Legged Robots

Lecture 3

Overview of control approaches

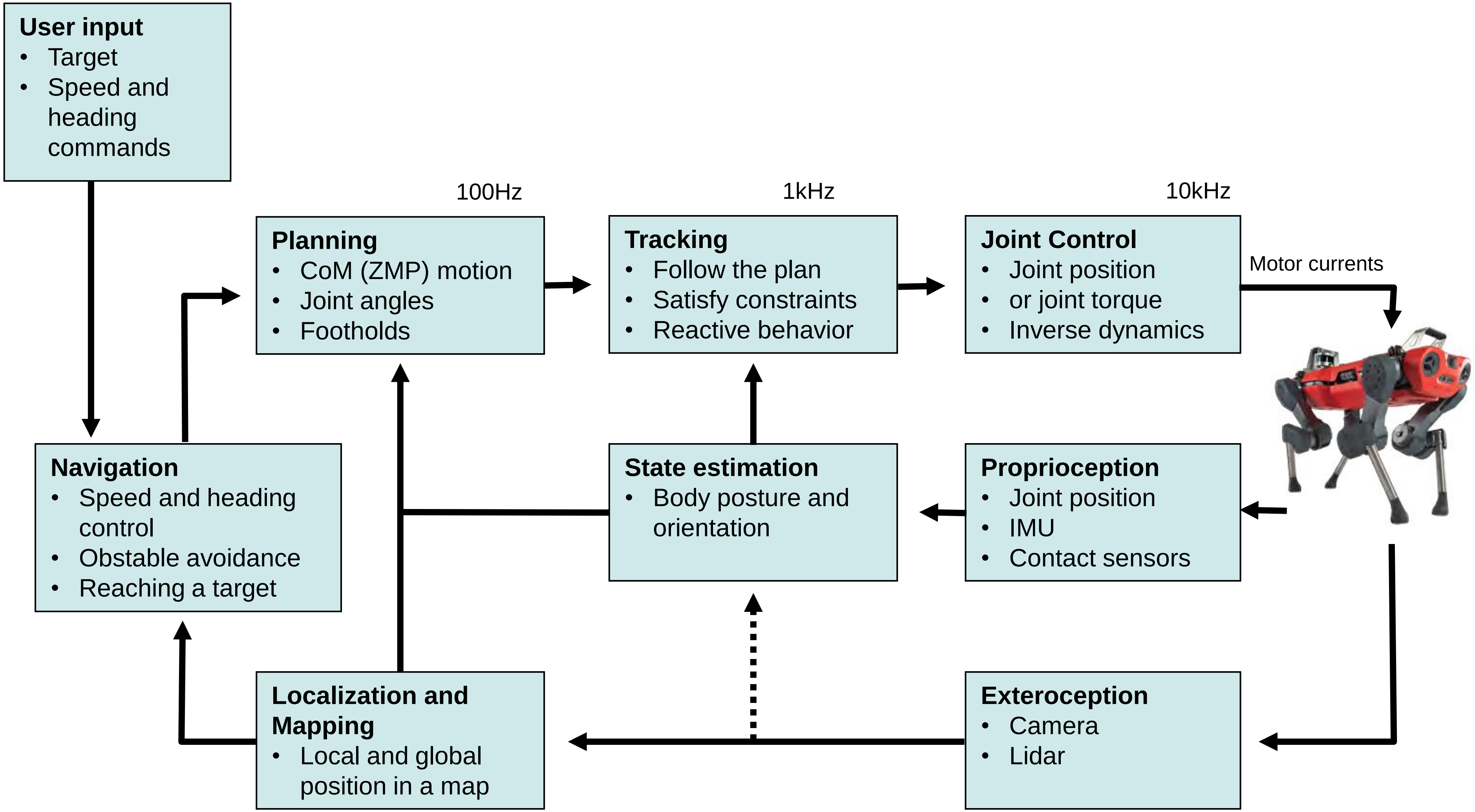
Auke Jan Ijspeert

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** (and heuristics-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

Typical control architecture



User input
- Target
- Speed and heading commands

100Hz

Planning
- CoM (ZMP) motion
- Joint angles
- Footholds

1kHz

Tracking
- Follow the plan
- Satisfy constraints
- Reactive behavior

10kHz

Joint Control
- Joint position
- or joint torque
- Inverse dynamics

Motor currents

Navigation
- Speed and heading control
- Obstable avoidance
- Reaching a target

State estimation
- Body posture and orientation

Proprioception
- Joint position
- IMU
- Contact sensors

Localization and Mapping
- Local and global position in a map

Exteroception
- Camera
- Lidar

Adapted from Marco Hutter’s lectures

Model-based methods

- Most extensively used until 5 years ago
- 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)
- Note: some approaches use models implicitly, e.g. virtual leg control and the SLIP model, I call these **heuristics-based methods**

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. Model predictive control (MPC)

Trajectory based methods

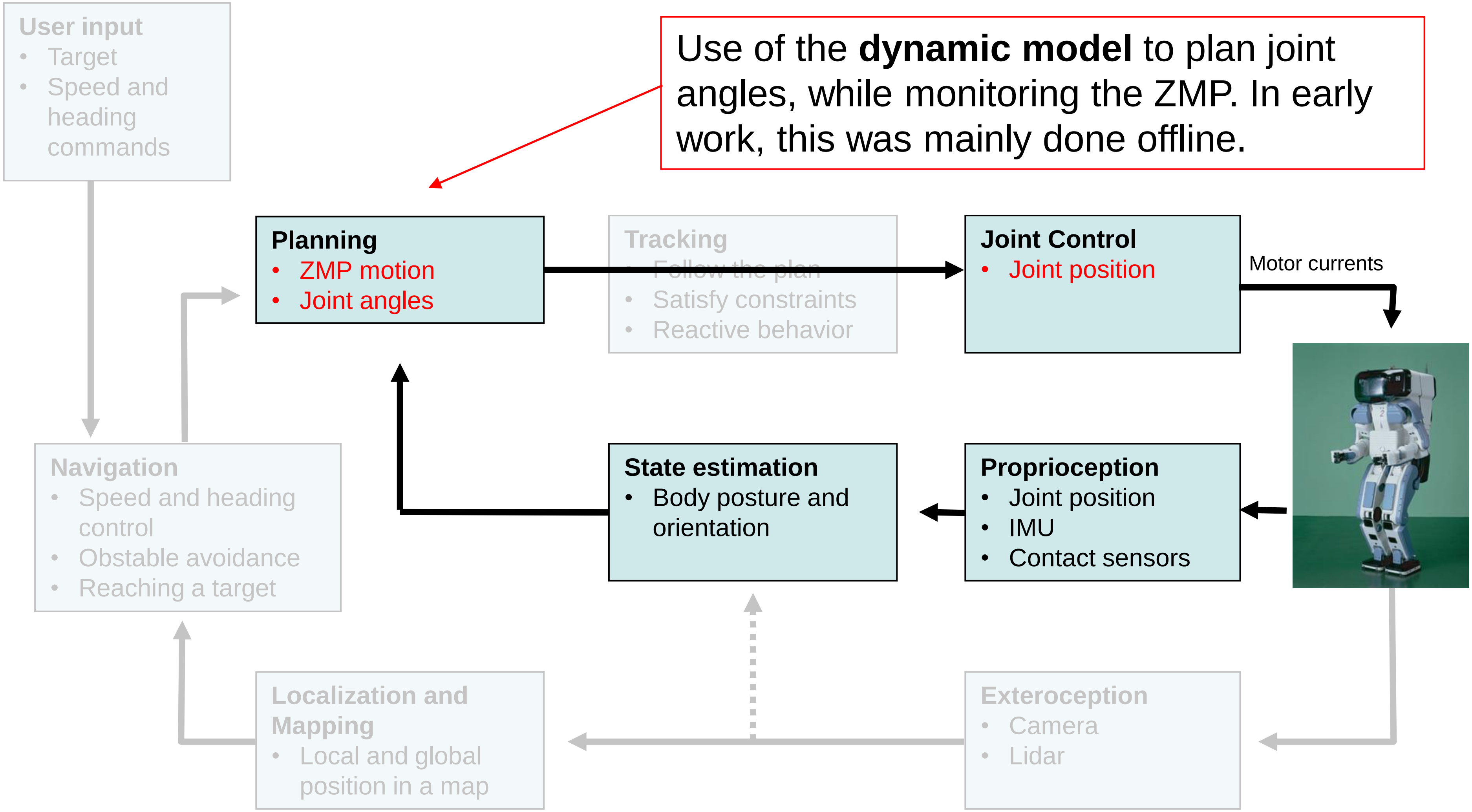
- Main idea: design **walking kinematic trajectories** (i.e. joint angles over time), and use the dynamic equations to test and prove that locomotion is stable

- **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)

Trajectory-based control

Use of the **dynamic model** to plan joint angles, while monitoring the ZMP. In early work, this was mainly done offline.



User input
- Target
- Speed and heading commands

Planning
- ZMP motion
- Joint angles

Tracking
- Follow the plan
- Satisfy constraints
- Reactive behavior

Joint Control
- Joint position

Motor currents

Navigation
- Speed and heading control
- Obstable avoidance
- Reaching a target

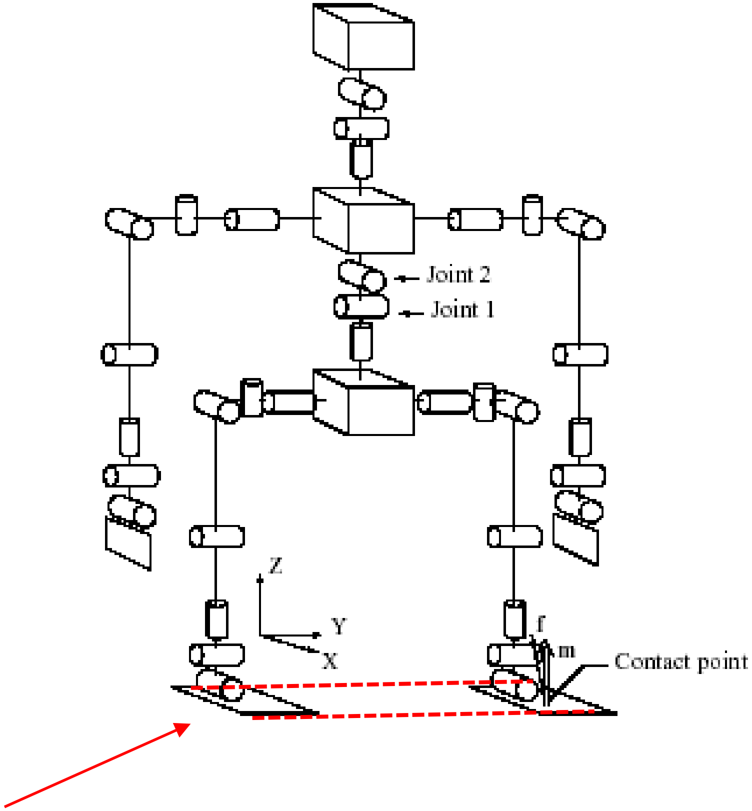
State estimation
- Body posture and orientation

Proprioception
- Joint position
- IMU
- Contact sensors

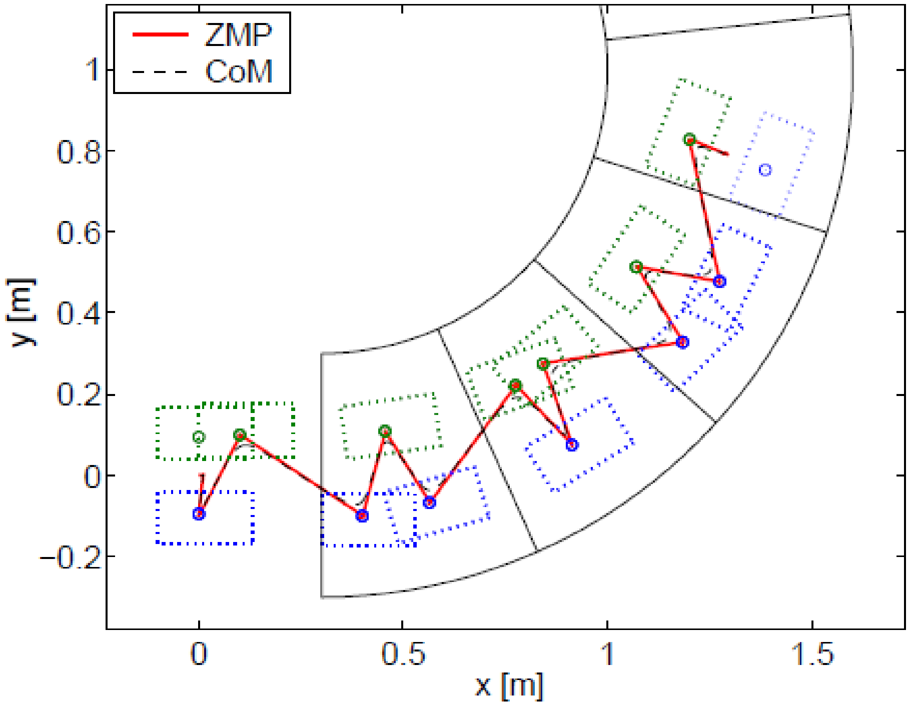
Localization and Mapping
- Local and global position in a map

Exteroception
- Camera
- Lidar

ZMP



Foot-print polygon



Kajita, et al. 2003. “Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point.” In *ICRA 2003* https://doi.org/10.1109/ROBOT.2003.1241826.

Locomotion is stable if the ZMP remains within the foot-print polygons <u>over time</u>

Trajectory-based with ZMP

- Example: (early) Honda robot, Asimo

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

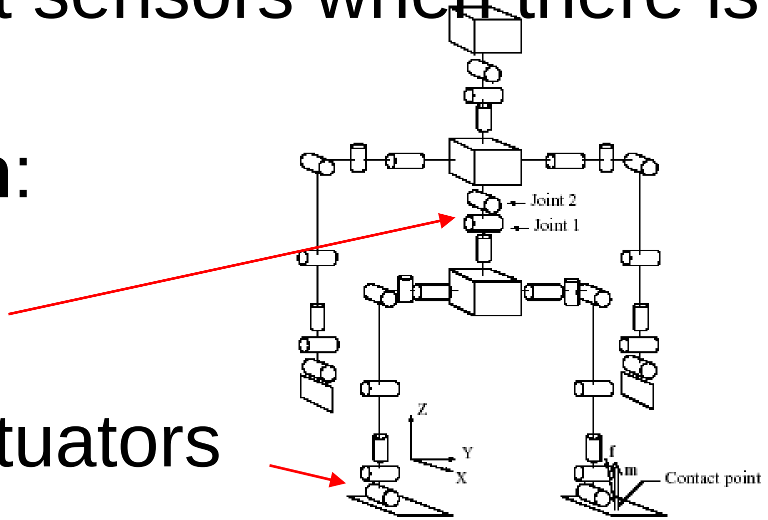
Trajectory-based with ZMP

Most used method in earlier papers:

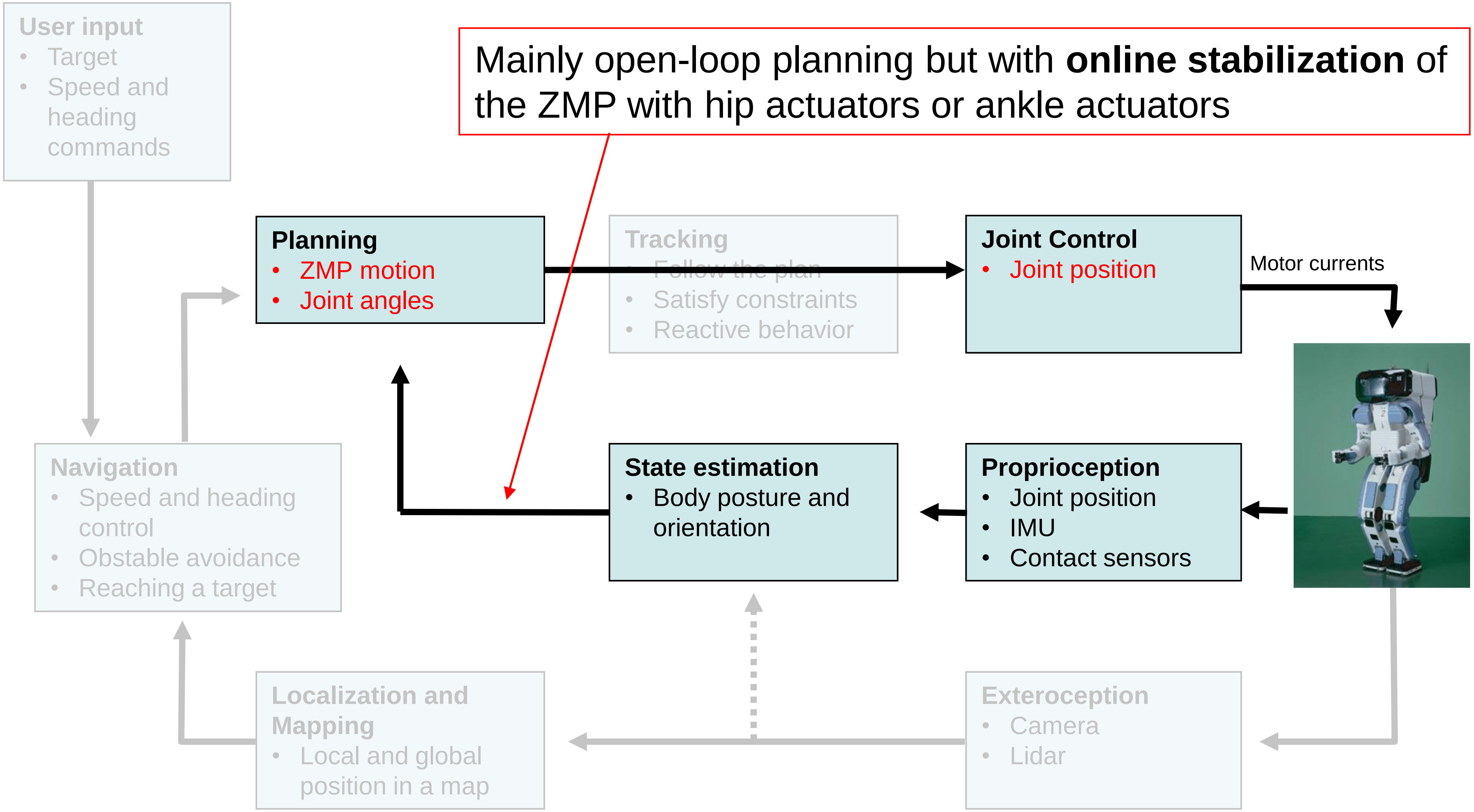
1. Human motion capture, educated guesses for getting trajectories, and/or use simple models like LIP for online footstep planning.
2. **Plan/modify trajectories offline** such that locomotion is **stable according to the ZMP criterion**
3. Add online stabilization to deal with perturbations, with ZMP estimated from foot sensors when there is a foot contact (CoP).

Example of **online stabilization**:

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



Trajectory-based control



Mainly open-loop planning but with **online stabilization** of the ZMP with hip actuators or ankle actuators

User input
- Target
- Speed and heading commands

Planning
- ZMP motion
- Joint angles

Tracking
- Follow the plan
- Satisfy constraints
- Reactive behavior

Joint Control
- Joint position

Motor currents

Navigation
- Speed and heading control
- Obstacle avoidance
- Reaching a target

State estimation
- Body posture and orientation

Proprioception
- Joint position
- IMU
- Contact sensors

Localization and Mapping
- Local and global position in a map

Exteroception
- Camera
- Lidar

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**
- Fragile against unexpected events (e.g. pushes)
- 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.

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. 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**.

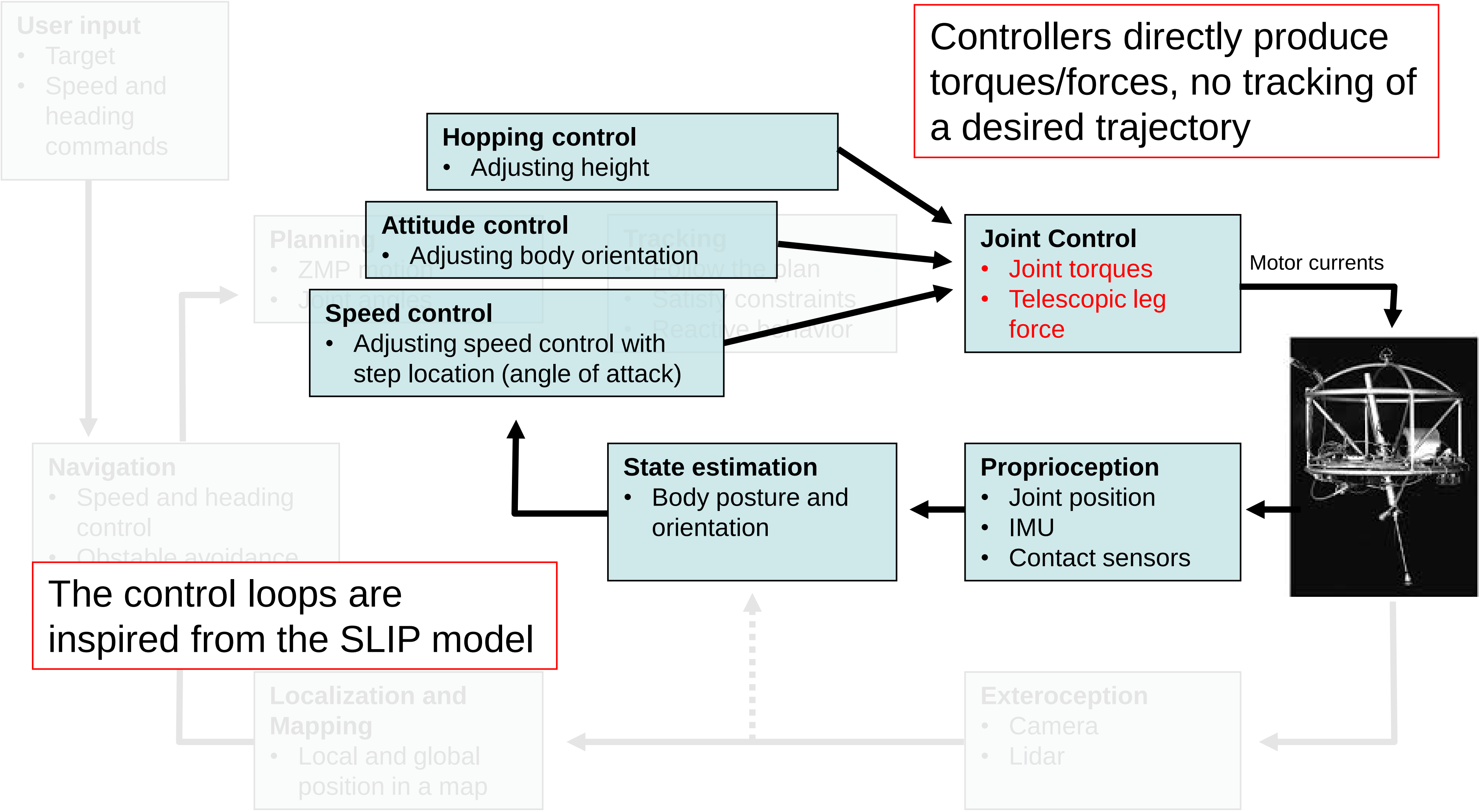
- One- two- and four-legged robots controlled by a similar approach

- 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*)

Virtual Leg Control

Controllers directly produce torques/forces, no tracking of a desired trajectory



User input
- Target
- Speed and heading commands

Hopping control
- Adjusting height

Attitude control
- Adjusting body orientation

Planning
- ZMP motion
- Joint angles

Tracking
- Follow the plan
- Satisfy constraints
- Reactive behavior

Speed control
- Adjusting speed control with step location (angle of attack)

Joint Control
- Joint torques
- Telescopic leg force

Navigation
- Speed and heading control
- Obstacle avoidance

State estimation
- Body posture and orientation

Proprioception
- Joint position
- IMU
- Contact sensors

The control loops are inspired from the SLIP model

Localization and Mapping
- Local and global position in a map

Exteroception
- Camera
- Lidar

Robots at MIT LegLab

Similarities between 1, 2, and 4 legs

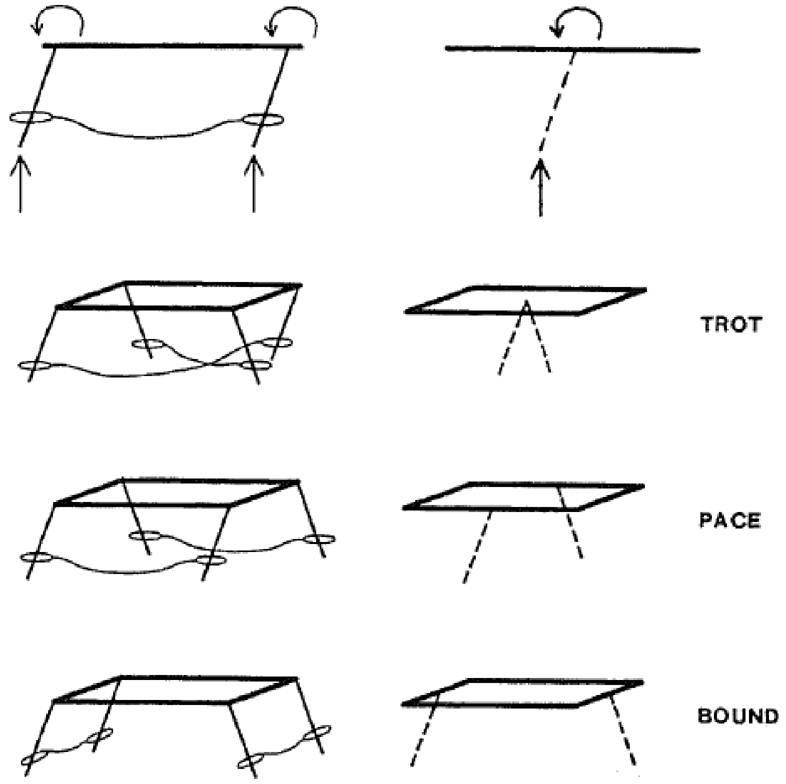


Fig. 3. Virtual legs. When two legs are coordinated to act in unison, they can be represented by a functionally equivalent *virtual leg*. The virtual leg and the original pair of physical legs both exert the same forces and moments on the body, so they both result in the same behavior. When each pair 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 for many years (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.

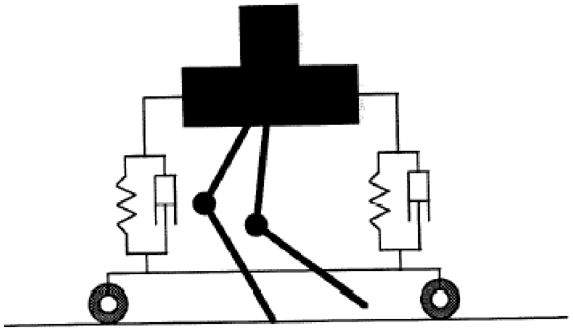
Examples of model-based approaches

Model-based control:

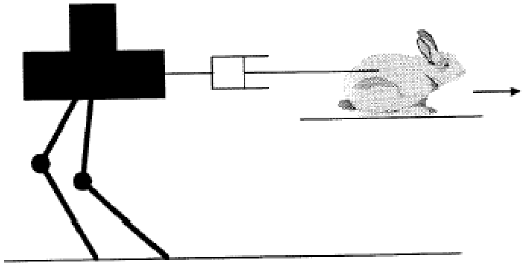
1. Trajectory based methods (ZMP)
2. Virtual leg control (Raibert)
3. **Virtual model control (Pratt et al)**
4. Hybrid Zero Dynamics control
5. Planning methods (Little dog project)
6. Model predictive control (MPC)

Virtual Model Control

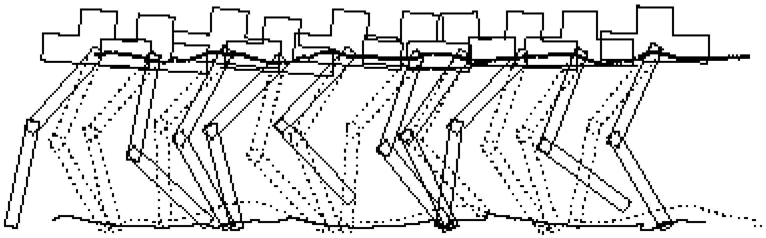
- Nice example of model-based control: Virtual Model Control (G.Pratt)
- Idea: create **virtual elements** to keep the robot upright and have it move forward
- Then compute the necessary torques such that the robot motors replicate the effect of those virtual elements



Virtual granny walker
for balance control



Virtual bunny
for velocity control



Virtual Model Control

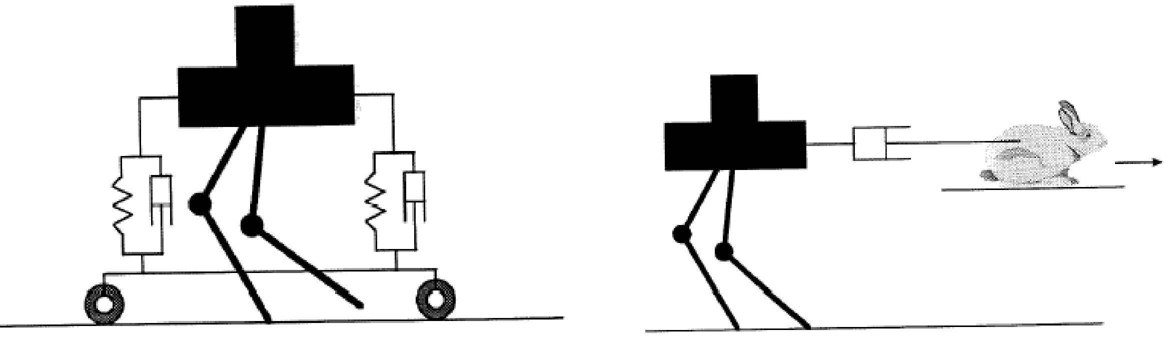
- For each virtual element producing a force F, the joint torque needed to produce that virtual force can be computed with:

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

- **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 $\vec{\tau} = [\tau_a\ \tau_k\ \tau_h]^T$

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

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

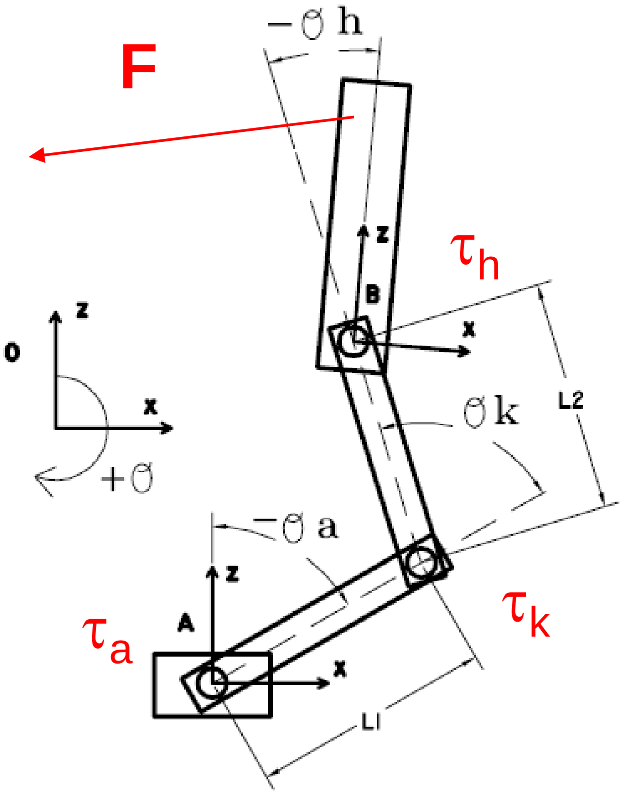
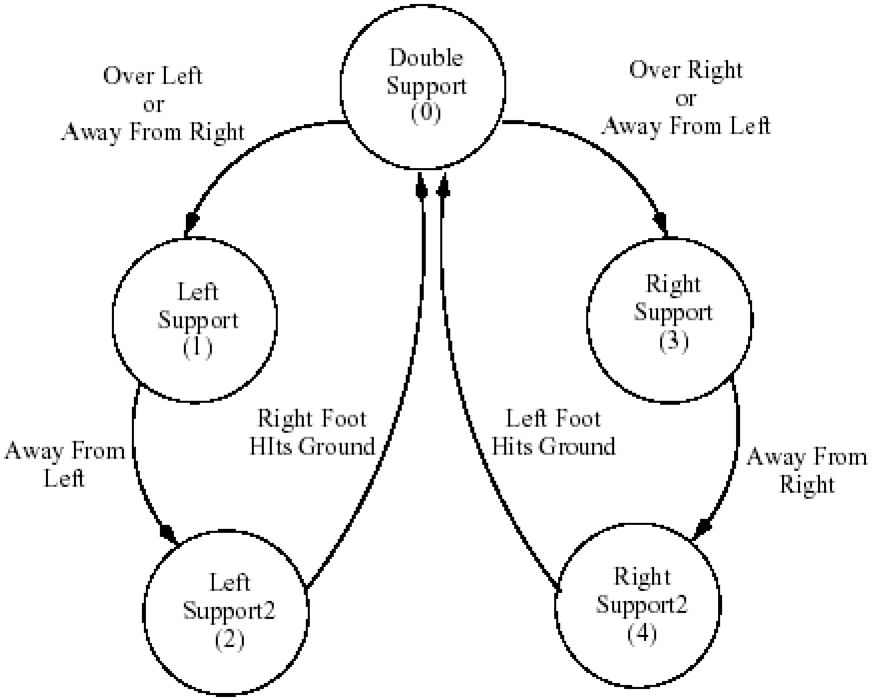


Fig. 3. Single-leg implementation. Reaction frame $\{A\}$ is assumed to be in the same orientation as reference frame $\{O\}$ so that ${}^O_A R = I$.

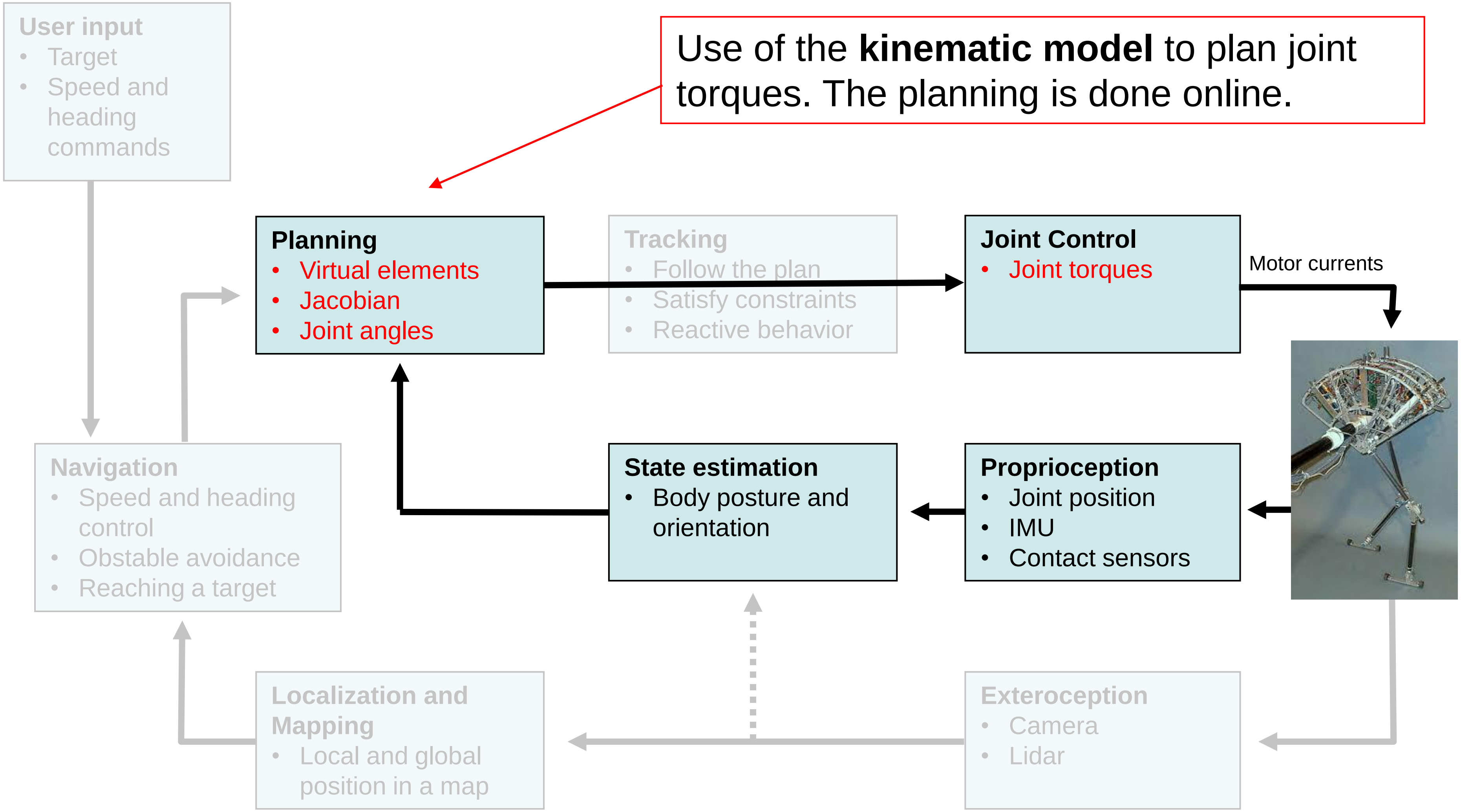
Virtual Model Control

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

Virtual Model Control



Use of the **kinematic model** to plan joint torques. The planning is done online.

User input
- Target
- Speed and heading commands

Planning
- Virtual elements
- Jacobian
- Joint angles

Tracking
- Follow the plan
- Satisfy constraints
- Reactive behavior

Joint Control
- Joint torques

Motor currents

Navigation
- Speed and heading control
- Obstacle avoidance
- Reaching a target

State estimation
- Body posture and orientation

Proprioception
- Joint position
- IMU
- Contact sensors

Localization and Mapping
- Local and global position in a map

Exteroception
- Camera
- Lidar

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??

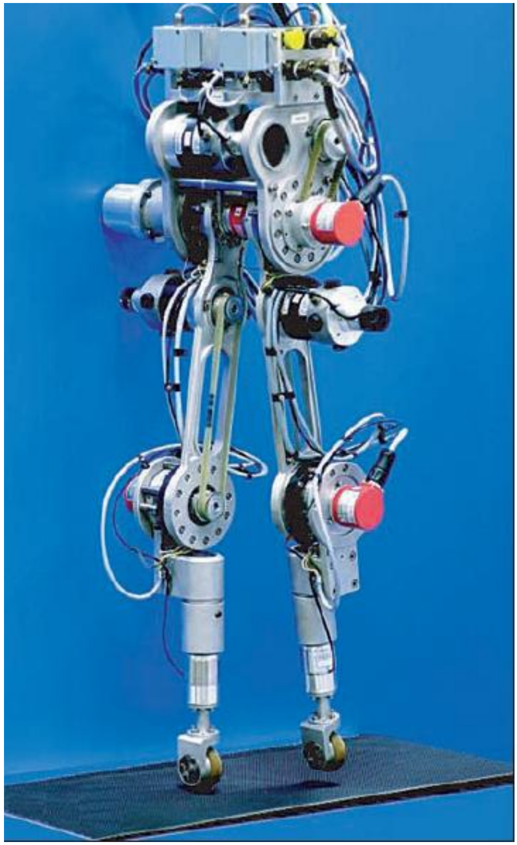
Reference: Pratt et al, Virtual Model Control: An intuitive approach for bipedal locomotion, The International Journal of Robotics Research, Vol. 20, No. 2, 129-143 2001 .

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. Model predictive control (MPC)

Hybrid Zero Dynamics control



Rabbit robot

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

Specific property: no feet

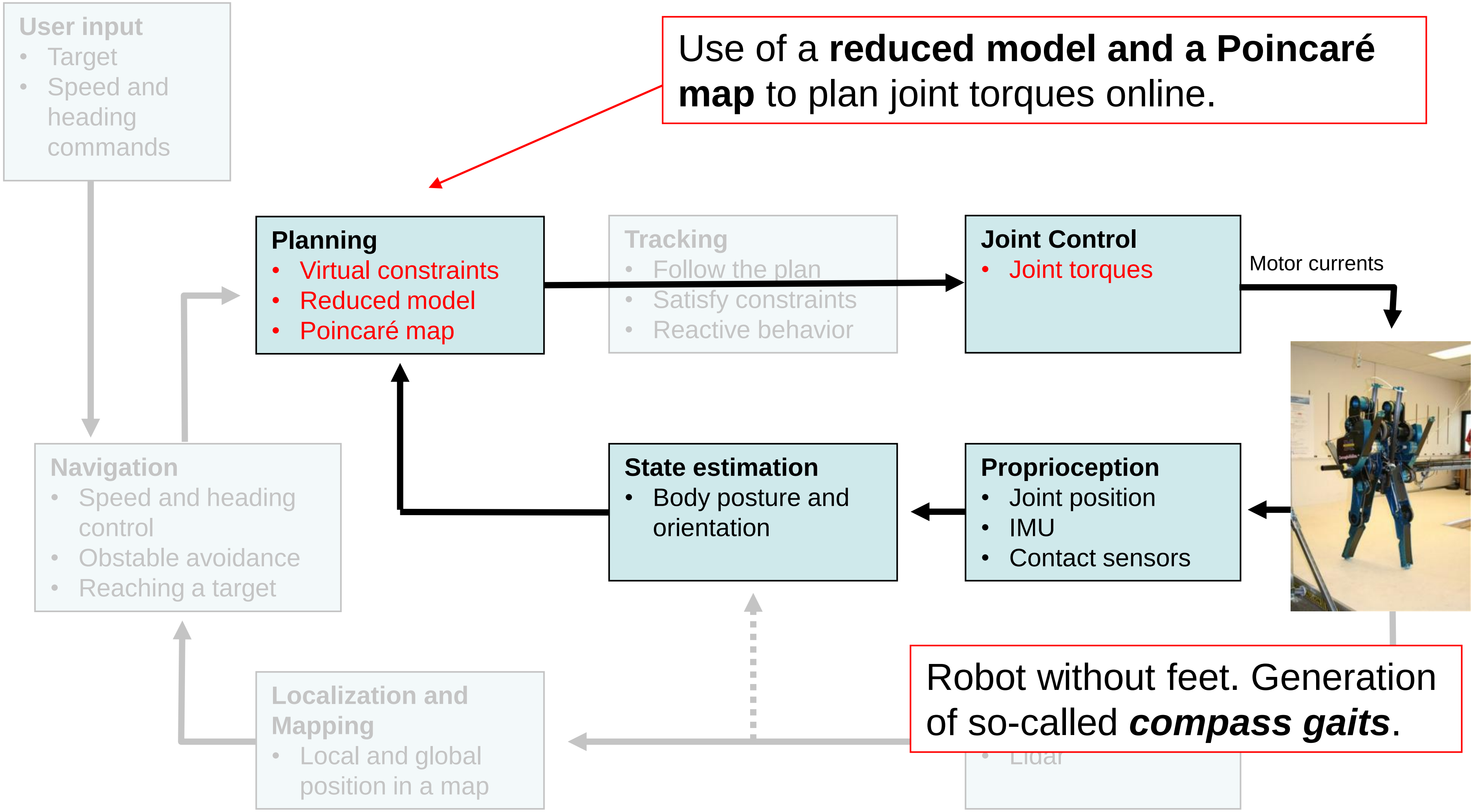
Very nice theoretical framework to obtain **provably asymptotically stable walking and running gaits**

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**

Hybrid Zero Dynamics

Use of a **reduced model and a Poincaré map** to plan joint torques online.



User input
- Target
- Speed and heading commands

Planning
- Virtual constraints
- Reduced model
- Poincaré map

Tracking
- Follow the plan
- Satisfy constraints
- Reactive behavior

Joint Control
- Joint torques

Motor currents

Navigation
- Speed and heading control
- Obstable avoidance
- Reaching a target

State estimation
- Body posture and orientation

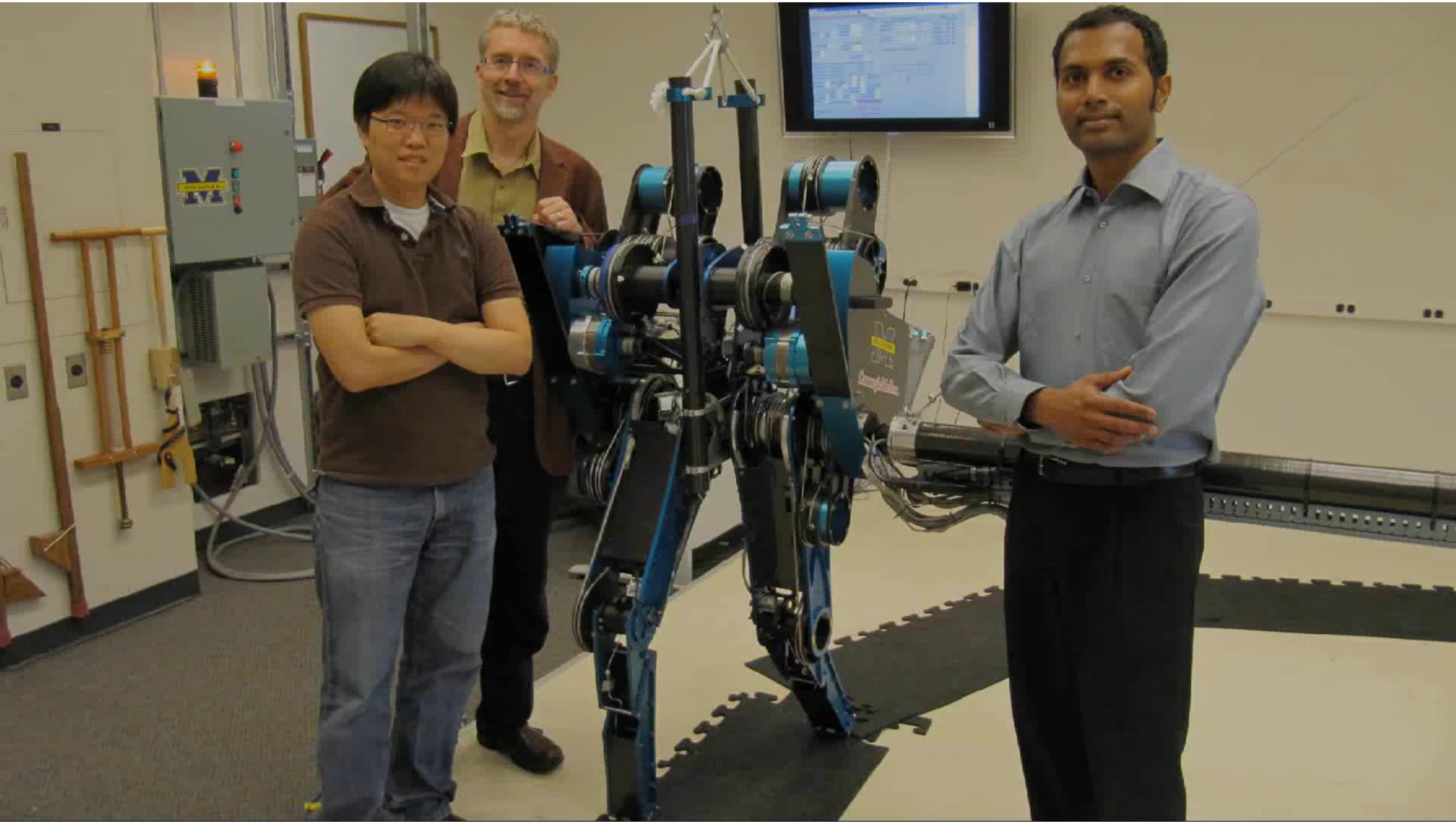
Proprioception
- Joint position
- IMU
- Contact sensors

Localization and Mapping
- Local and global position in a map

- Lidar

Robot without feet. Generation of so-called ***compass gaits***.

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.
- Highly-dependent on good state estimation

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:

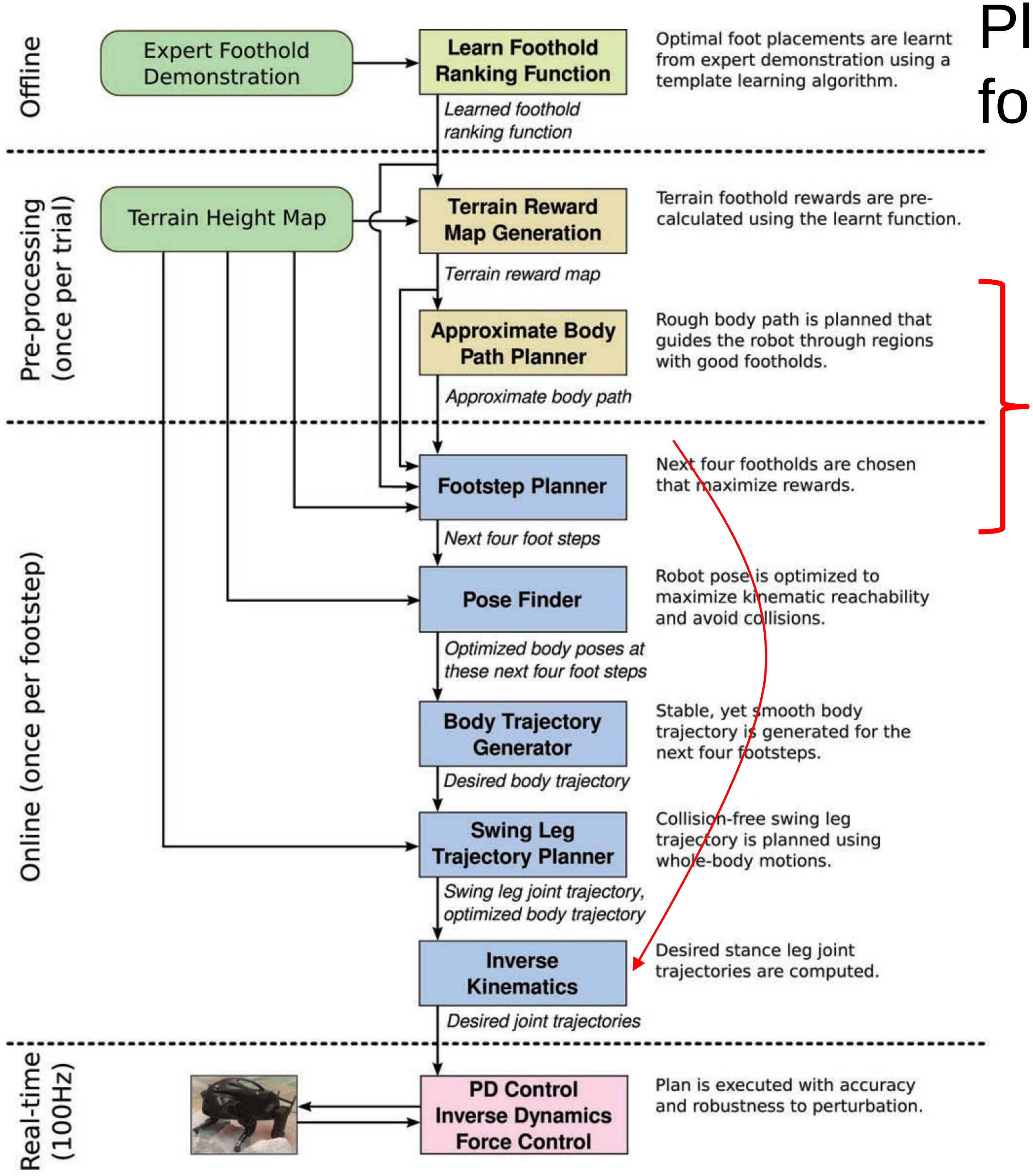
1. Trajectory based methods (ZMP)
2. Virtual leg control (Raibert)
3. Virtual model control (Pratt et al)
4. Hybrid Zero Dynamics control
5. **Planning methods (Little dog project)**
6. Model predictive control (MPC)

Planning methods for complex terrains

Buchli et al 2009

- 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

Planning methods for complex terrains



Offline

Expert Foothold Demonstration

Learn Foothold Ranking Function

Optimal foot placements are learnt from expert demonstration using a template learning algorithm.

Learned foothold ranking function

Pre-processing (once per trial)

Terrain Height Map

Terrain Reward Map Generation

Terrain foothold rewards are pre-calculated using the learnt function.

Terrain reward map

Approximate Body Path Planner

Rough body path is planned that guides the robot through regions with good footholds.

Approximate body path

Online (once per footstep)

Footstep Planner

Next four footholds are chosen that maximize rewards.

Next four foot steps

Pose Finder

Robot pose is optimized to maximize kinematic reachability and avoid collisions.

Optimized body poses at these next four foot steps

Body Trajectory Generator

Stable, yet smooth body trajectory is generated for the next four footsteps.

Desired body trajectory

Swing Leg Trajectory Planner

Collision-free swing leg trajectory is planned using whole-body motions.

Swing leg joint trajectory, optimized body trajectory

Inverse Kinematics

Desired stance leg joint trajectories are computed.

Desired joint trajectories

Real-time (100Hz)

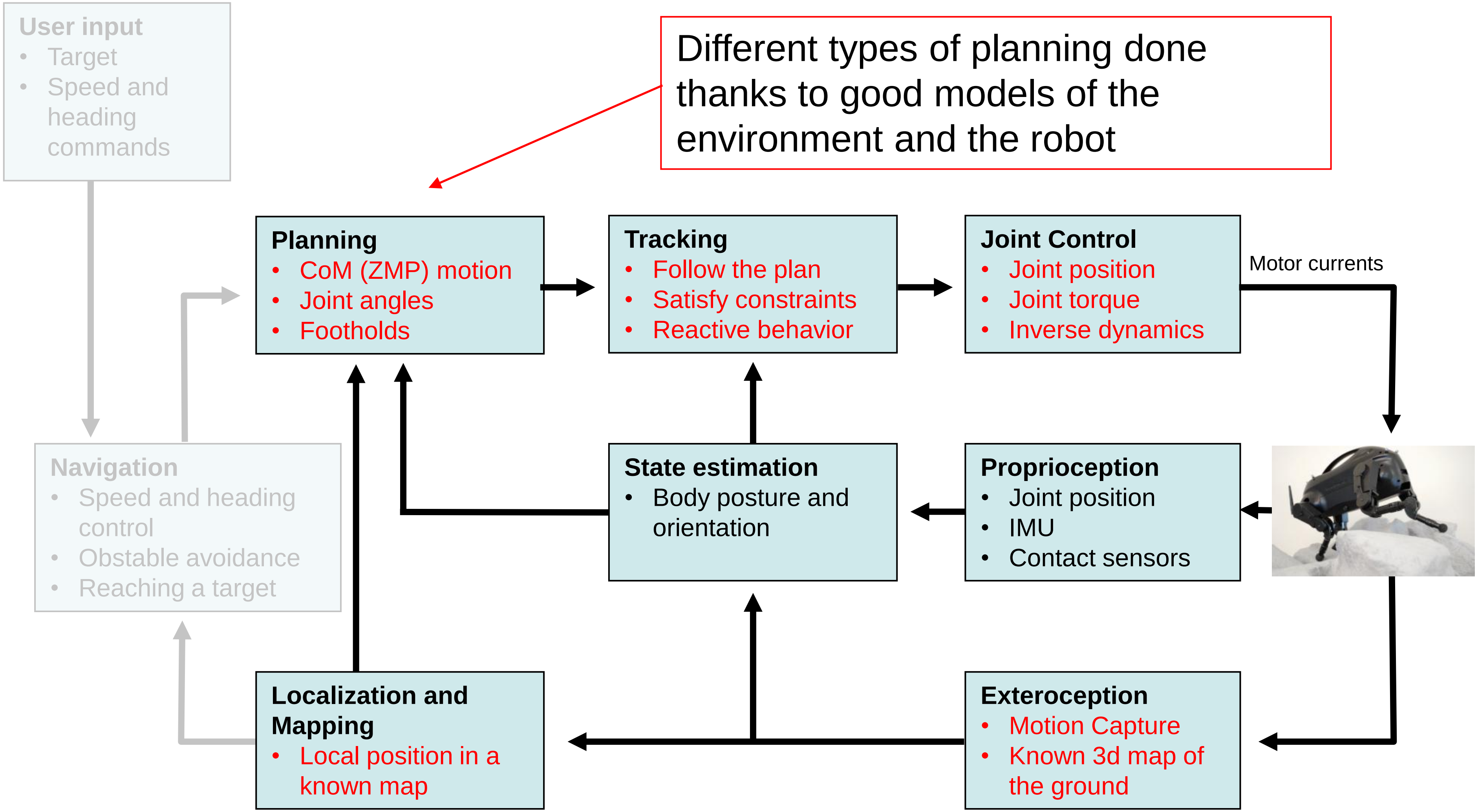
PD Control Inverse Dynamics Force Control

Plan is executed with accuracy and robustness to perturbation.

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

Kalakrishnan, M., Buchli, J., Pastor, P., Mistry, M., & Schaal, S. (2011). Learning, planning, and control for quadruped locomotion over challenging terrain. The International Journal of Robotics Research, 30(2), 236–258. https://doi.org/10.1177/0278364910388677

Planning methods for complex terrains



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

User input
- Target
- Speed and heading commands

Planning
- CoM (ZMP) motion
- Joint angles
- Footholds

Tracking
- Follow the plan
- Satisfy constraints
- Reactive behavior

Joint Control
- Joint position
- Joint torque
- Inverse dynamics

Navigation
- Speed and heading control
- Obstacle avoidance
- Reaching a target

State estimation
- Body posture and orientation

Proprioception
- Joint position
- IMU
- Contact sensors

Localization and Mapping
- Local position in a known map

Exteroception
- Motion Capture
- Known 3d map of the ground

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., Mistry, M., & Schaal, S. (2011). Learning, planning, and control for quadruped locomotion over challenging terrain. The International Journal of Robotics Research, 30(2), 236–258. https://doi.org/10.1177/0278364910388677.

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. **Model predictive control (MPC)**

Model predictive control

- Model Predictive Control (MPC), a particular type of optimal control, has become a popular method for controlling legged robots

- MPC uses a **dynamic model** of the robot and **optimization** to compute **sequences of control actions** (desired joint angles or torques).

- At each control step, MPC solves an **optimization problem** to determine the best **sequence of control actions** that minimizes **a cost function** and satisfies some **constraints** over a **finite time horizon.**

- It is/was the most popular approach the last 10-15 years (now Reinforcement Learning is more popular), we will see many examples in the guest lectures and the paper to read.

Model predictive control

State and Control space: $\mathcal{S}, \mathcal{U}$

State: $x_k \in \mathcal{S}, k = 0, 1, \ldots$

Control: $u_k \in \mathcal{U}, k = 0, 1, \ldots$

Dynamical System:

$$x_{k+1} = f(x_k, u_k)$$
$$x_{k+1} = f(x_k, u_k, w_k)$$

stochastic

Output/observation (feature):

$$y_k = h(x_k, u_k) + n_k$$

stochastic

A model of the robot *f(x,u)* (complete or simplified) must be available

Horizon of N steps

Cost: $g(x_k, u_k) \in \mathbb{R}$

Total cost function:

$$J(x_0; u_0, \ldots, u_N) = \sum_{k=0}^{N} g(x_k, u_k)$$

Control law: $u(x_k);\ u^*(x_k)$

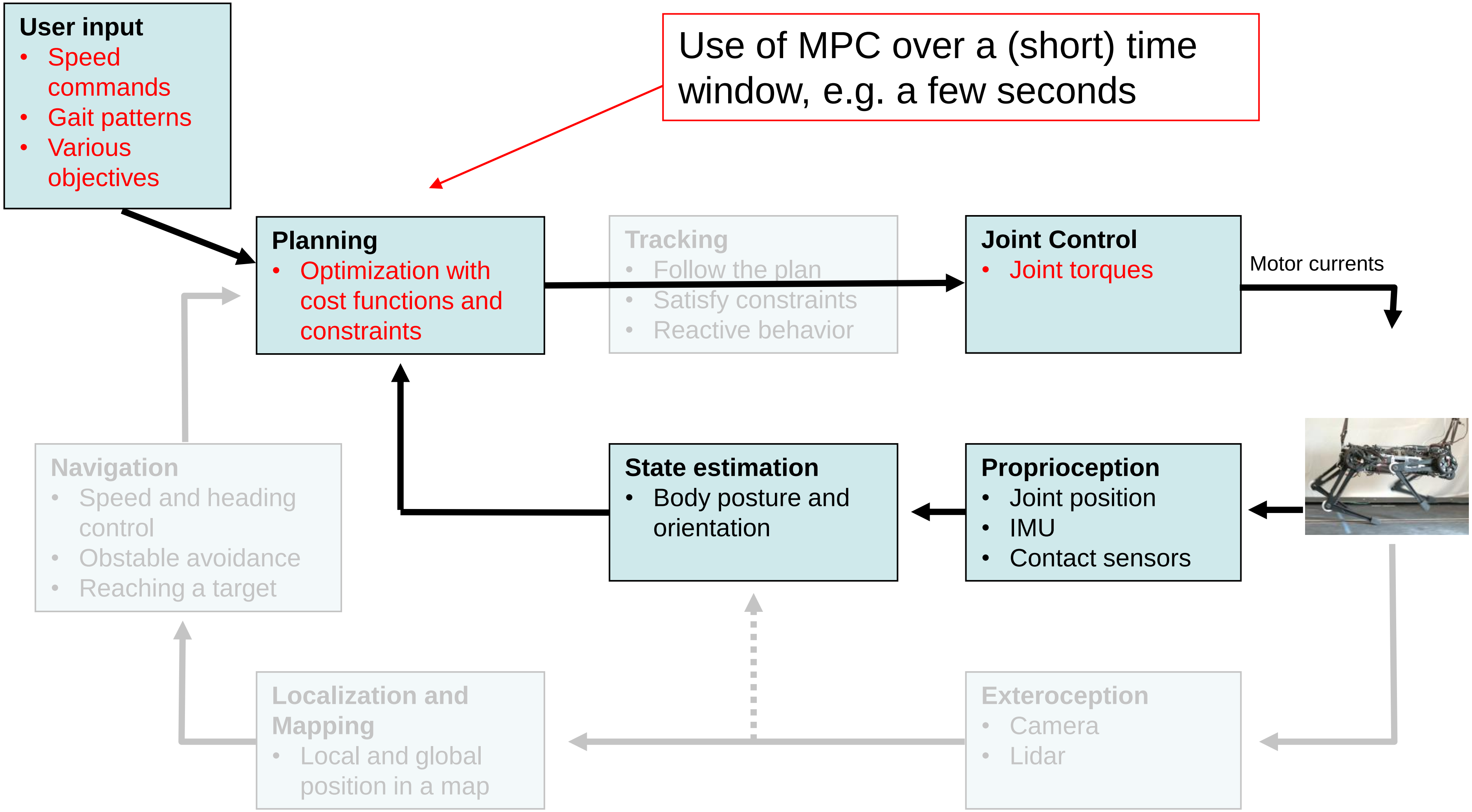
$$x_{k+1} = f(x_k, u(x_k))$$
$$u_{k+1} = u(x_{k+1})$$

Value function (minimal cost to go):

$$J^*(x_0) = \min_{u(\cdot)} \sum_{k=0}^{N} g(x_k, u(x_k, k))$$

Multiple constraints can be added, e.g. *u < Constant, x_COM > Constant, …*

Example of MPC



Use of MPC over a (short) time window, e.g. a few seconds

User input
- Speed commands
- Gait patterns
- Various objectives

Planning
- Optimization with cost functions and constraints

Tracking
- Follow the plan
- Satisfy constraints
- Reactive behavior

Joint Control
- Joint torques

Motor currents

Navigation
- Speed and heading control
- Obstacle avoidance
- Reaching a target

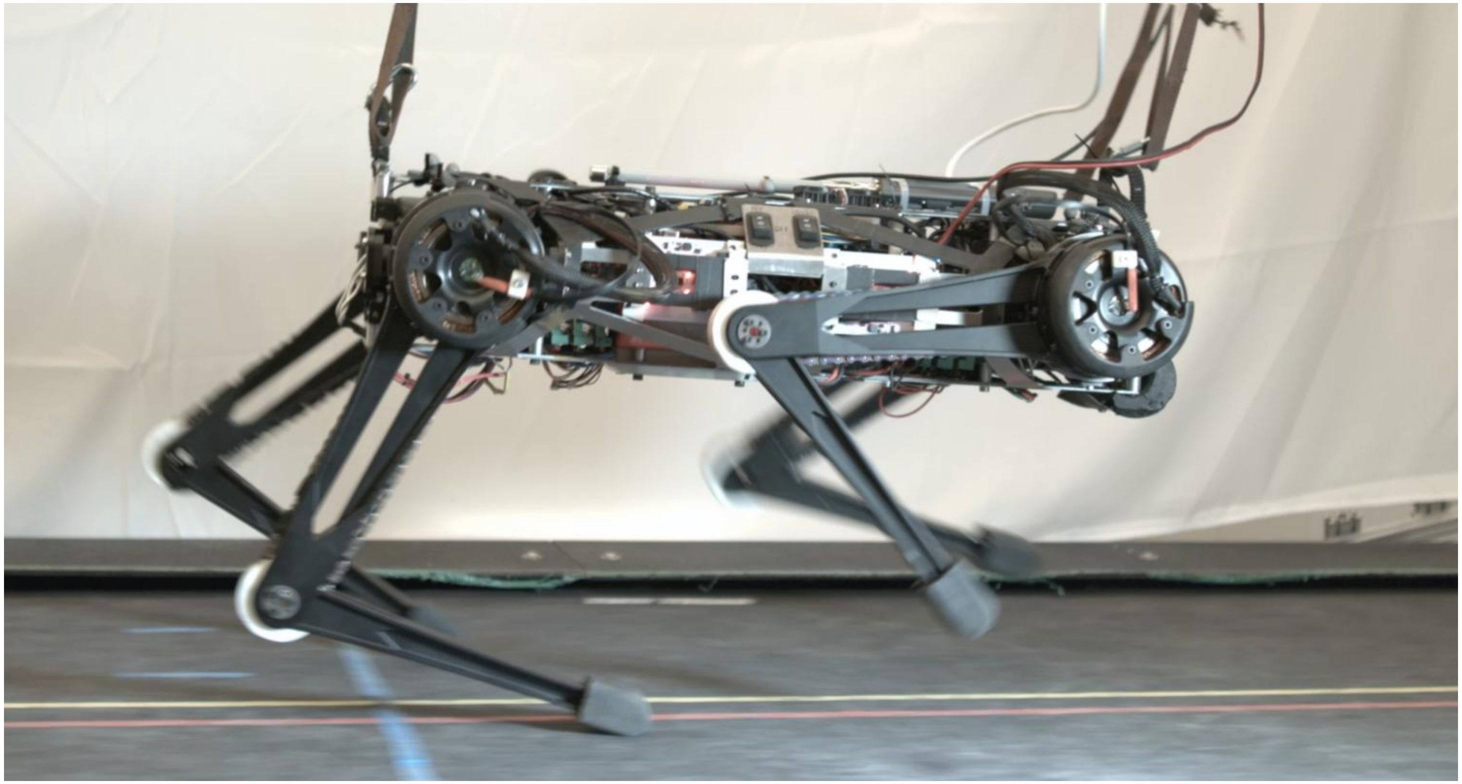
State estimation
- Body posture and orientation

Proprioception
- Joint position
- IMU
- Contact sensors

Localization and Mapping
- Local and global position in a map

Exteroception
- Camera
- Lidar

Example: MPC on the MIT Cheetah 3



https://www.youtube.com/watch?v=q6zxCvCxhic

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. https://doi.org/10.1109/IROS.2018.8594448

Model predictive control

- Examples of **cost functions**: minimizing energy, minimizing the difference to a desired velocity (speed and heading), satisfying a gait (footfall pattern), ...

- MPC can be used for many things, **foot step planning, body posture, joint torques**, etc. sometimes with different models and optimizations running in parallel.

- Very useful to obtain **highly versatile gaits and whole body control**.

- Probably used on the ATLAS videos of Boston Dynamics

- Impressive gaits on the MIT Cheetah 3 robot

Model predictive control

- Pros:
  - Ability to generate a **large class of movements**: walking + many others
  - **Versatile:** allows one to design controllers in **task space in addition to joint space**

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

- Cons:
  - Requires **accurate dynamic models**
  - Requires **iterative tuning of costs and constraints**
  - **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

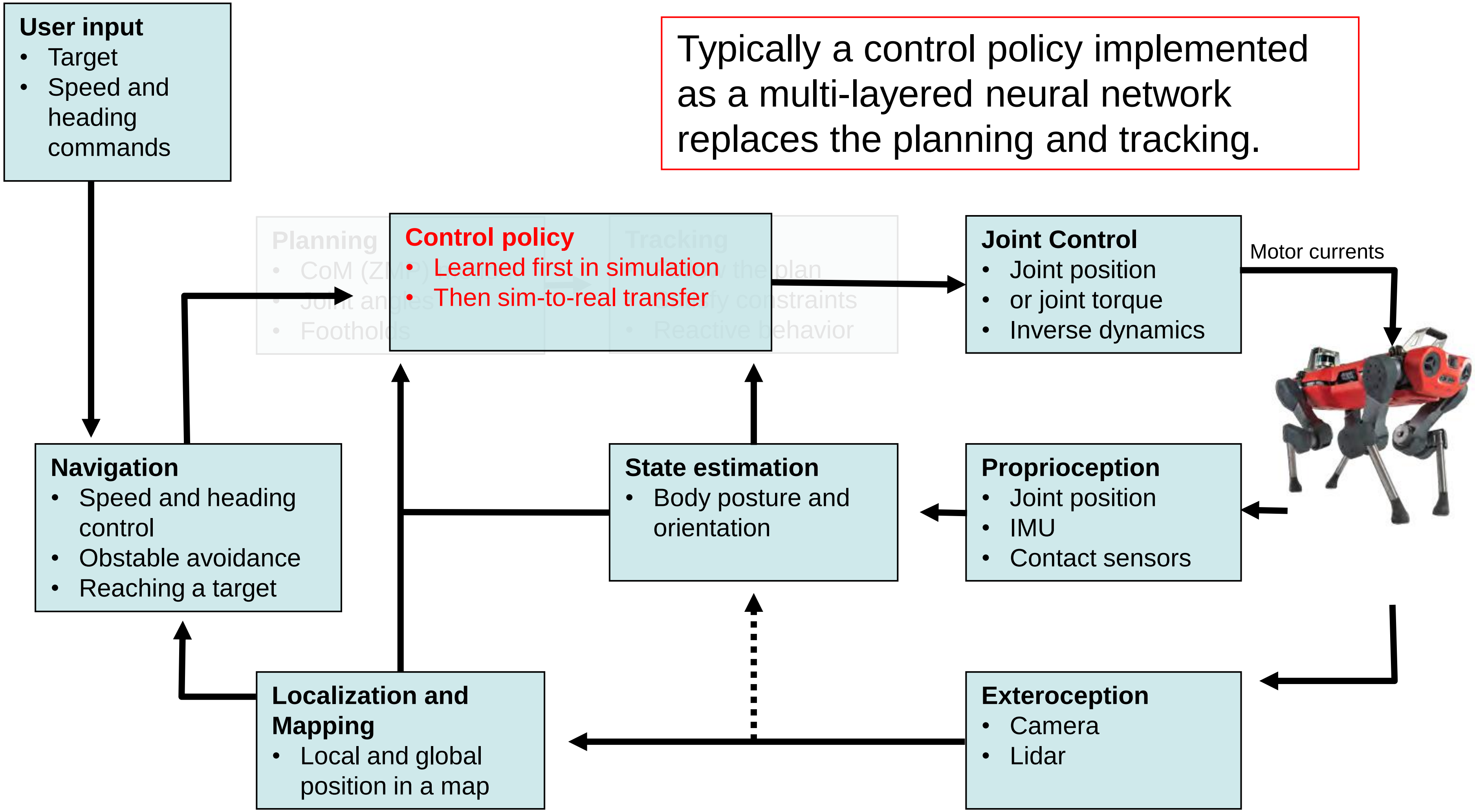
- Learning-based methods are recently becoming highly popular thanks to progress in machine learning, increased computation power, and faster simulators.

- 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).

Reinforcement learning

Typically a control policy implemented as a multi-layered neural network replaces the planning and tracking.



User input
- Target
- Speed and heading commands

Planning
- CoM (ZMP)
- Joint angles
- Footholds

Control policy
- Learned first in simulation
- Then sim-to-real transfer

Tracking
- w the plan
- Satisfy constraints
- Reactive behavior

Joint Control
- Joint position
- or joint torque
- Inverse dynamics

Motor currents

Navigation
- Speed and heading control
- Obstable avoidance
- Reaching a target

State estimation
- Body posture and orientation

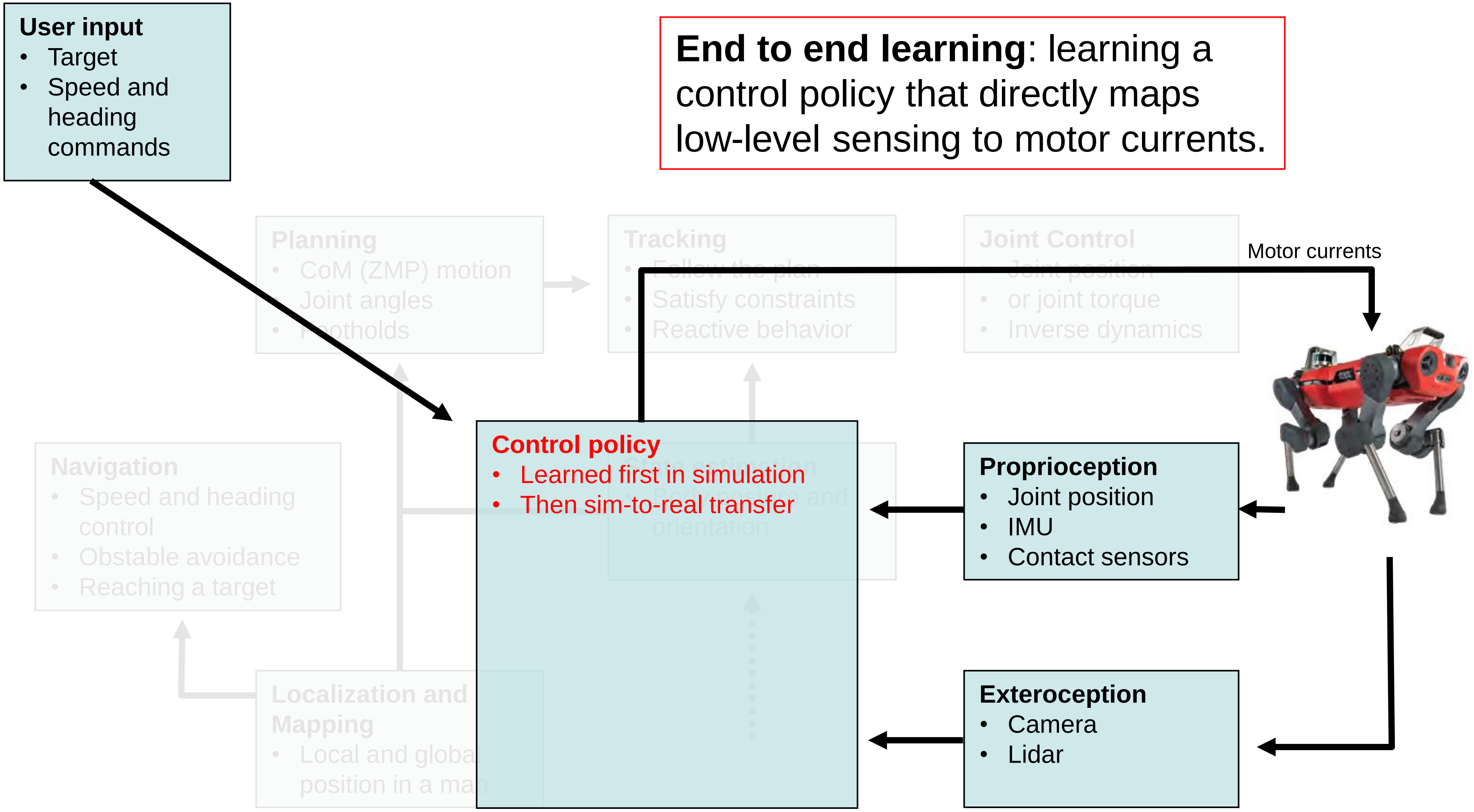
Proprioception
- Joint position
- IMU
- Contact sensors

Localization and Mapping
- Local and global position in a map

Exteroception
- Camera
- Lidar

Reinforcement learning, **end to end** learning

End to end learning: learning a control policy that directly maps low-level sensing to motor currents.



User input
- Target
- Speed and heading commands

Planning
- CoM (ZMP) motion
- Joint angles
- Footholds

Tracking
- Follow the plan
- Satisfy constraints
- Reactive behavior

Joint Control
- Joint position
- or joint torque
- Inverse dynamics

Motor currents

Navigation
- Speed and heading control
- Obstable avoidance
- Reaching a target

Control policy
- Learned first in simulation
- Then sim-to-real transfer

State estimation
- Body posture and orientation

Proprioception
- Joint position
- IMU
- Contact sensors

Localization and Mapping
- Local and globa
- position in a ma

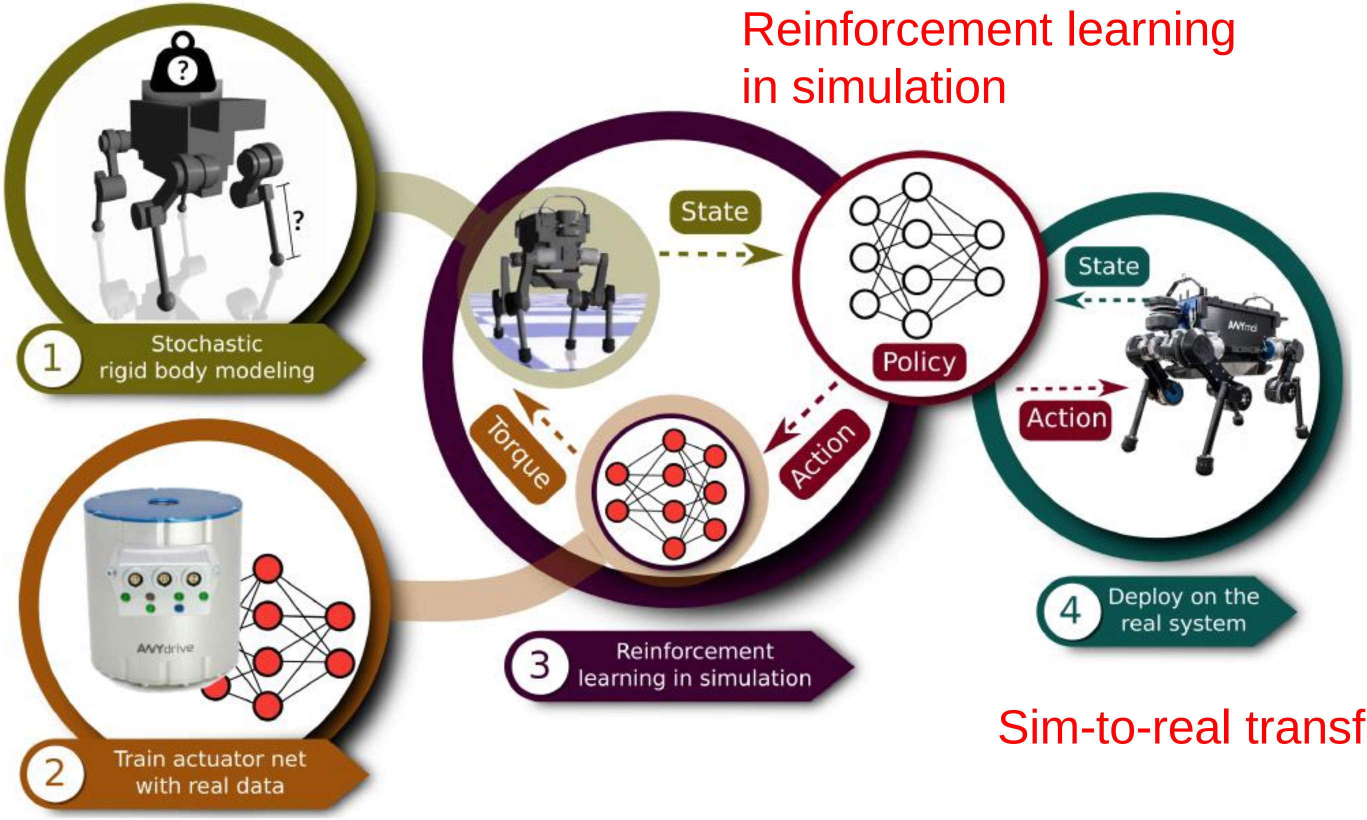
Exteroception
- Camera
- Lidar

Learning with ANYmal

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

Good simulator

Reinforcement learning in simulation



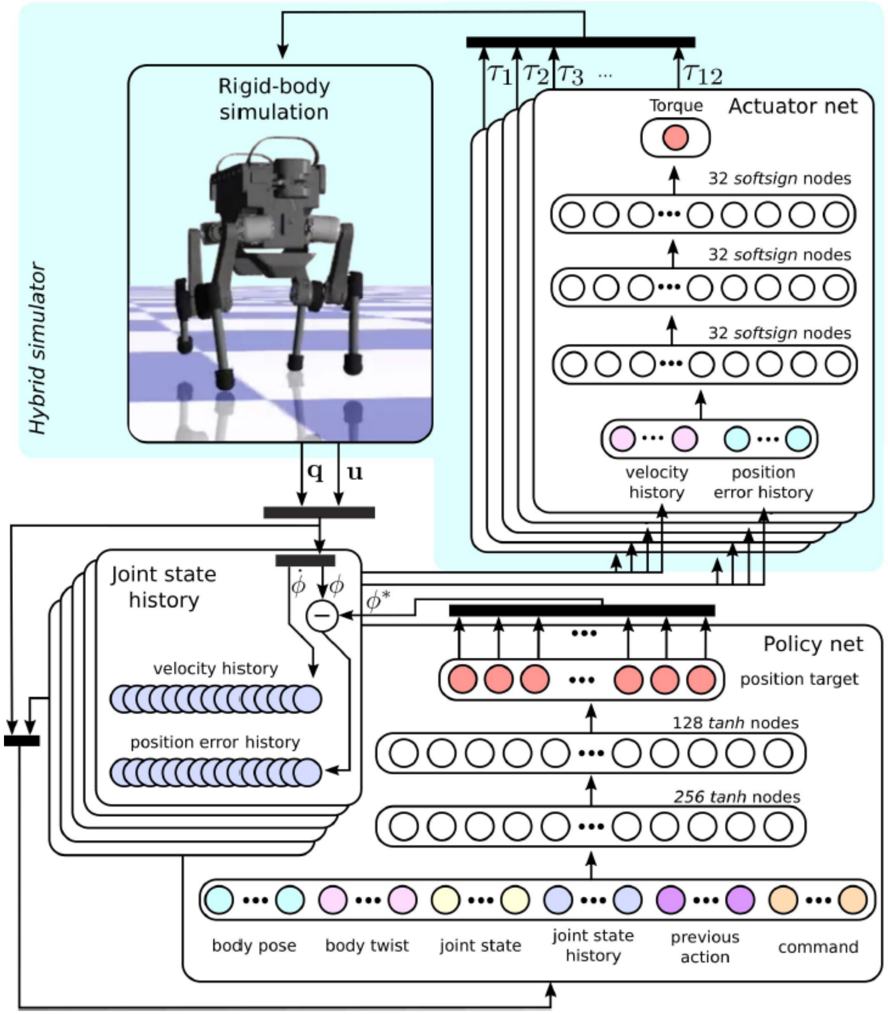
Supervised learning of actuator dynamics

Sim-to-real transfer

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

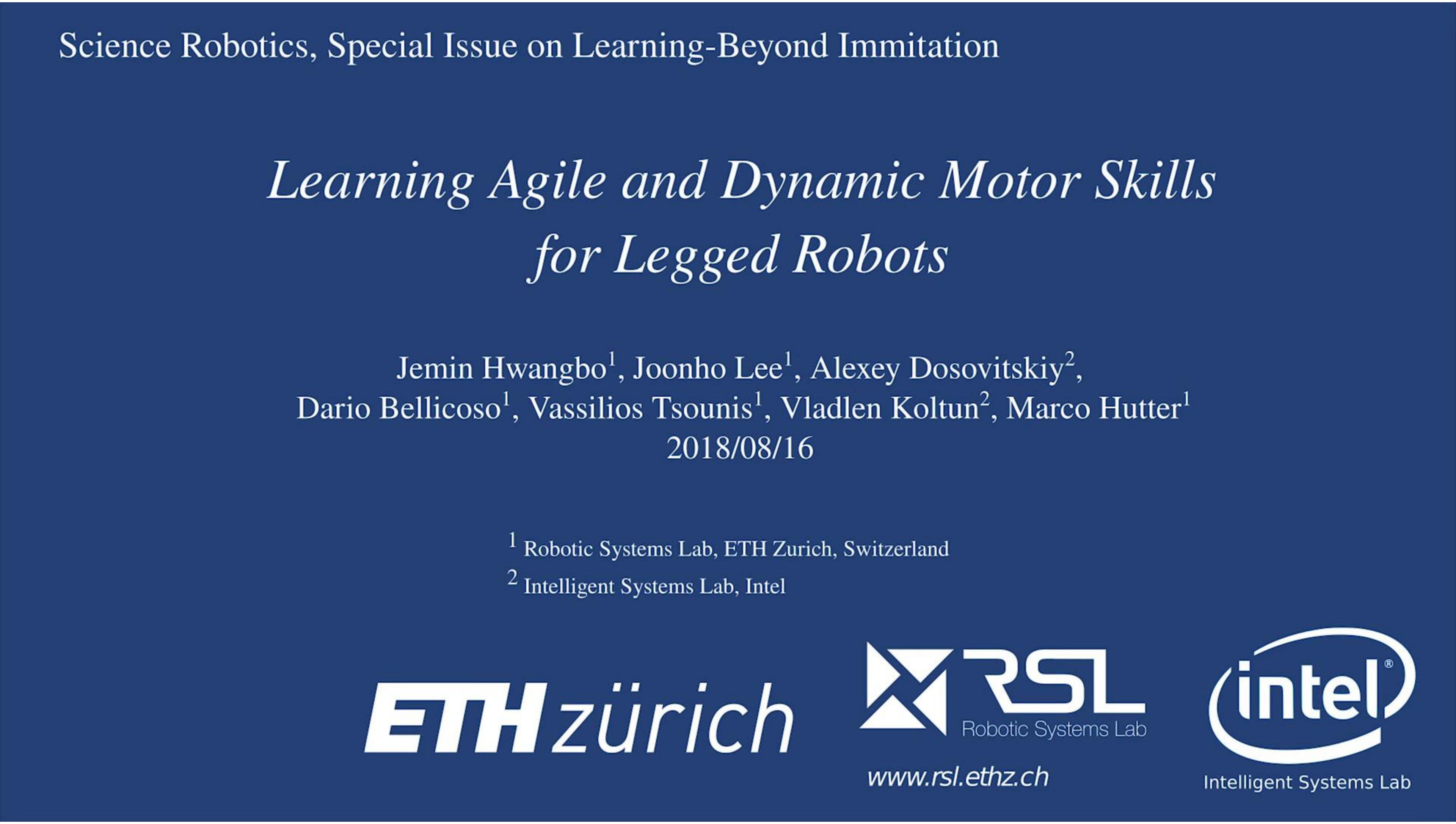


ig. 5. **Training control policies in simulation.** The policy network maps the current observation and the joint state istory to the joint position targets. The actuator network maps the joint state history to the joint torque, which is used n rigid-body simulation. The state of the robot consists of 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.

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, **long iterative process** (often not described in articles)
- Black-box controller, **lack of proof of stability/performance**

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

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,

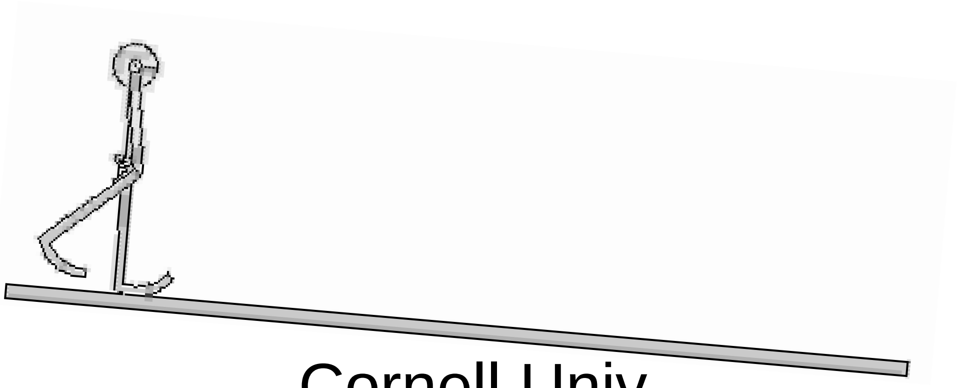
3. CPG-and-reflex based methods

Robotics: passive walkers

- The laws of physics should be exploited: passive walkers

Nice example of mechanical limit cycle behavior

Movie by Jun Nakanishi



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.

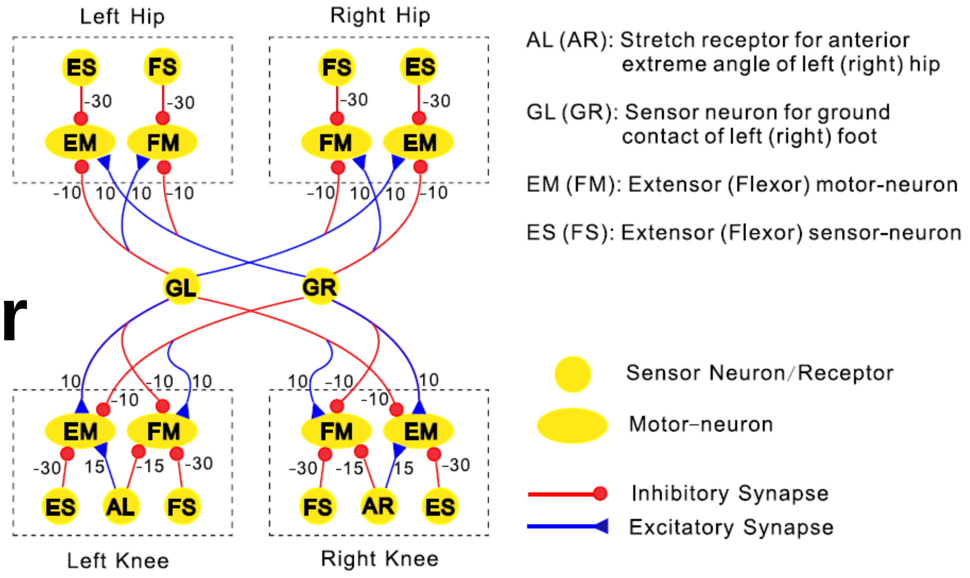
Note: this sensory-driven controller **share similarities with neuromechanical models of human locomotion** (e.g. Geyer and Herr 2010).

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 implemented with a neural network



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:

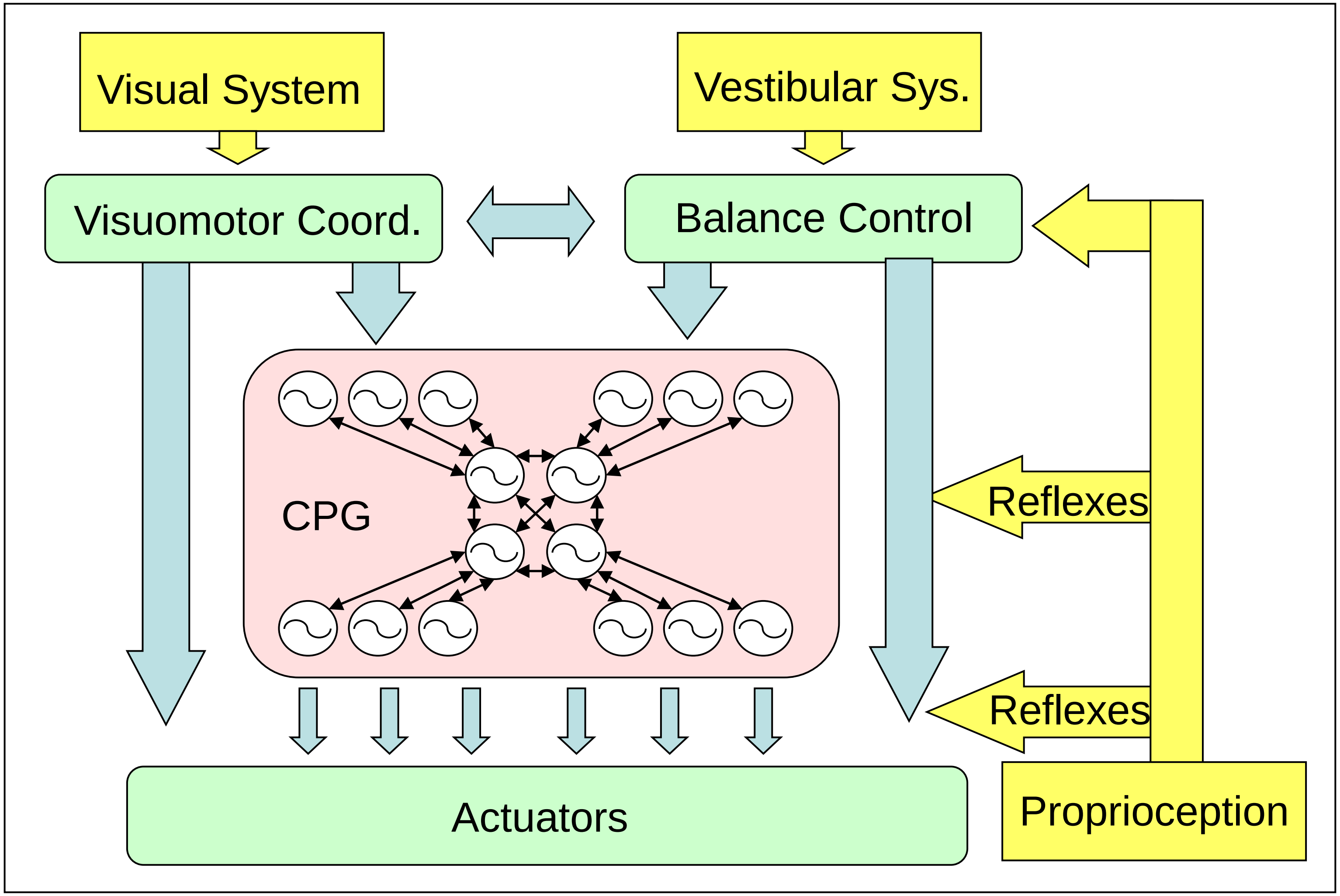
- because of the lack of a centrally generated rhythm, non-negligible **risk that locomotion might be completely stopped** because of damage in the sensors and/or external constraints that force the robot in a particular posture.

Examples of bio-inspired approaches

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

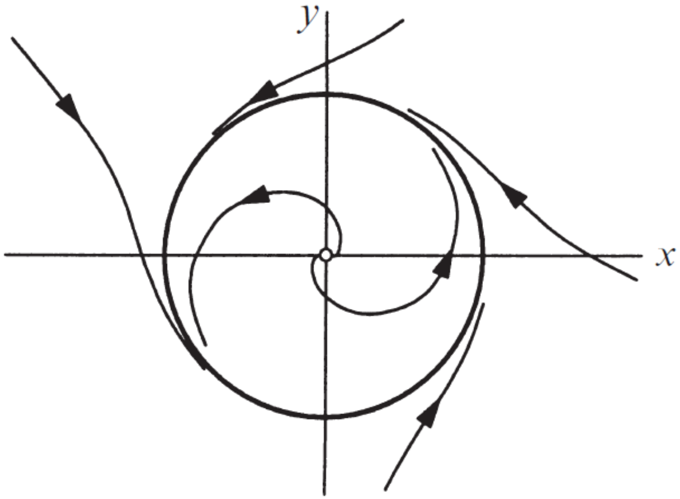
CPG-and-reflex control

- 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:

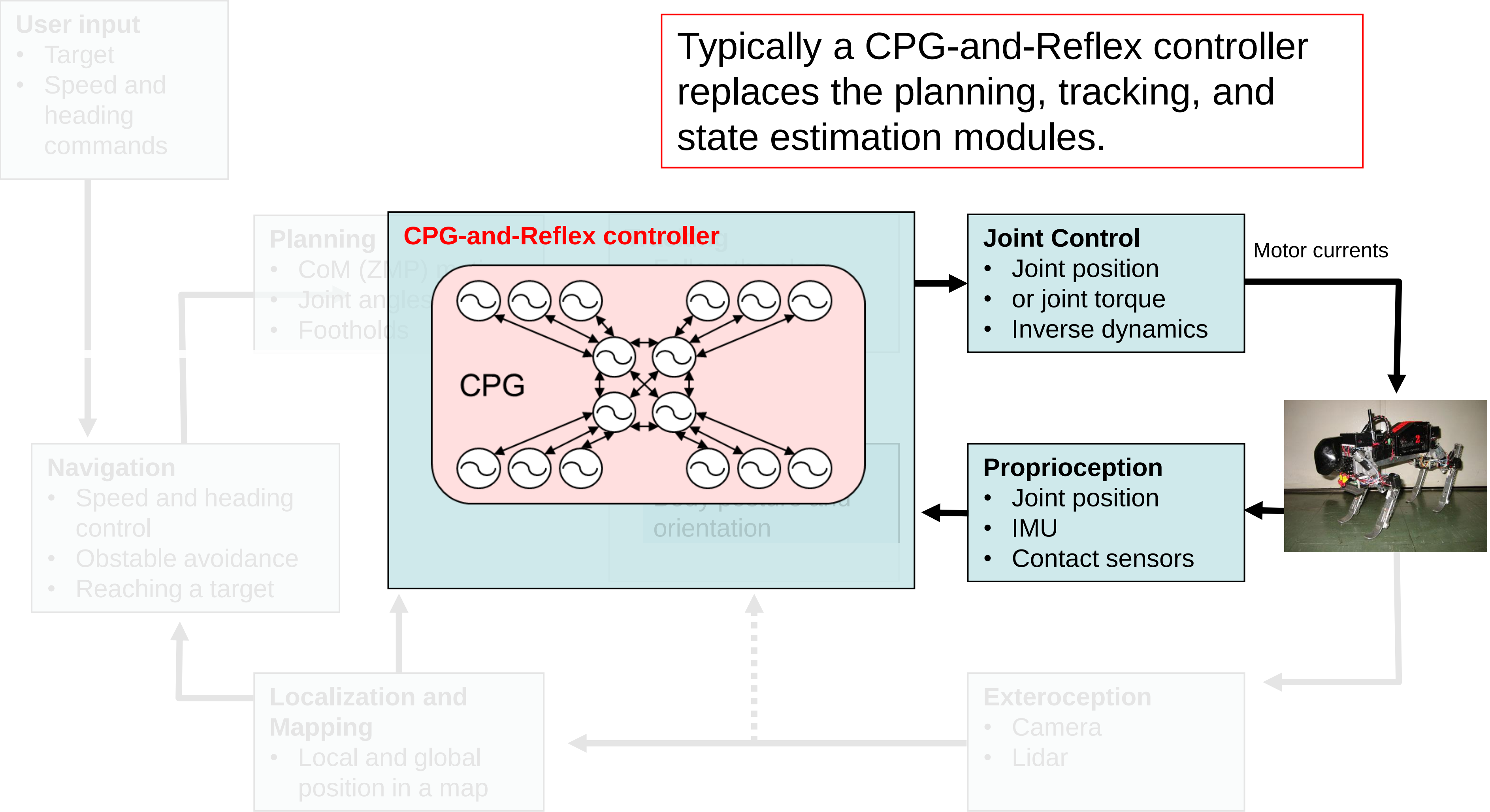


Limit cycles

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

CPG-and-Reflex control

Typically a CPG-and-Reflex controller replaces the planning, tracking, and state estimation modules.



User input
- Target
- Speed and heading commands

Planning
- CoM (ZMP) m
- Joint angles
- Footholds

orientation

Joint Control
- Joint position
- or joint torque
- Inverse dynamics

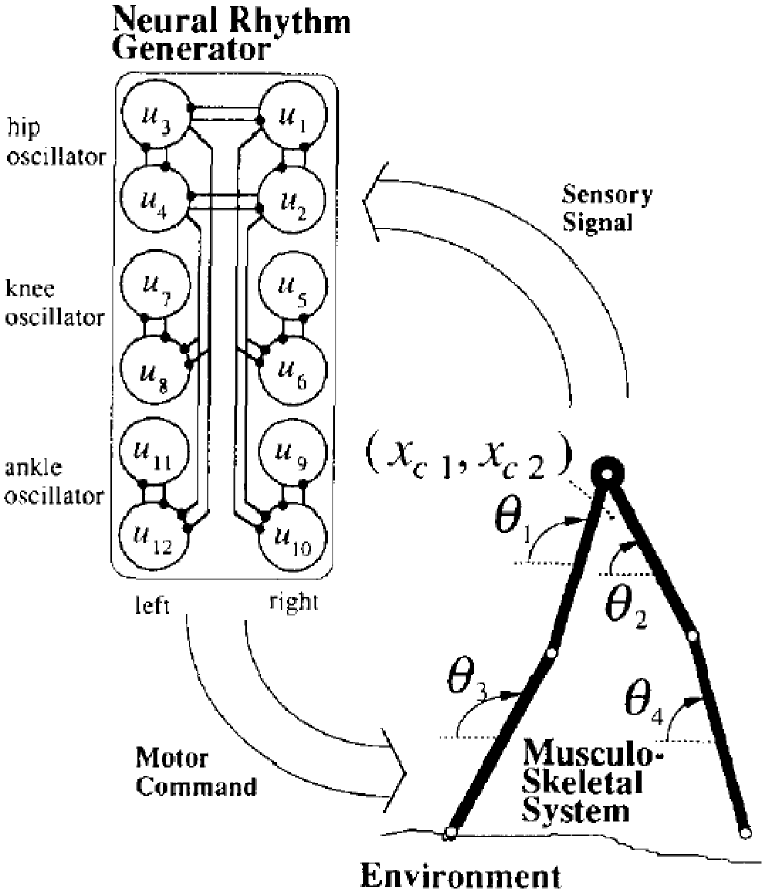
Proprioception
- Joint position
- IMU
- Contact sensors

Navigation
- Speed and heading control
- Obstable avoidance
- Reaching a target

Localization and Mapping
- Local and global position in a map

Exteroception
- Camera
- Lidar

Taga's neuromechanical simulation



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

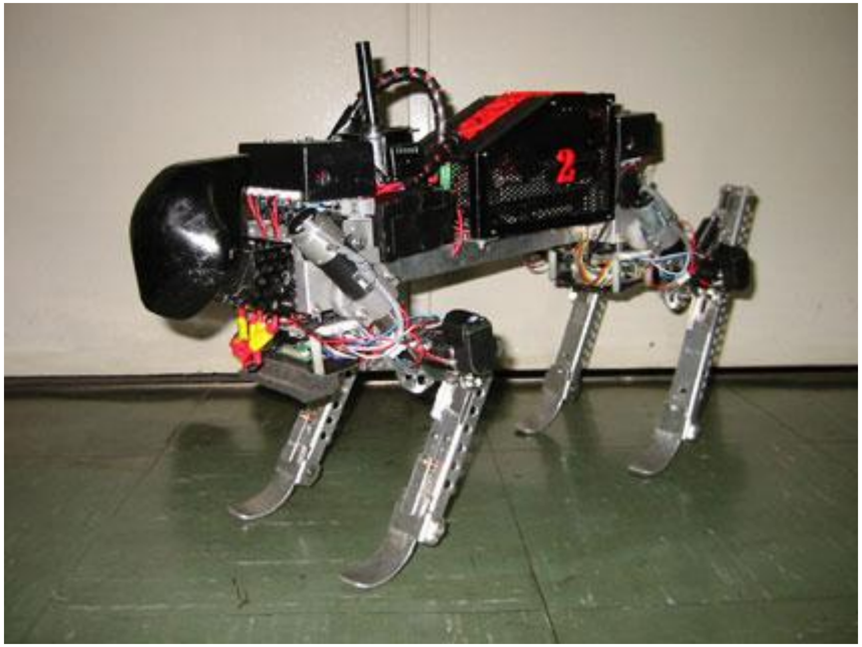
Quite a few labs have taken a similar approach,

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.

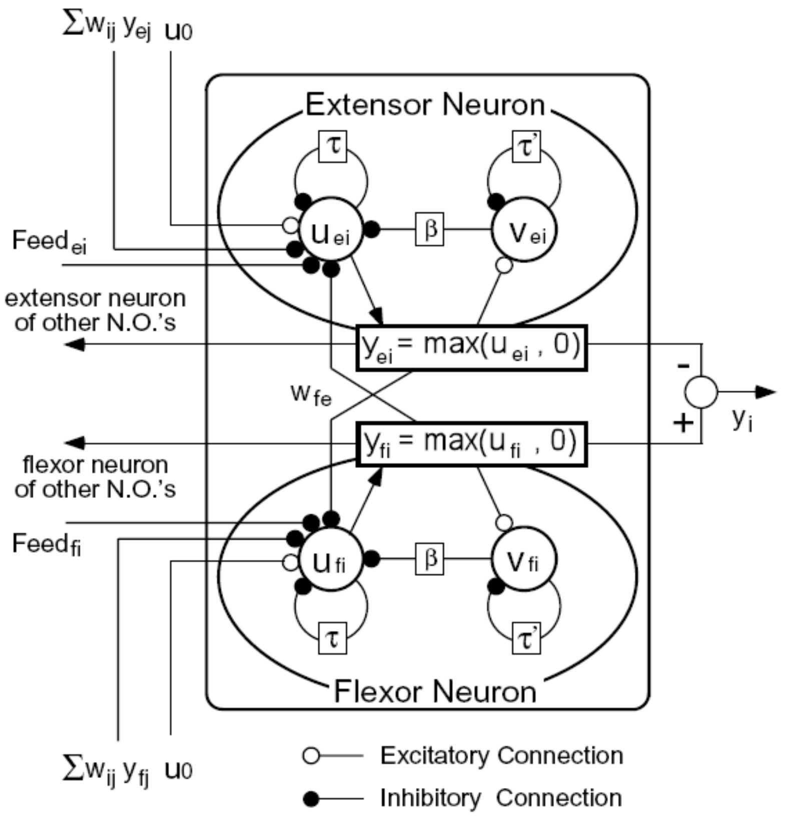


3 actuated DOF per limb

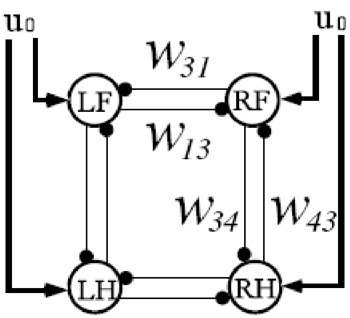
CPG model

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

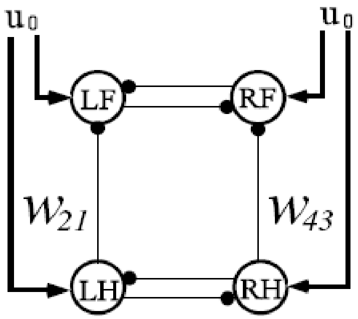
$$\begin{aligned}
\tau \dot{u}_{\{e,f\}i} &= -u_{\{e,f\}i} + w_{fe} y_{\{f,e\}i} - \beta v_{\{e,f\}i} \\
&\quad + u_0 + Feed_{\{e,f\}i} + \sum_{j=1}^{n} w_{ij} y_{\{e,f\}j} \\
y_{\{e,f\}i} &= \max\ (u_{\{e,f\}i}, 0) \\
\tau' \dot{v}_{\{e,f\}i} &= -v_{\{e,f\}i} + y_{\{e,f\}i}.
\end{aligned}$$



CPG architecture

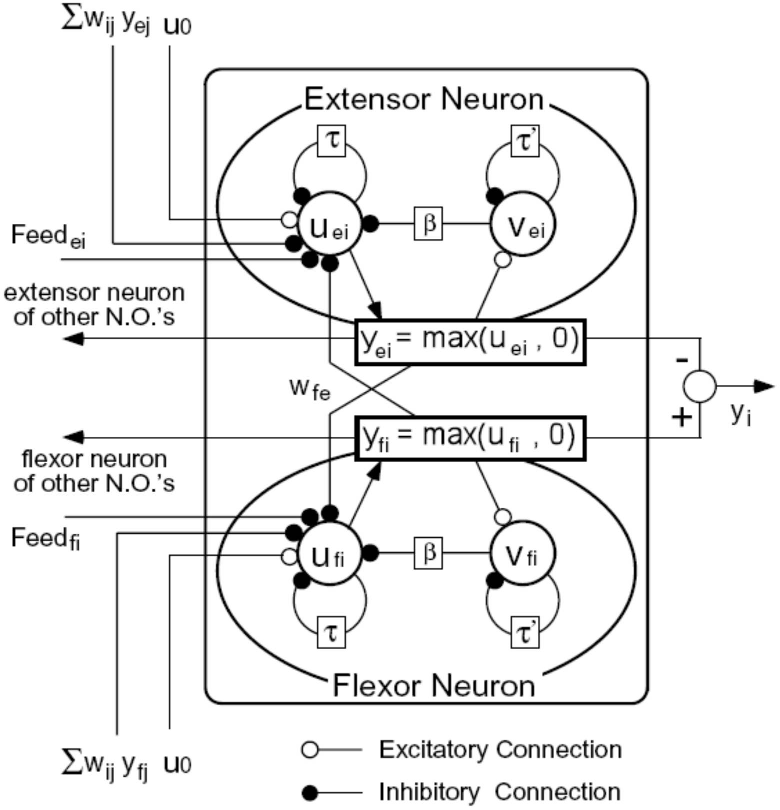


(a)



(b)

Gait transitions by changing connection weights in the network of oscillators

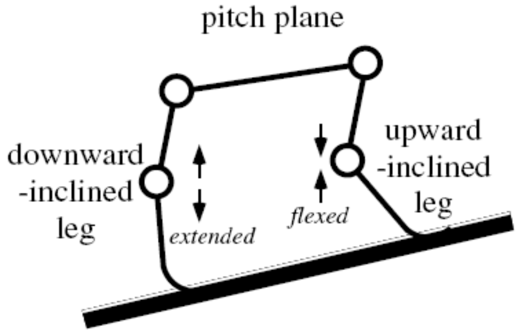


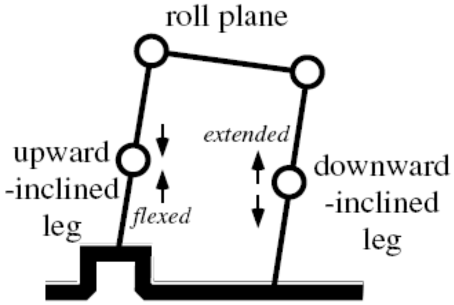
Reflexes:

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





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.

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

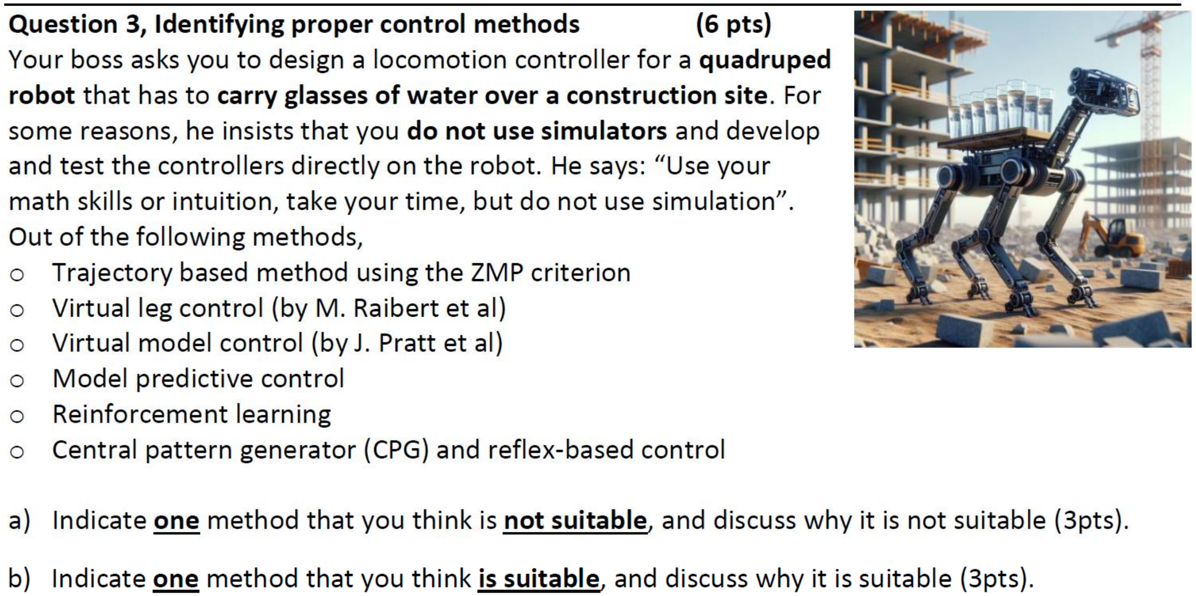
- Cons:
  - Fewer mathematical tools than model-based methods

  - **Not (yet) a clear design methodology**, it is recommended to use reinforcement learning or optimization algorithms

See more examples next week

Possible exam questions

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



Question 3, Identifying proper control methods (6 pts)

Your boss asks you to design a locomotion controller for a **quadruped robot** that has to **carry glasses of water over a construction site**. For some reasons, he insists that you **do not use simulators** and develop and test the controllers directly on the robot. He says: "Use your math skills or intuition, take your time, but do not use simulation".

Out of the following methods,

- Trajectory based method using the ZMP criterion
- Virtual leg control (by M. Raibert et al)
- Virtual model control (by J. Pratt et al)
- Model predictive control
- Reinforcement learning
- Central pattern generator (CPG) and reflex-based control

a) Indicate **one** method that you think is **not suitable**, and discuss why it is not suitable (3pts).

b) Indicate **one** method that you think **is suitable**, and discuss why it is suitable (3pts).

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

Check out the articles

Finalize student teams by Sept 29