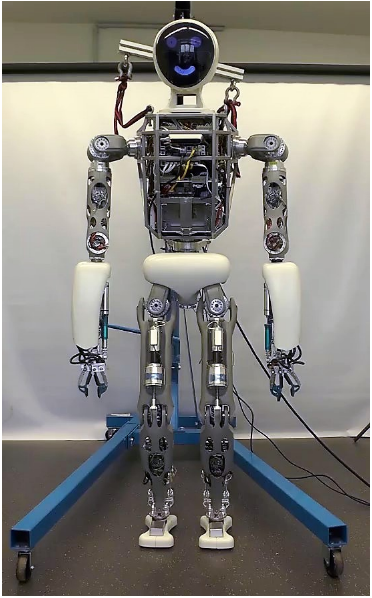


Deutsches Forschungszentrum für Künstliche Intelligenz
German Research Center for Artificial Intelligence

Universität Bremen

Deriving Rewards for Reinforcement Learning from Symbolic Behaviour Descriptions of Bipedal Walking

Daniel Harnack, **Christoph Lüth**, Lukas Gross, Shivesh Kumar, Frank Kirchner

`christoph.lueth@dfki.de`

CDC 2023, Singapore, 13.12.2023.

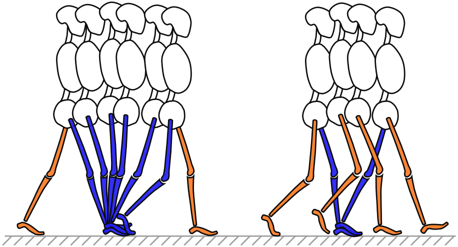
Bundesministerium für Bildung und Forschung

Research supported grant no. 01IW20004.

Motivation

- Humans learn not only by **doing**, but also by being taught on a symbolic level.
- Symbolic instruction **reduces state space** so subsymbolic learning can be more efficient.
- **Example**: Learning to ski.

Human Walking

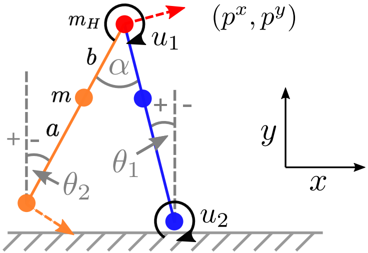


Picture adapted from [1]

- Combines different phases described **informally**[1]:
  - “*The swing foot lifts until the body weight is aligned over the forefoot of the stance leg.*”
  - “*The second phase begins as the swinging limb is opposite of the stance limb. The phase ends when the swinging limb is forward and the tibia is vertical, i.e. hip and knee flexion postures are equal.*”
- Phases characterized by relations of body parts, described **formally** as partitions of state space.
- Gait sequence characterised by traversed partitions.

[1] J. Perry: *Gait Analysis — Normal and Pathological Function*. Thorofare, NJ. SLACK Inc, 1992.

The Compass Walker



The compass walker: stance leg, swing leg.

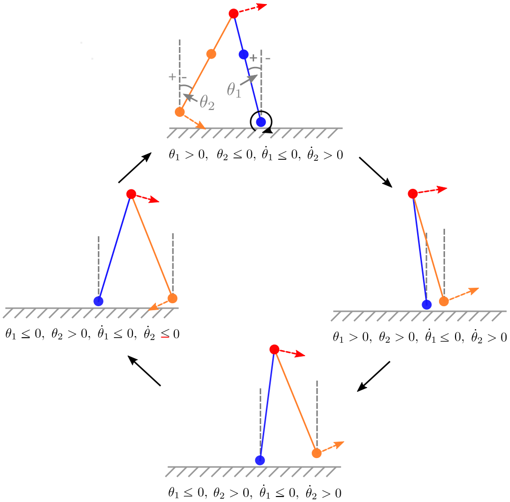
Dynamics:

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{g}(\boldsymbol{\theta}) = \mathbf{S}\boldsymbol{u}$$

with

$\boldsymbol{\theta} = [\theta_1, \theta_2]^T$	system configuration
$\boldsymbol{u} = [u_1, u_2]^T$	torque
$\boldsymbol{M}$	inertia
$\boldsymbol{C}$	Coriolis force
$\boldsymbol{g}$	gravity
$\boldsymbol{S}$	control

Different Phases of Walking



Sixteen **orthants**

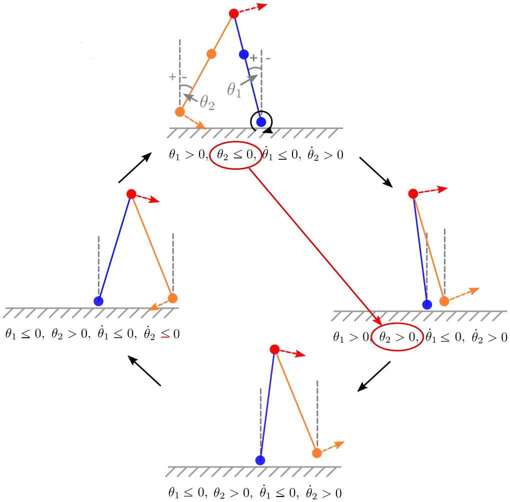
$$\mathcal{O}_{1,\ldots,16}$$

given by

$$\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2$$

θ_1 stance leg,
θ_2 swing leg

Different Phases of Walking



Sixteen **orthants**

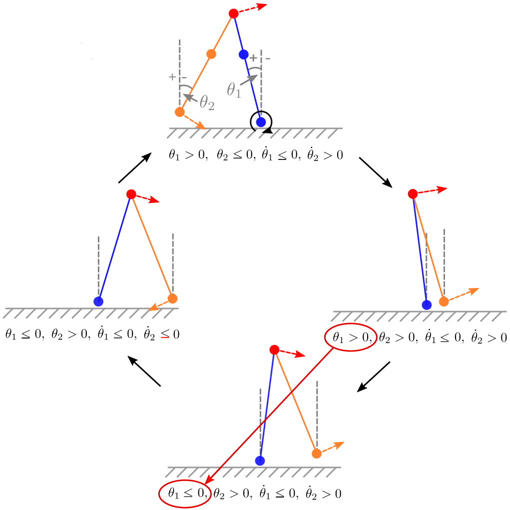
$$\mathcal{O}_{1,\ldots,16}$$

given by

$$\theta_1, \theta_2, \dot\theta_1, \dot\theta_2$$

θ_1 stance leg,
θ_2 swing leg

Different Phases of Walking



Sixteen **orthants**

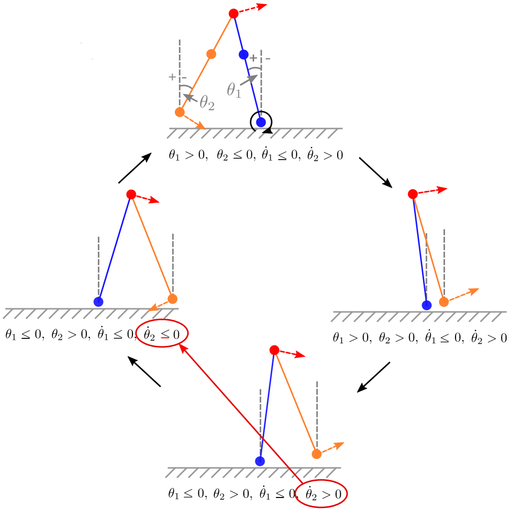
$$\mathcal{O}_{1,\ldots,16}$$

given by

$$\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2$$

θ_1 stance leg,
θ_2 swing leg

Different Phases of Walking



Sixteen **orthants**

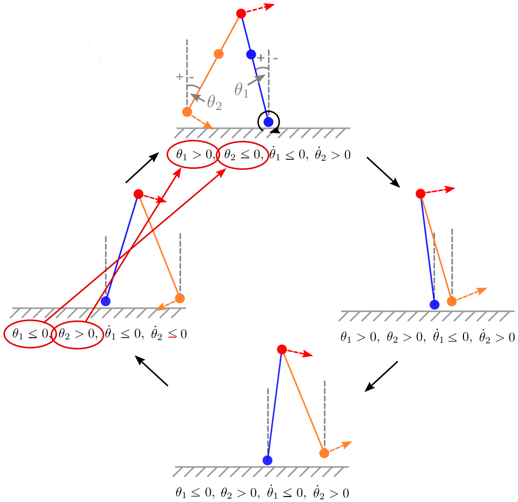
$$\mathcal{O}_{1,\ldots,16}$$

given by

$$\theta_1, \theta_2, \dot\theta_1, \dot\theta_2$$

θ_1 stance leg,
θ_2 swing leg

Different Phases of Walking



Sixteen **orthants**

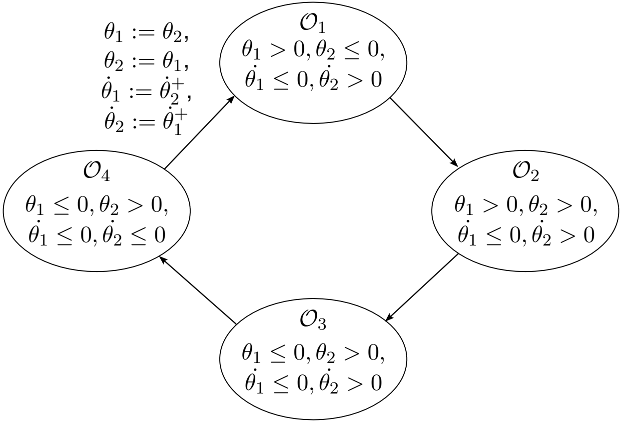
$$\mathcal{O}_{1,\ldots,16}$$

given by

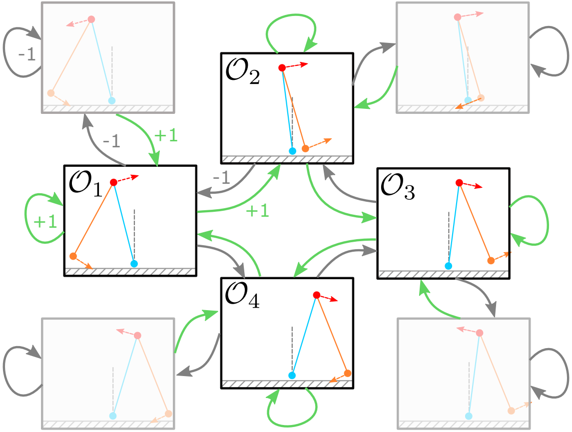
$$\theta_1, \theta_2, \dot\theta_1, \dot\theta_2$$

θ_1 stance leg,
θ_2 swing leg

Mathematical Model: Hybrid Automaton



Partitioning the State Space: Orthant Sequences



Reward Formulation

Reward function for **orthants**:

$$r_{\mathrm{or}}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) = \begin{cases} +1 & \text{if} \quad \mathcal{O}(\boldsymbol{x}_{t-1}) \in Q \wedge \\ & \quad\quad \mathcal{O}(\boldsymbol{x}_t) \in Q \wedge \\ & \quad\quad (\mathcal{O}(\boldsymbol{x}_{t-1}), \mathcal{O}(\boldsymbol{x}_t)) \in E \\ +1 & \text{if} \quad \mathcal{O}(\boldsymbol{x}_{t-1}) \notin Q \wedge \\ & \quad\quad \mathcal{O}(\boldsymbol{x}_t) \in Q \\ -1 & \text{else} \end{cases}$$

Comparison: reward for **distance travelled**

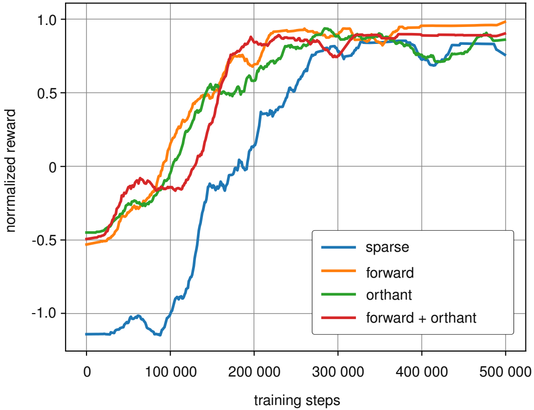
$$r_{\mathrm{for}}(\boldsymbol{p}_t, \boldsymbol{p}_{t-1}) = 2H(p_t^x - p_{t-1}^x) - 1$$

Further reward for smooth control, penalty for falling over.

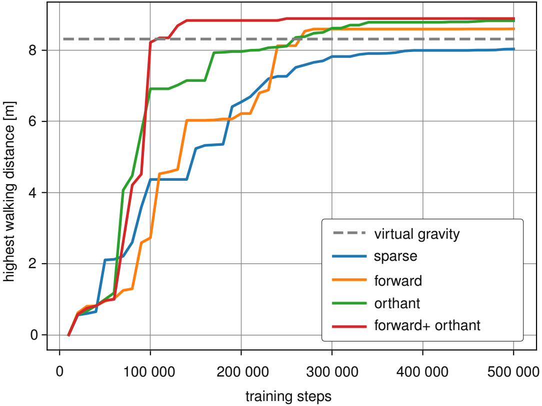
Training setup:

- Training a policy $\phi(\boldsymbol{x}_t) \to \boldsymbol{u}_t$, training algorithm PPO
- **Baseline**: virtual gravity controller (slope of $\phi = -0.07$ rad)

Evaluation Results

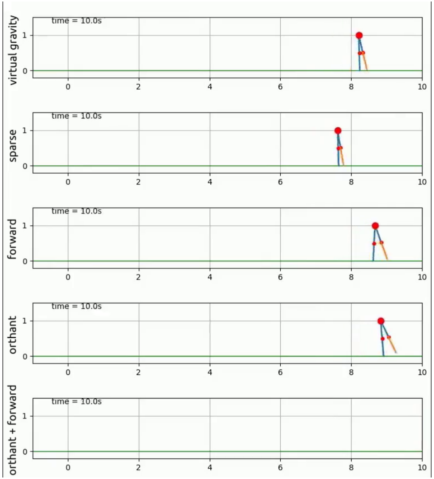


Normalized rewards



Highest walking distance for $t = 10s$

Evaluation Results: The movie



Conclusions

- **Combination** of forward and orthants achieves fastest convergence and highest distance.
  - Combination seems redundant — **optimal combination** of rewards?
- Deriving a reward function in **three easy steps**:

1. Derive hybrid automaton from informal description
2. Hybrid automaton restricts state space (here: orthants)
3. Restriction gives reward function (here: r_{or})

- Approach **combines well**.
- Approach applicable to **other problems** as well.