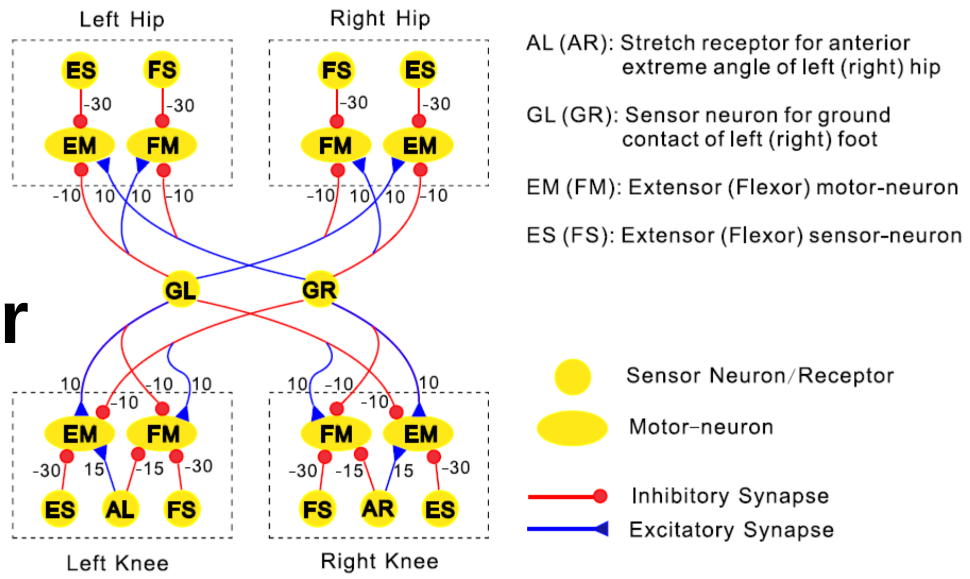
Note: this sensory-driven controller **share similarities with neuromechanical models of human locomotion** (e.g. Geyer and Herr 2010).

Reference: Gen, Porr, and Wörgötter, Fast Biped Walking with a Sensor-driven Neuronal Controller and Real-time Online Learning, The International Journal of Robotics Research, Vol. 25, No. 3, 243-259, 2006.

Runbot project

Exploitation of natural dynamics

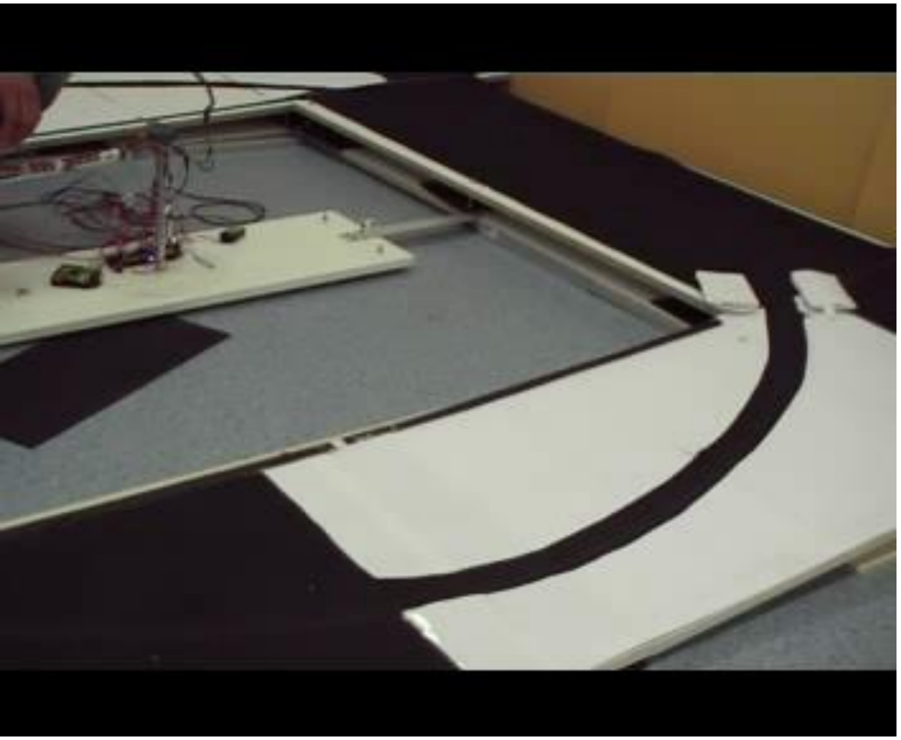
Sensor driven controller (like Geyer's model) implemented with a neural network

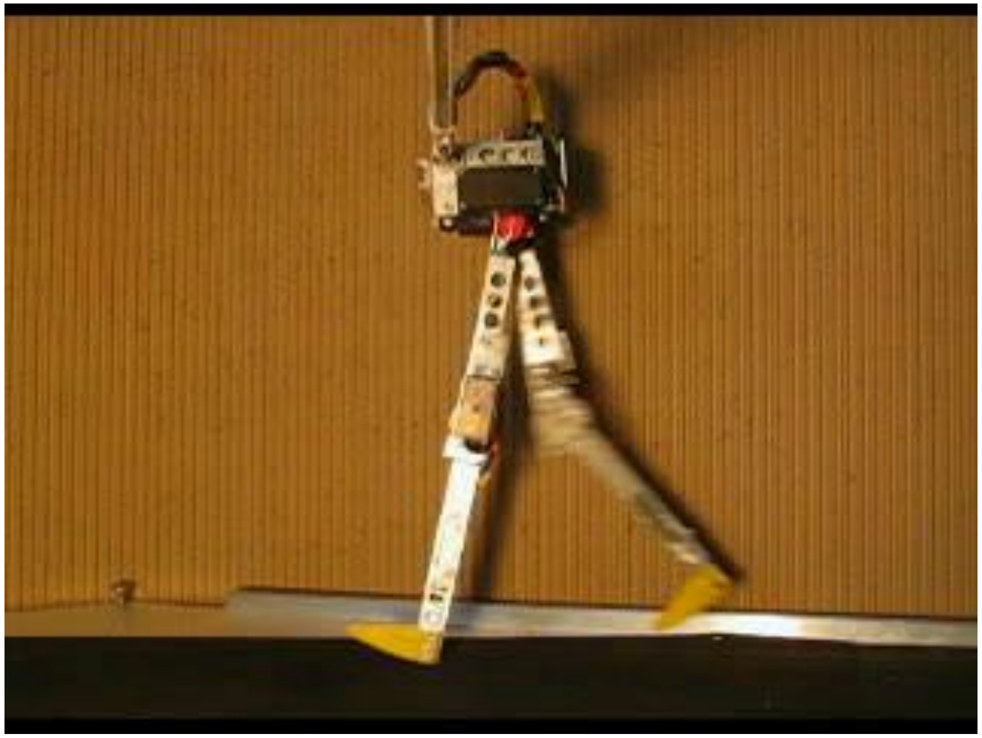

Left Hip
Right Hip
ES
FS
EM
FM
GL
GR
AL
AR
-30
-10
10
-15
15
Left Knee
Right Knee
AL (AR): Stretch receptor for anterior extreme angle of left (right) hip
GL (GR): Sensor neuron for ground contact of left (right) foot
EM (FM): Extensor (Flexor) motor-neuron
ES (FS): Extensor (Flexor) sensor-neuron
Sensor Neuron/Receptor
Motor-neuron
Inhibitory Synapse
Excitatory Synapse

Policy gradient **reinforcement learning algorithm** to tune the parameters in real time

Gen, Porr, and Wörgötter, Fast Biped Walking with a Sensor-driven Neuronal Controller and Real-time Online Learning, The International Journal of Robotics Research, Vol. 25, No. 3, 243-259, 2006.

Runbot project





Sensory-driven control: summary

Pros:

- Very close link between the controller and what the robot actual does
- Can be very energy efficient by benefiting from passive dynamics (as opposed to stiff actuation)

Cons:

- because of the lack of a centrally generated rhythm, non-negligible risk that locomotion might be completely stopped because of damage in the sensors and/or external constraints that force the robot in a particular posture.

Examples of bio-inspired approaches

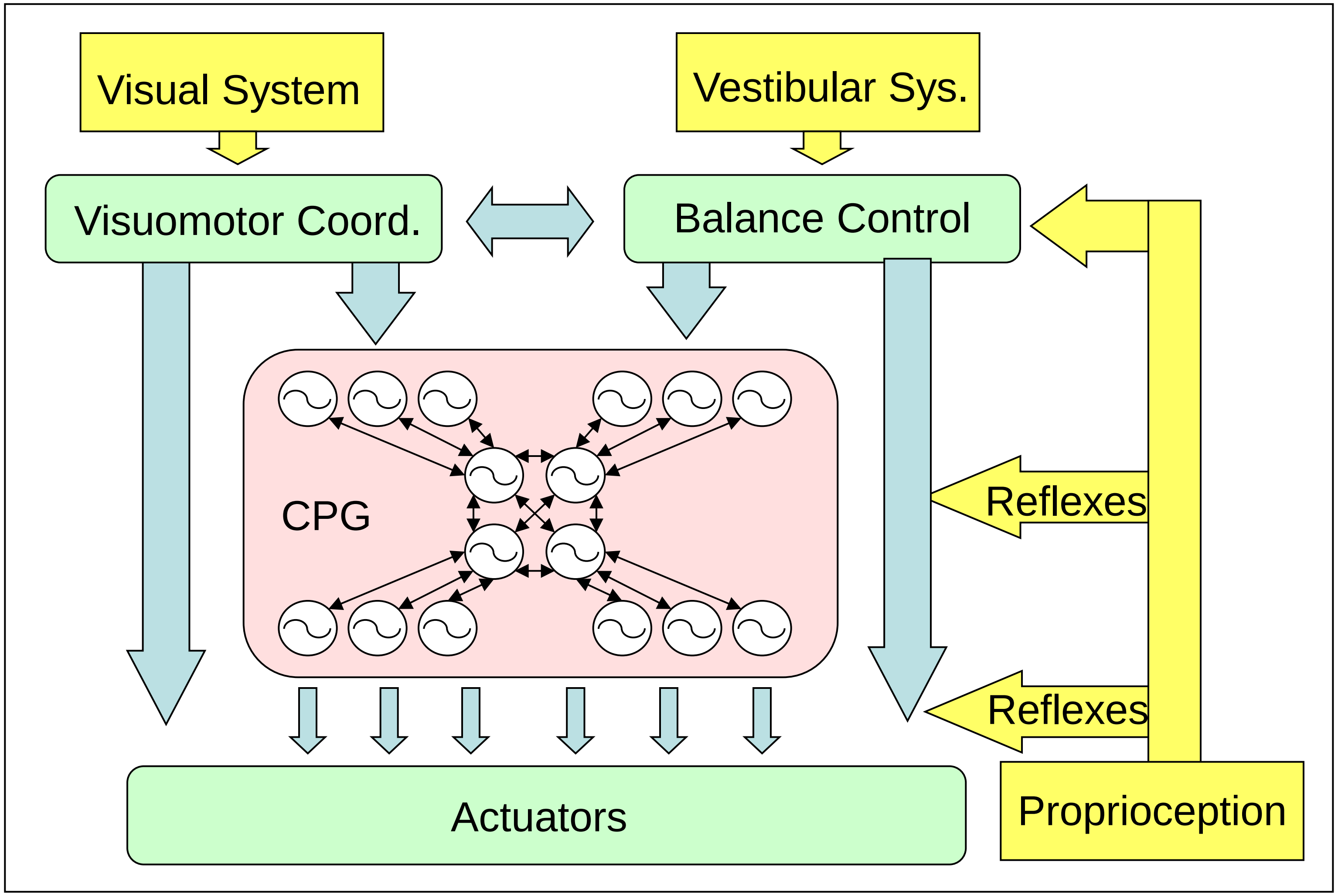
1. Passive and dynamic walkers

2. Sensory-driven methods,

3. **CPG-and-reflex based methods**

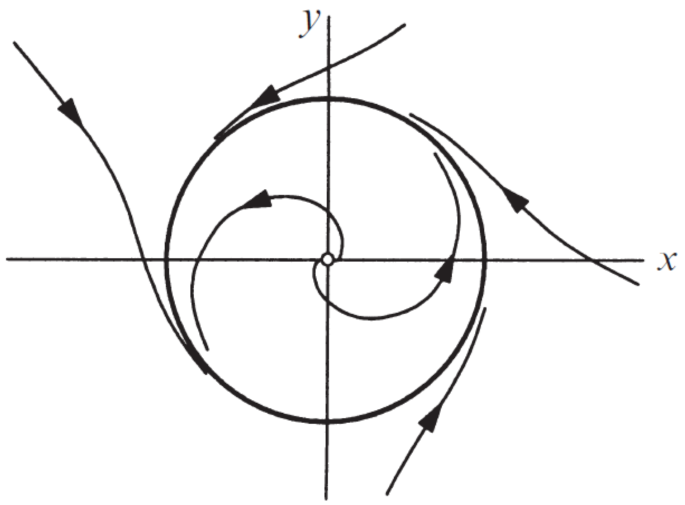
CPG-and-reflex control

- Main idea: to use oscillators and to replicate the distributed control mechanisms found in vertebrates. CPG = Central Pattern Generator



Concept of Limit Cycle

- A *limit cycle* is an oscillatory regime in a dynamical system:



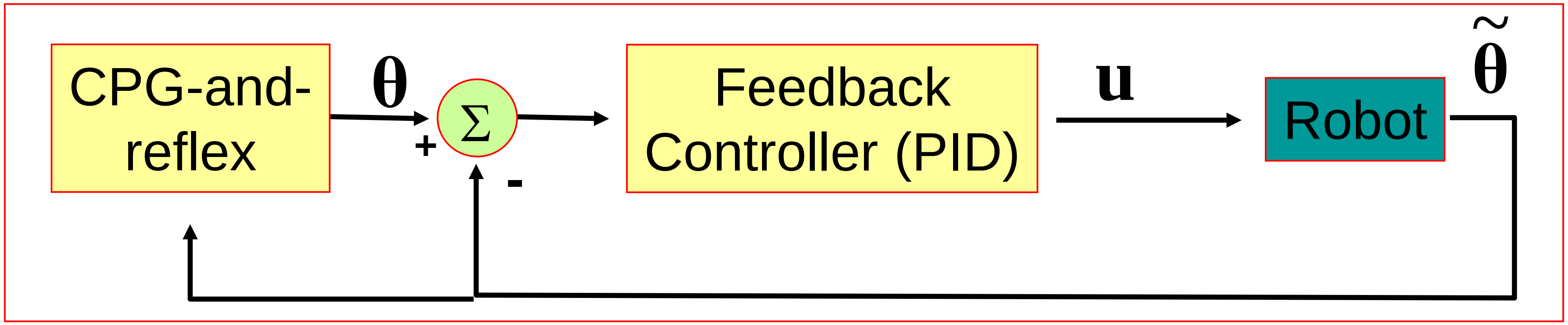
Limit cycles

- If the limit cycle is stable, the states of the system will return to it after perturbations

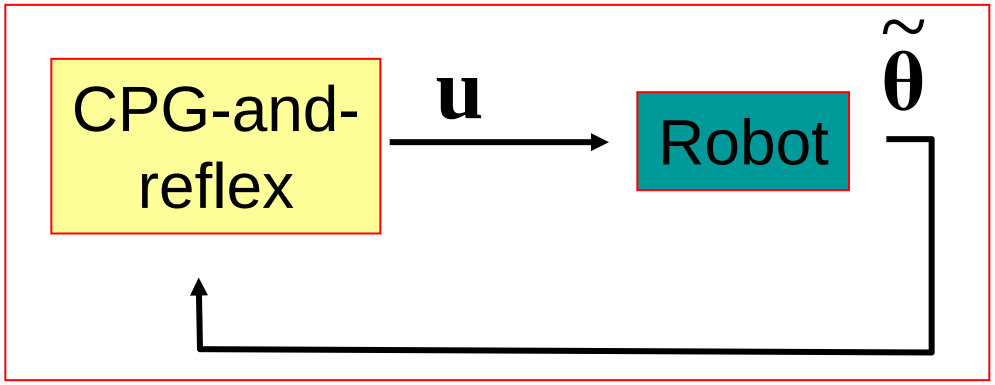
CPG-and-reflex control

Two types of implementations:

CPG produces **desired positions**:



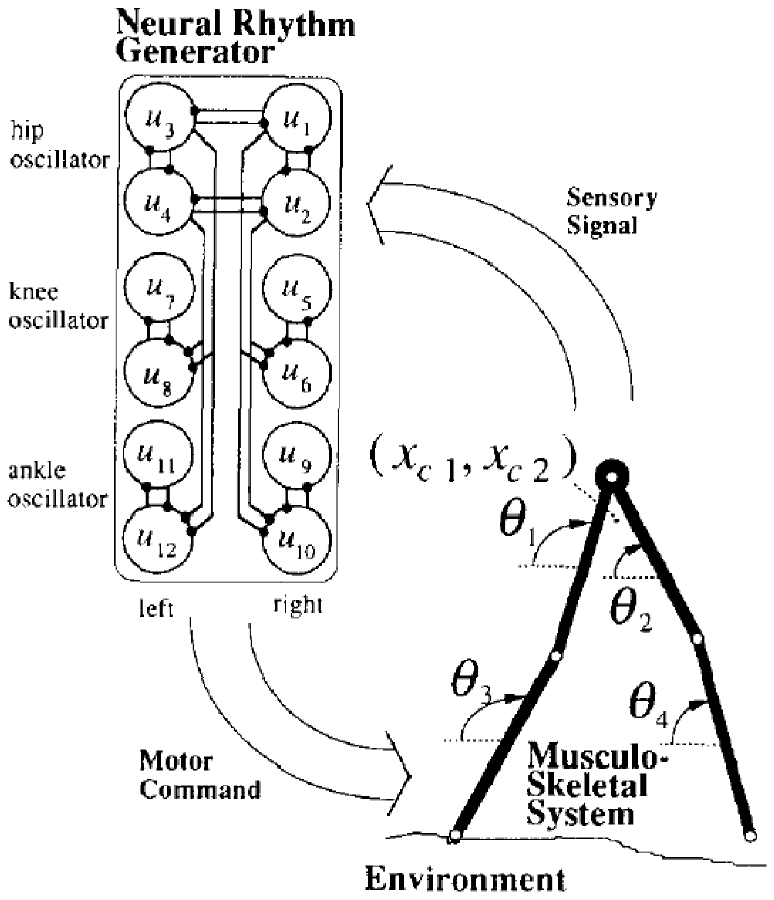
CPG directly produces **torques**:



Taga’s neuromechanical simulation

This approach has been strongly influenced by Taga’s models of biped locomotion.

Quite a few labs have taken a similar approach,

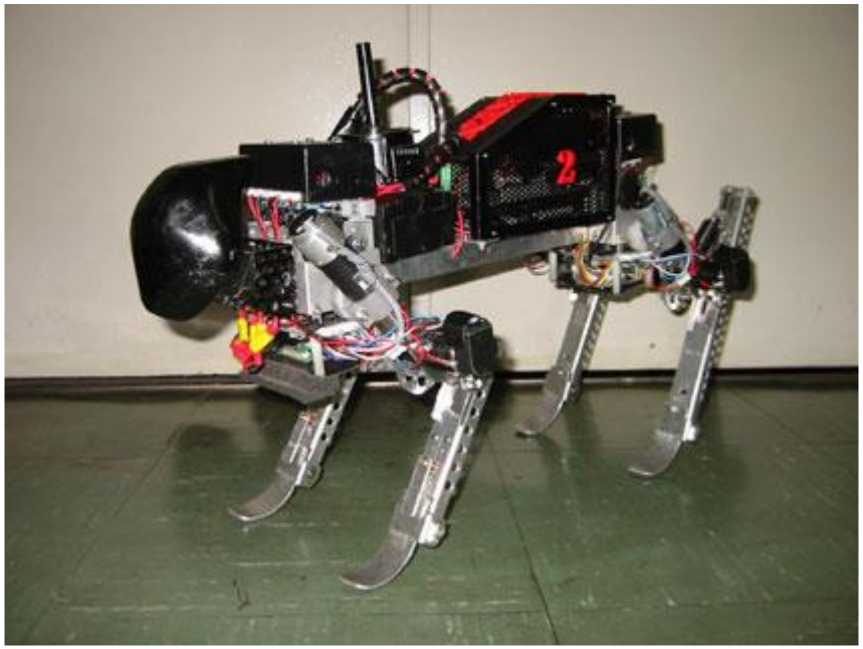


G. Taga. Emergence of bipedal locomotion through entrainment among the neuro-musculo-skeletal system and the environment. *Physica D: Nonlinear Phenomena*, 75(1-3):190-208, 1994

G. Taga. A model of the neuro-musculo-skeletal system for human locomotion. i. emergence of basic gait. *Biological Cybernetics*, 73(2):97-111, 1995

Using CPG models for quadruped robots

Y. Fukuoka, H. Kimura, and A.H. Cohen. Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. The International Journal of Robotics Research, 3-4:187-202, 2003.

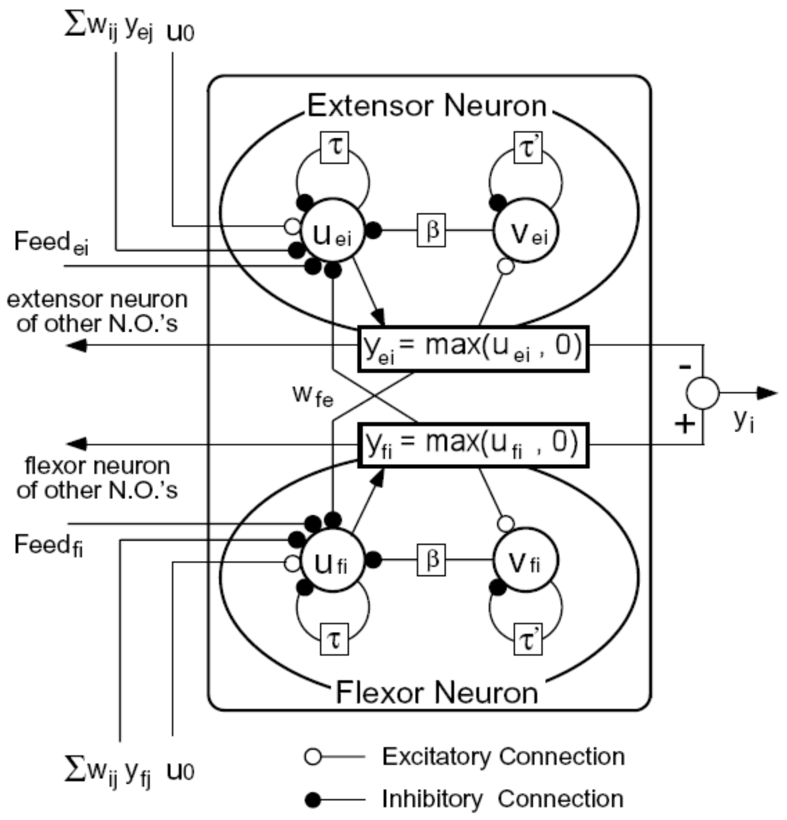


3 actuated DOF per limb

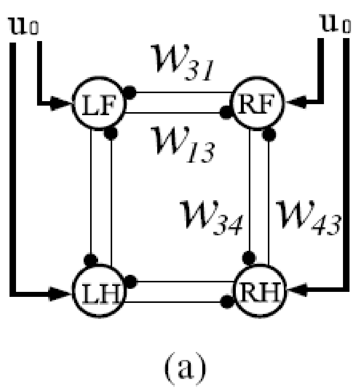
CPG model

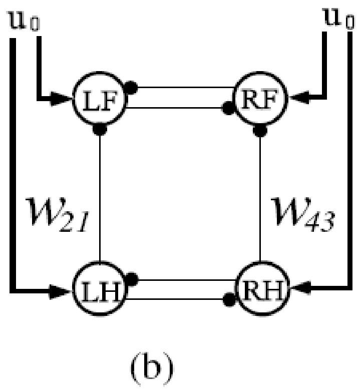
The CPG is made of Matsuoka oscillators: two mutually inhibiting neurons.

$$
\begin{aligned}
\tau \dot{u}_{\{e,f\}i} &= -u_{\{e,f\}i} + w_{fe} y_{\{f,e\}i} - \beta v_{\{e,f\}i} \\
&\quad + u_0 + Feed_{\{e,f\}i} + \sum_{j=1}^{n} w_{ij} y_{\{e,f\}j} \\
y_{\{e,f\}i} &= \max\ (u_{\{e,f\}i}, 0) \\
\tau' \dot{v}_{\{e,f\}i} &= -v_{\{e,f\}i} + y_{\{e,f\}i}.
\end{aligned}
$$

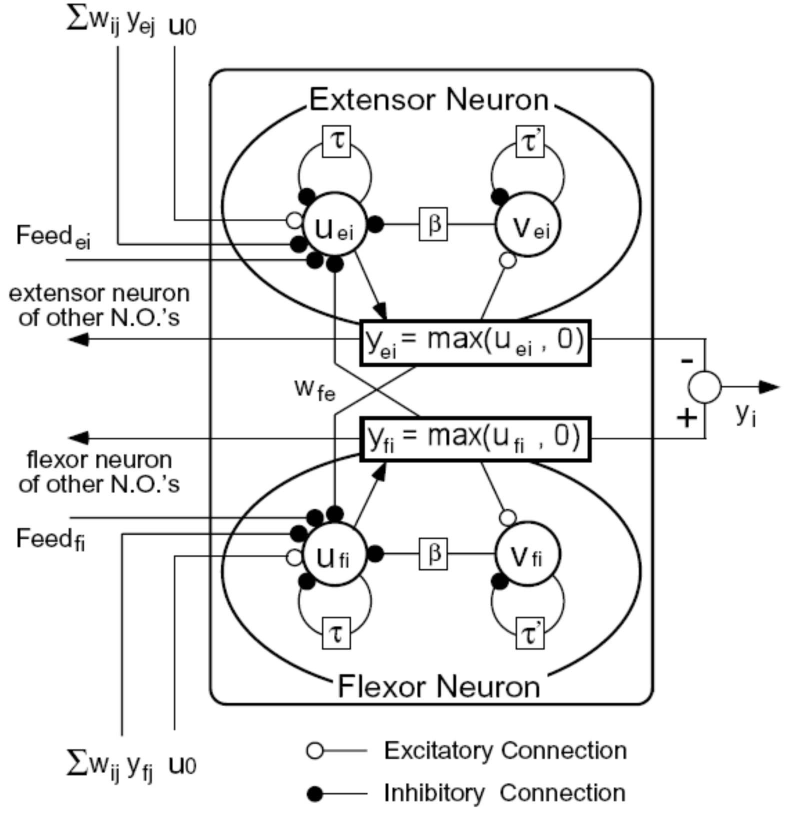


CPG architecture





Gait transitions by changing connection weights in the network of oscillators

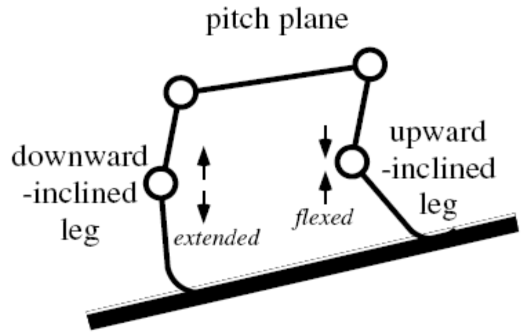


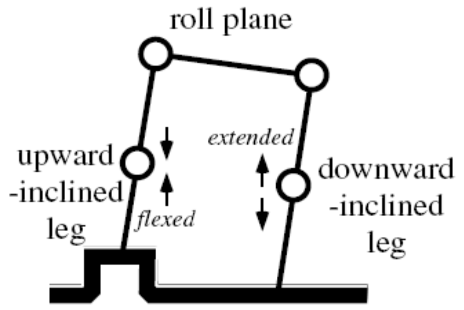
Reflexes:

The following reflexes are implemented:

- **stumbling-corrective reaction**. Contact to the paw dorsum generates an extension or retraction of the limb depending if it is loaded or not.

- **Vestibulospinal reflex** to maintain the body close to horizontal





Examples

CPG-and-reflex Control: summary

- Pros:
  - Distributed control, potentially robust against hardware faults.
  - Limit cycle behavior (controller-body-environment)
  - Robust against pertubations
  - Smooth trajectories due to the oscillators
- Cons:
  - Fewer mathematical tools than model-based methods
  - Not (yet) a clear design methodology, it is recommended to use reinforcement learning or optimization algorithms

Overall Summary: Model-based methods

Pros:

- **well established design methods** from control engineering,
- can benefit from mechanical design expertise for making kinematic and dynamic models.
- offer **mathematical proofs of stability** for some approaches, like hybrid zero dynamics.
- When used with offline and online optimization, can lead to impressive motor behaviors that would be difficult to hand-design.
- They can be **well suited for planning and for anticipatory behavior**

Cons:

- require **extensive expert knowledge**.
- The tuning of different control modules can take a long time.
- Their **performance is highly dependent on the quality of the robot and environment models**.
- They are generally **fragile against hardware lesions**.
- Methods relying on online optimization and planning are probably not as reactive as bio-inspired approaches.
- And **some approaches are computationally very intensive** (e.g. for offline and online optimization).

Overall Summary: Learning-based methods

Pros:

- They offer **generic design methods** that **require less expertise** than model-based approaches.
- Their performance **can beat human-designed model-based controllers**.
- They can combine supervised and unsupervised learning.
- They **can generate surprising new movements** that would be difficult/impossible to hand-design.

Cons:

- need for **very long training sequences**
- a strong **reliance on simulation and sim-to-real transfer**
- need of expertise in designing cost functions and training scenarios,
- **lack of proof of stability/performance**, no interpretability
- possibly poor generalization to handle situations (e.g. terrains) that have not been explored during training.

Overall Summary: Bio-inspired approaches

Pros:

- They are **very good for producing reactive and robust behavior**.
- They do not depend on explicit models.
- They **can be robust against hardware faults**.
- **Well suited for robots with passive dynamics, compliant behavior**.
- They are **computationally light**.

Cons:

- **No well-established design methods.**
- They normally cannot offer proof of stability/performance.
- They have **not much been used to do planning and to generate anticipatory behavior** (mainly focused on reactive behavior)

Possible exam questions

- Present key ideas, and the pros and cons of the different presented control approaches

- Note: this was just an introduction.
- Some next lectures and the student presentations will go deeper in several of these control approaches.