3.2 'Experience-Based Adaptation of Locomotion Behaviors for Kinematically Complex Robots in Unstructured Terrain' (LM-T-02)

Alexander Dettmann[(1)], *Anna Born*[(1)], *Sebastian Bartsch*[(2)], *and Frank Kirchner*[(1) (2)]

(1) Arbeitsgruppe Robotik, Universität Bremen, Robert-Hooke-Straße 1, 28359 Bremen, Germany
(2) Robotics Innovation Center, DFKI GmbH, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: `alexander.dettmann@dfki.de`

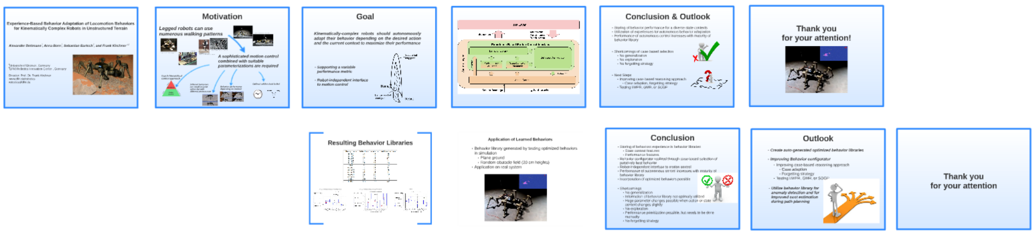
Abstract

Kinematically complex robots such as legged robots provide a large degree of mobility and flexibility, but demand a sophisticated motion control, which has more tunable parameters than a general planning and decision layer should take into consideration. A lot of parameterizations exist which produce locomotion behaviors that fulfill the desired action but with varying performance, e.g., stability or efficiency. In addition, the performance of a locomotion behavior at any given time is highly depending on the current environmental context. Consequently, a complex mapping is required that closes the gap between robot-independent actions and robot-specific control parameters considering the environmental context and a given prioritization of performance indices.

In the proposed approach, the robot learns from experiences made during its interaction with the environment. A knowledge base is created which links locomotion behaviors with performance features for visited contexts. This *behavior library* is utilized by a case-based reasoner to select motion control parameters for a desired action within the current context. The paper provides an overview of the control approach, the algorithms used to determine the current context and the robot's performance, as well as a description of the reasoner which selects appropriate locomotion behaviors.

In experiments, different *behavior libraries* were automatically built when operators had to control a walking robot manually through obstacle courses. Afterwards, the collected experiences and a trajectory follower were used to traverse an obstacle course autonomously. The provided experimental evaluation shows the performance dependency of the autonomous control with respect to different sizes and qualities of utilized *behavior libraries* and compares it to manual control.

Please note that the corresponding paper is published in:
Experience-based adaptation of locomotion behaviors for kinematically complex robots in unstructured terrain; A. Dettmann, A. Born, S. Bartsch, and F. Kirchner; In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015.

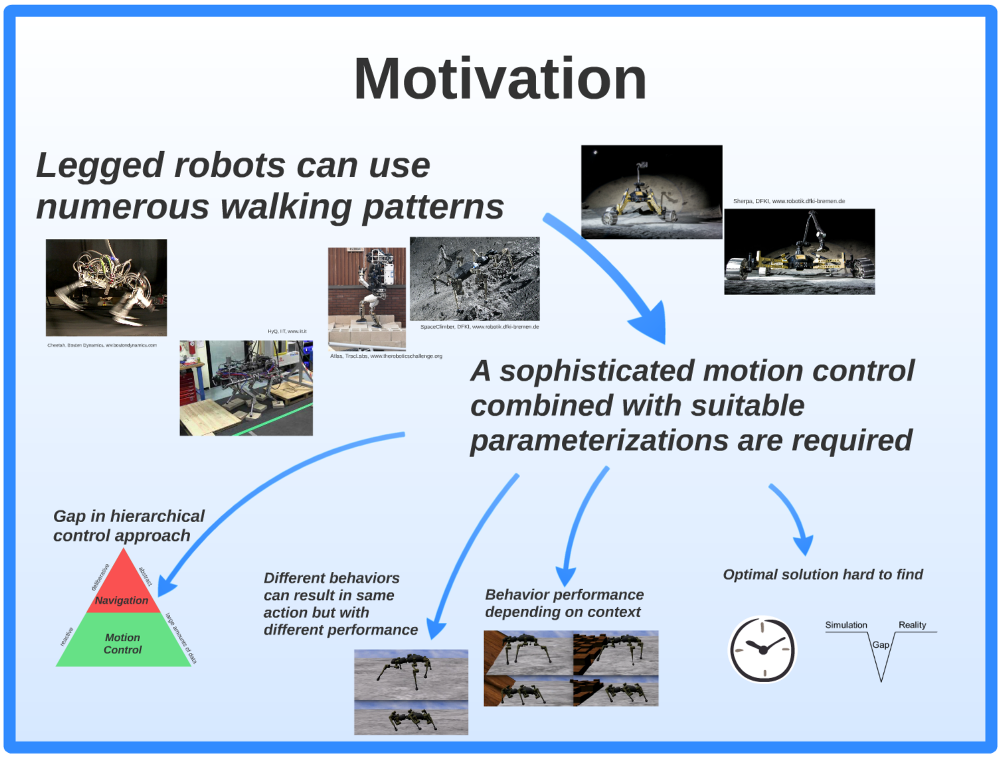


Experience-Based Behavior Adaptation of Locomotion Behaviors for Kinematically Complex Robots in Unstructured Terrain

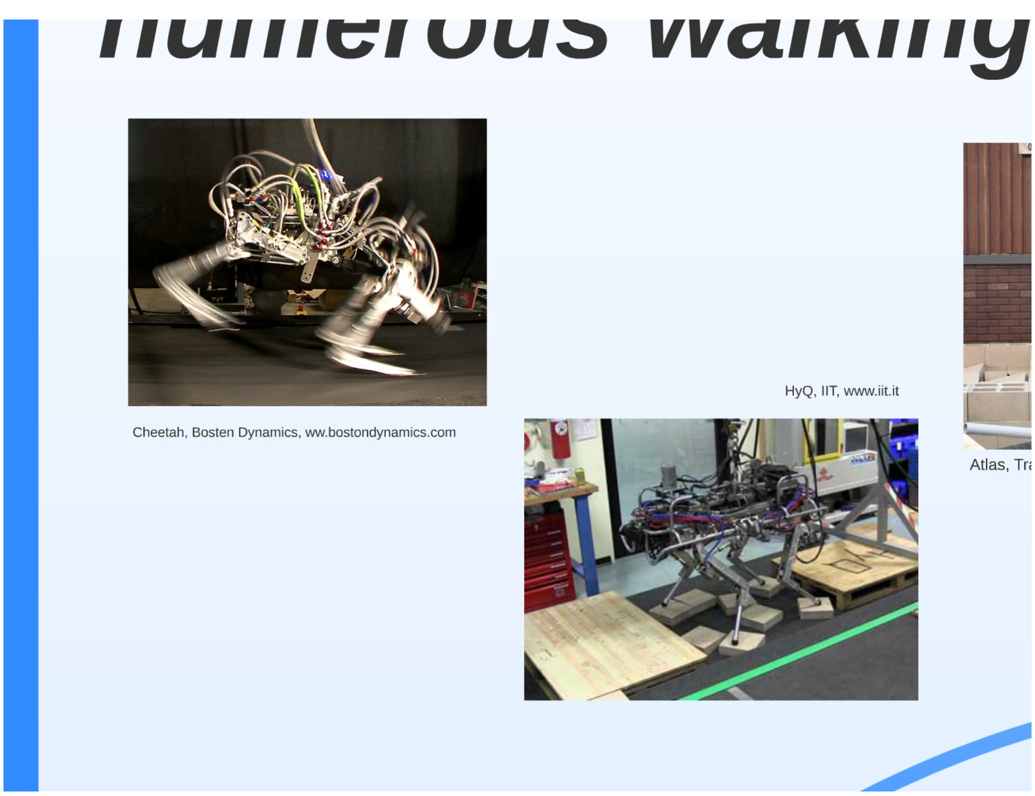
***Alexander Dettmann*[1]*, Anna Born*[1]*, Sebastian Bartsch*[2]*, and Frank Kirchner*[1,2]**

[1]University of Bremen, Germany
[2]DFKI Robotics Innovation Center , Germany

Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de

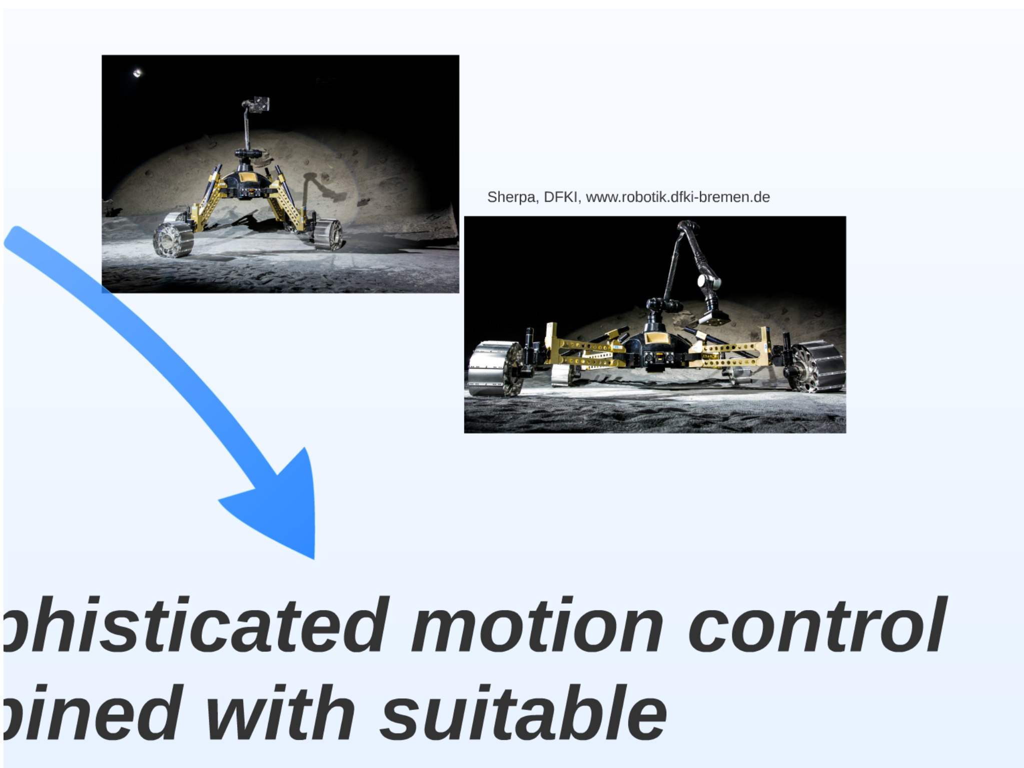


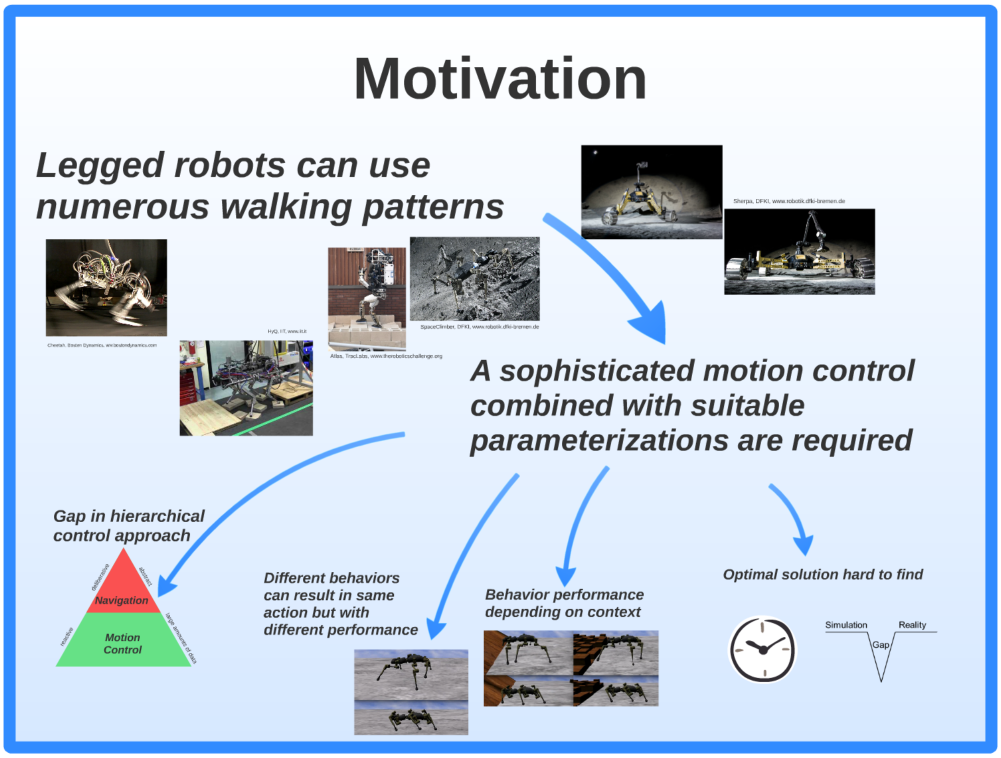
Motivation
Legged robots can use numerous walking patterns
A sophisticated motion control combined with suitable parameterizations are required
Gap in hierarchical control approach
Navigation
Motion Control
Different behaviors can result in same action but with different performance
Behavior performance depending on context
Optimal solution hard to find
Simulation
Reality
Gap

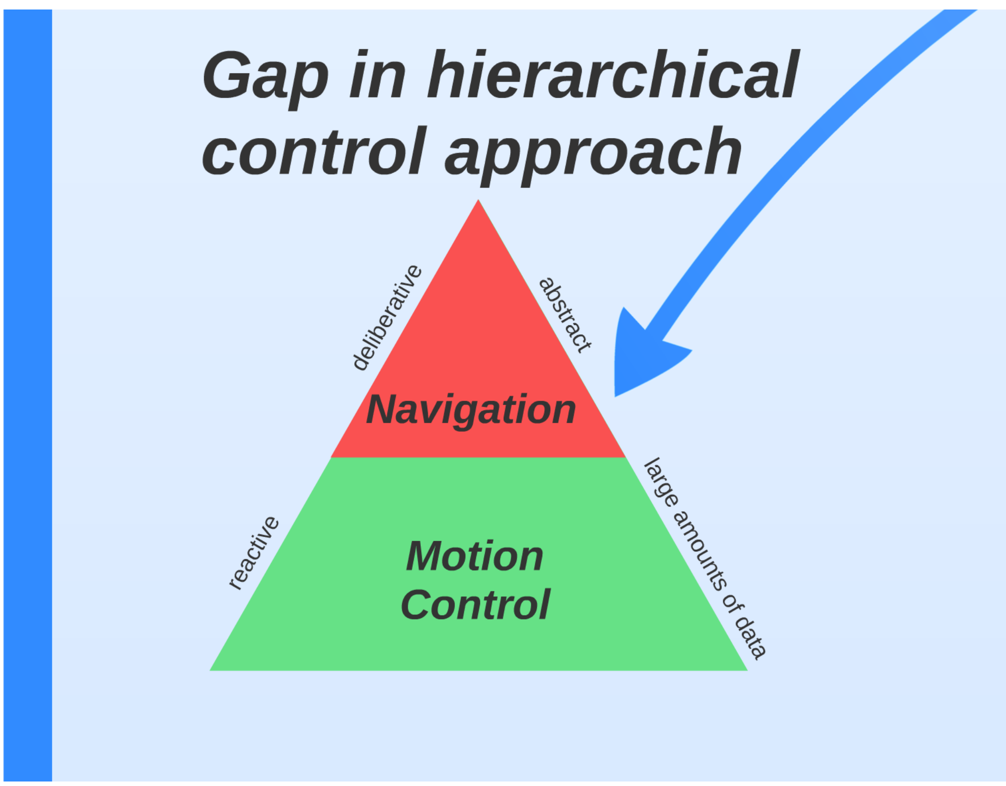


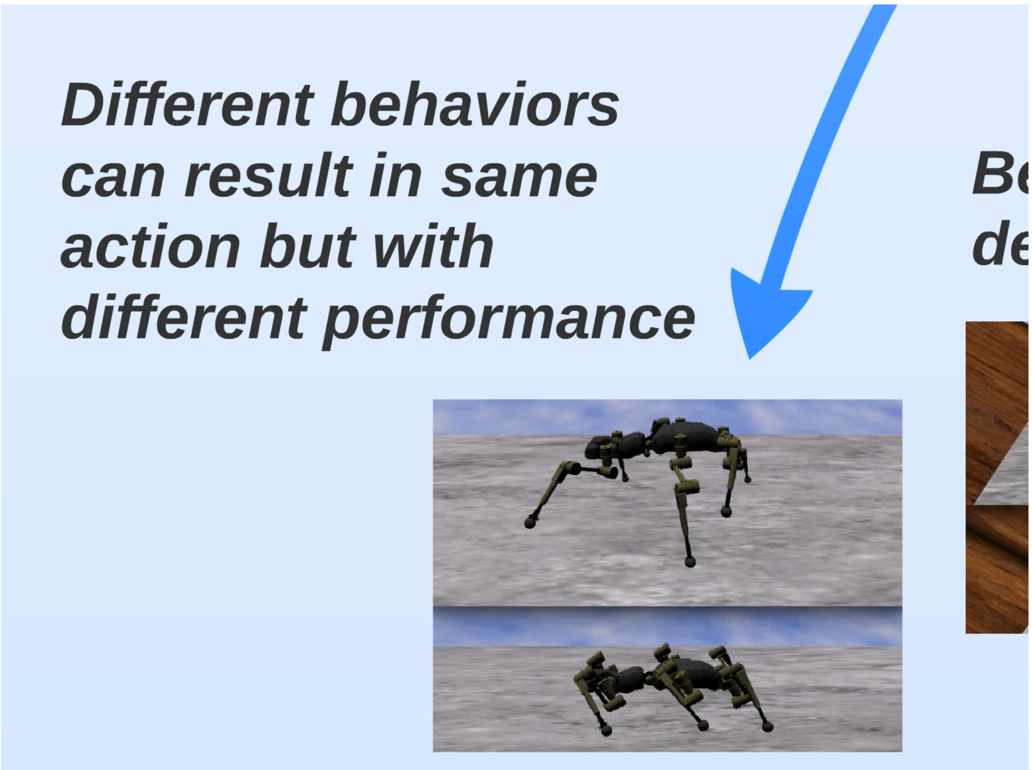
Cheetah, Bosten Dynamics, ww.bostondynamics.com
HyQ, IIT, www.iit.it

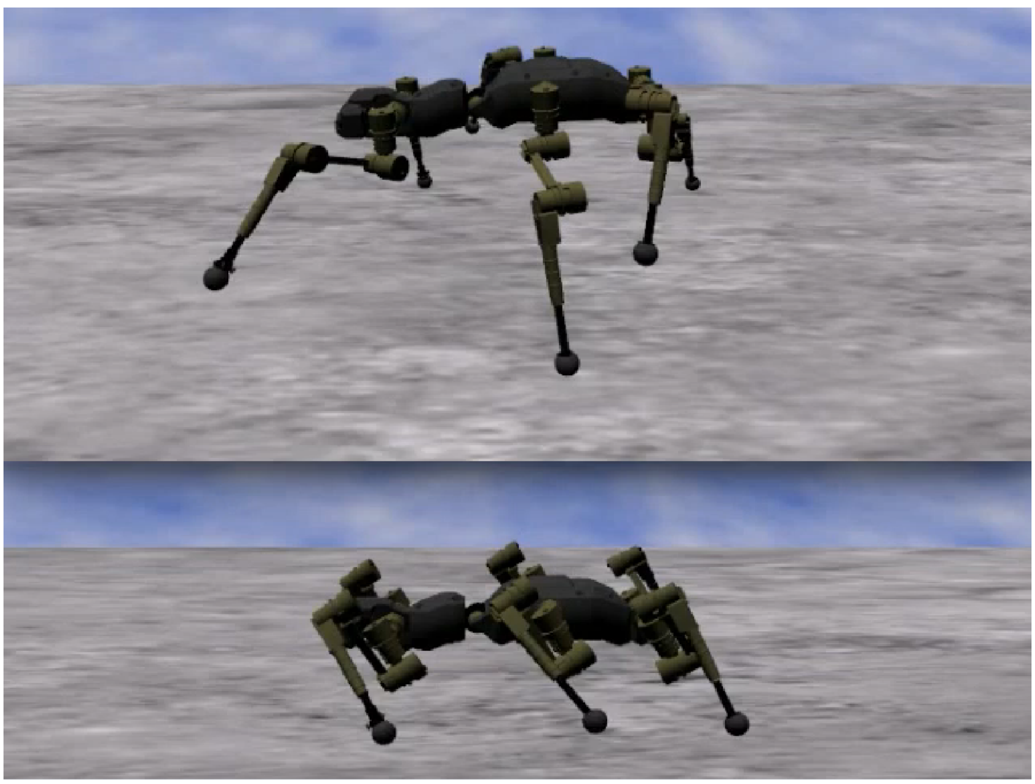

king patterns
, www.iit.it
SpaceClimber, DFKI, www.robotik.dfki-bremen.de
Atlas, TracLabs, www.theroboticschallenge.org
A soph
combin

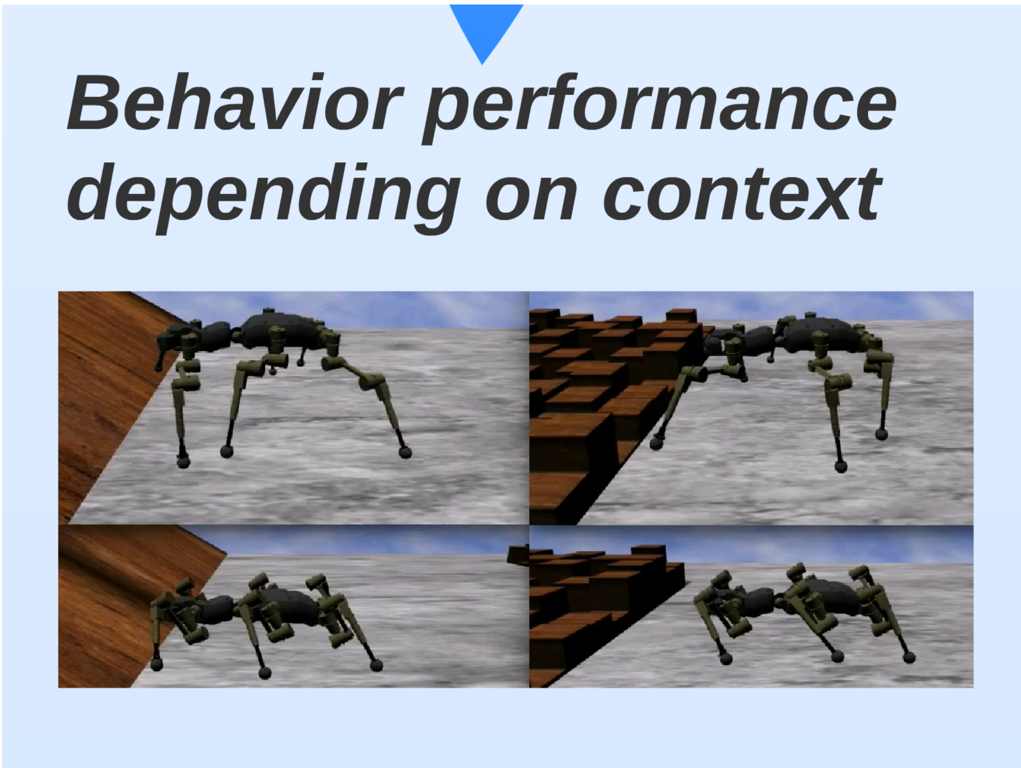

Sherpa, DFKI, www.robotik.dfki-bremen.de
phisticated motion control
bined with suitable

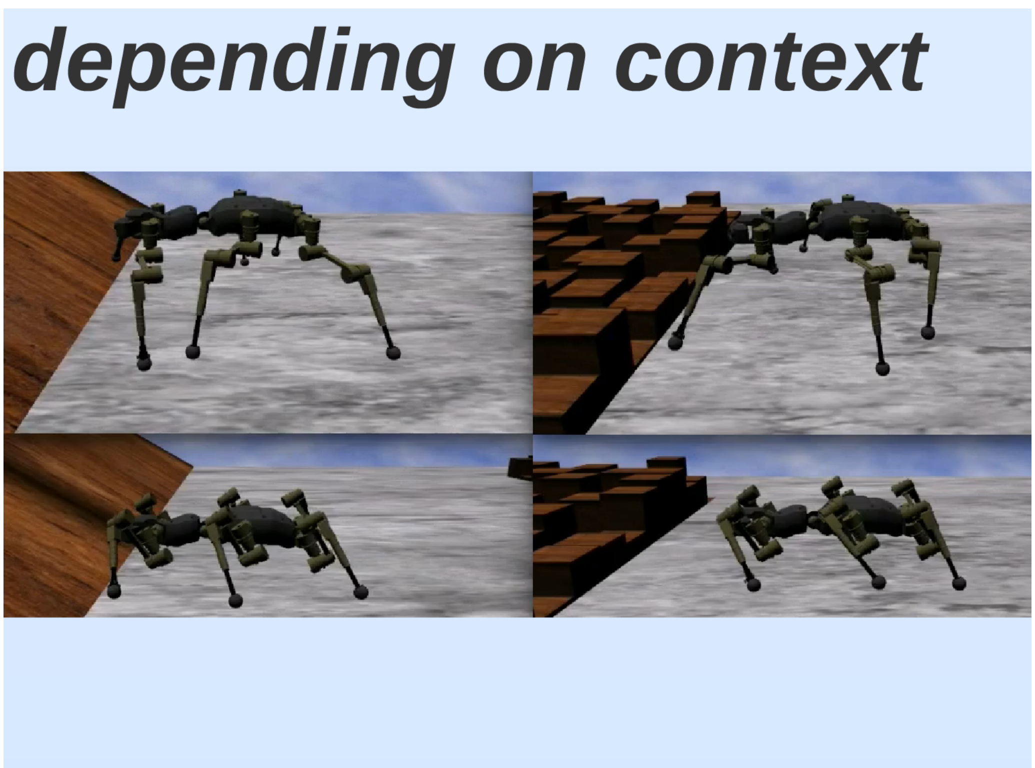

Motivation
Legged robots can use numerous walking patterns
A sophisticated motion control combined with suitable parameterizations are required
Gap in hierarchical control approach
Navigation
Motion Control
Different behaviors can result in same action but with different performance
Behavior performance depending on context
Optimal solution hard to find
Simulation
Reality
Gap

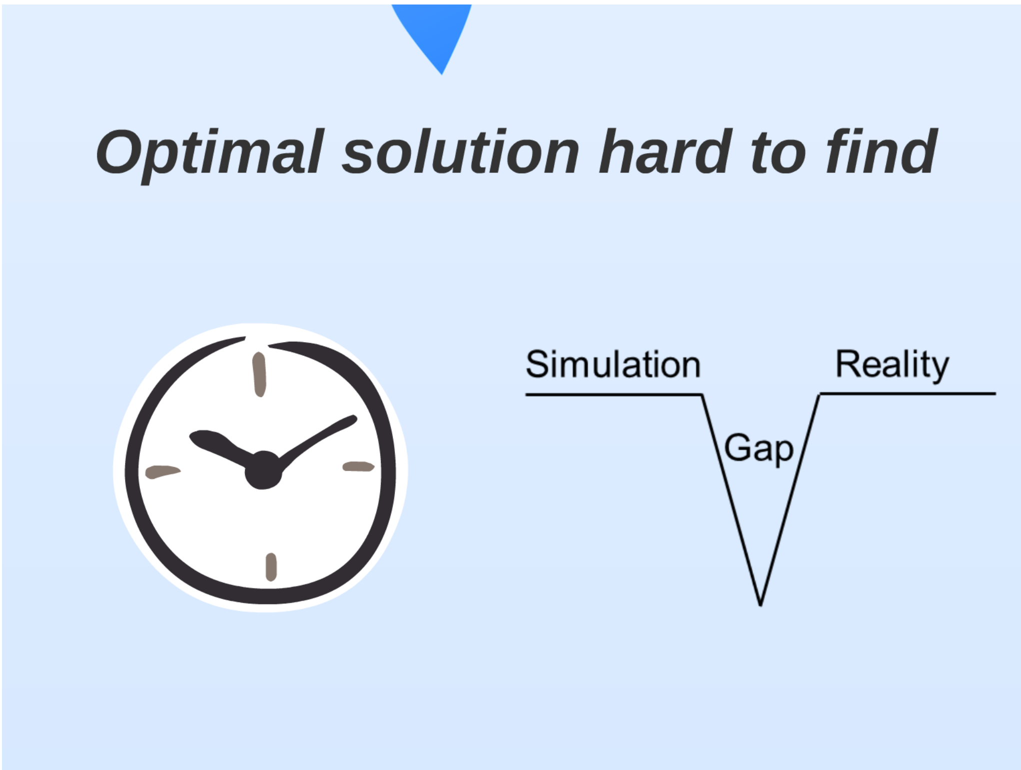

Gap in hierarchical control approach
deliberative
abstract
Navigation
reactive
large amounts of data
Motion Control

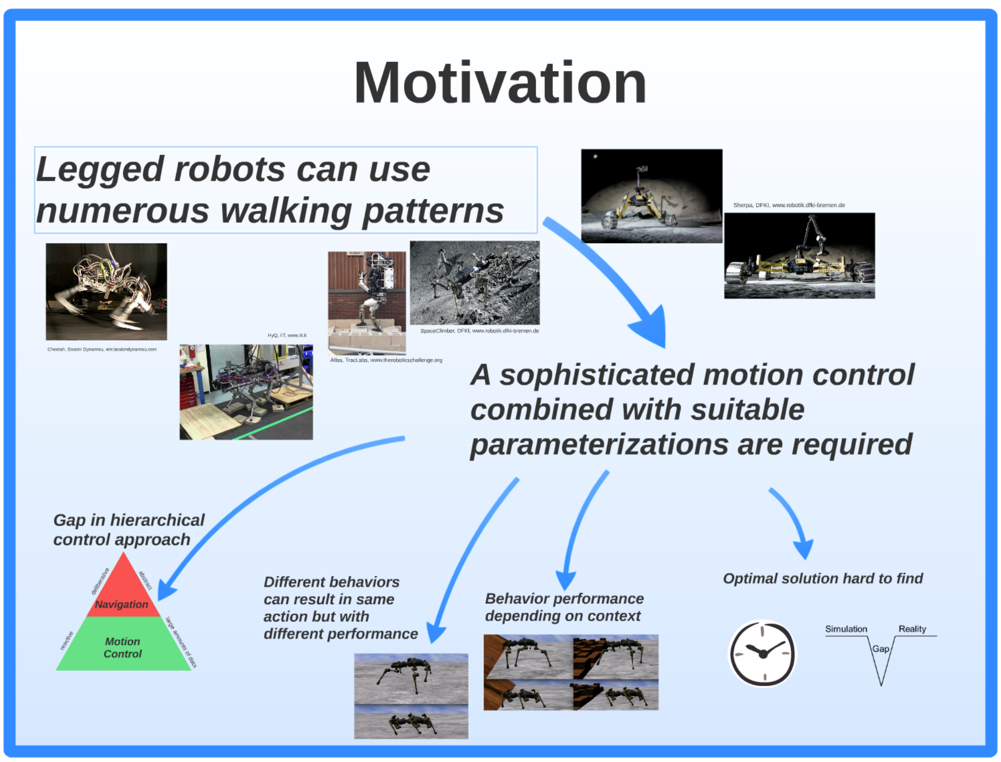

Different behaviors
can result in same
action but with
different performance




Behavior performance
depending on context


depending on context

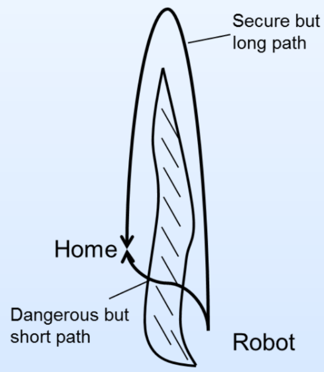

Optimal solution hard to find
Simulation
Reality
Gap

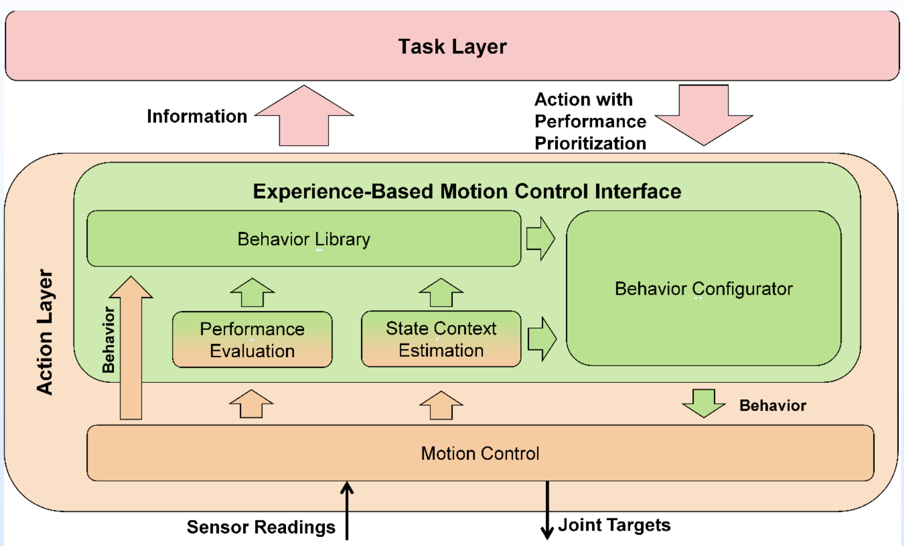

Motivation
Legged robots can use numerous walking patterns
A sophisticated motion control combined with suitable parameterizations are required
Gap in hierarchical control approach
Navigation
Motion Control
Different behaviors can result in same action but with different performance
Behavior performance depending on context
Optimal solution hard to find
Simulation
Reality
Gap

Goal

Kinematically-complex robots should autonomously adapt their behavior depending on the desired action and the current context to maximize their performance

- ***Supporting a variable performance metric***
- ***Robot-independent interface to motion control***

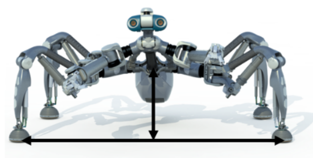




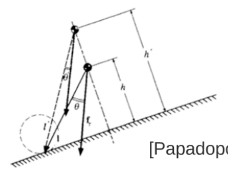
Performance Estimation

Performance features characterize locomotion behaviors

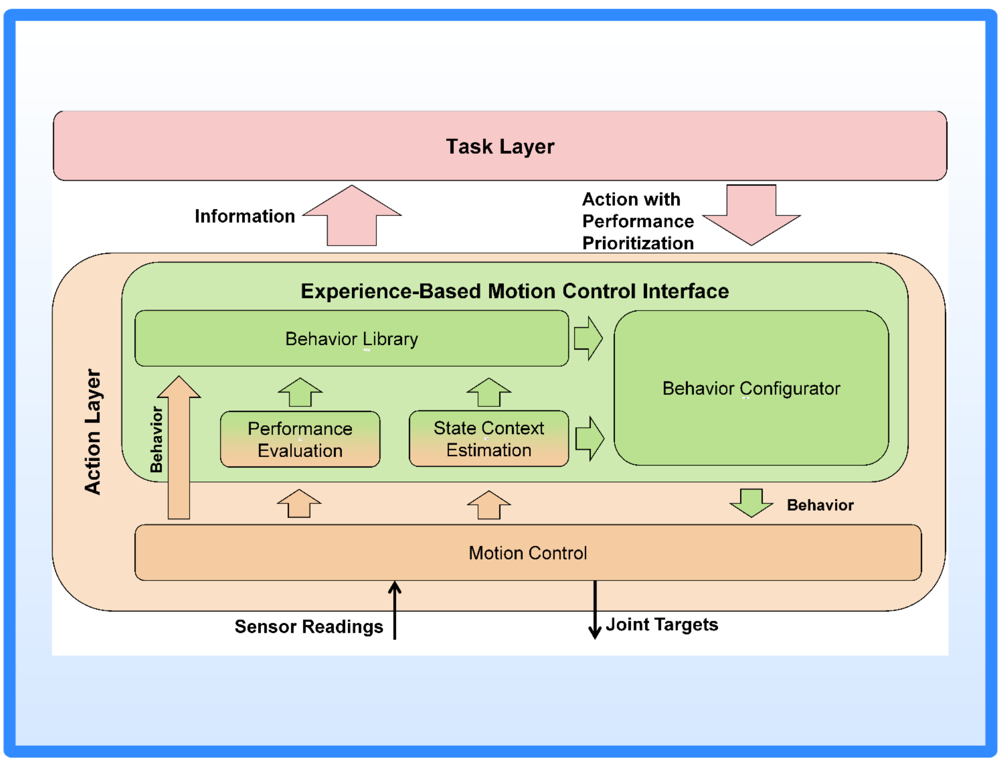
- Action performance features
  - Characterizing movement
    - Velocity x
    - Velocity y
    - Turn rate
  - Characterizing posture
    - Body height
    - Body width



- Meta performance features
  - Characterizing stability
    - Static stability margin (ssm)
    - Force-angle stability measure (dsa)
  - Characterizing efficiency
    - Power
    - Energy per distance (epd)
    - Body vibration



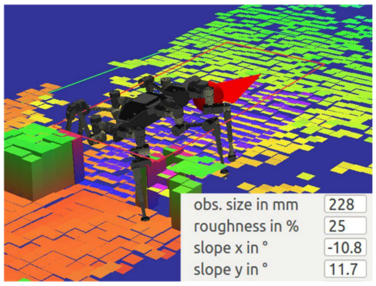
[Papadopoulos2000]

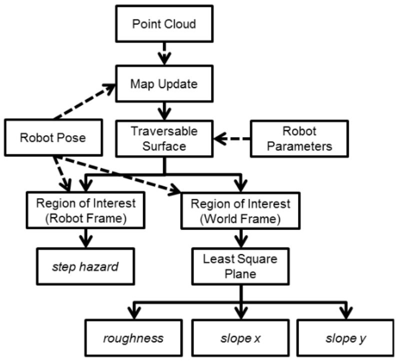


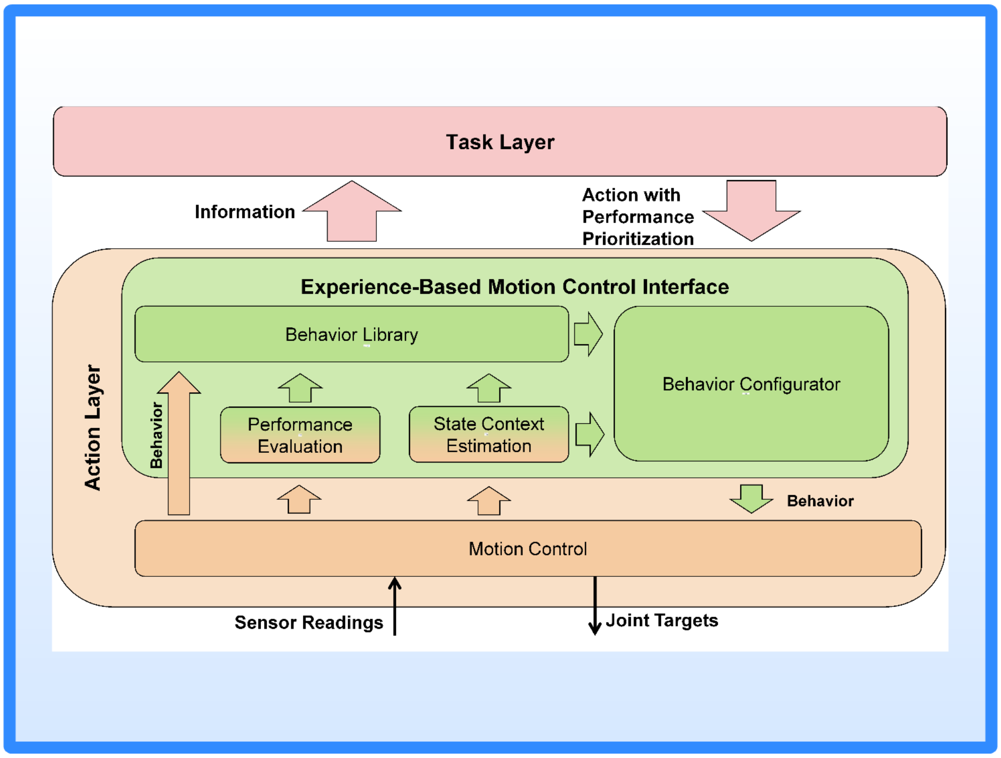
State Context Estimation

State context features characterize the environment

- MLS map from point cloud data and robot pose
- Region of interest
  - Area beneath robot
  - Area in direction of movement within next step cycle
- Max step height, roughness, slope x, slope y

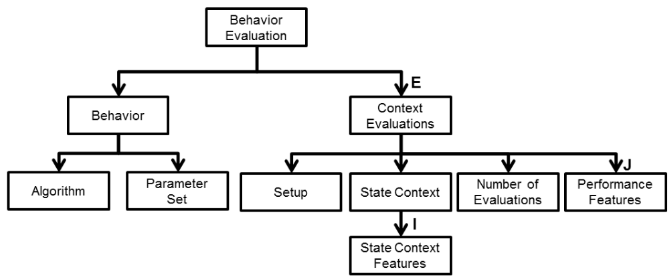






Behavior Library

= Knowledge base of robot

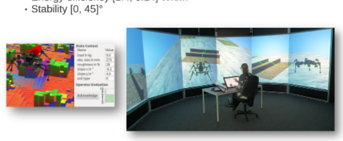


- Behavior experience update
  - Initiated when behavior was constant during evaluation period
  - State context and performance features are averaged and linked to a behavior

Generating Behavior Libraries

Methodology

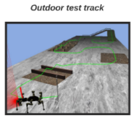
- Learning from demonstration by collecting experiences while operators control SpaceClimber through obstacle courses
- Evaluation of operator skills
  - Path following capability [0.5, 0.1] m
  - Energy efficiency [1.4, 0.14] Wh/m
  - Stability [0, 45]°



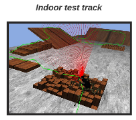
Test Tracks

DRC mobility test track replicate for operator selection

Outdoor test track



Indoor test track

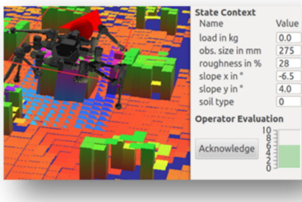


Resulting Behavior Libraries

library name	behaviors	contexts	total evaluations
in_op1	26	15	36
in_op2	26	31	39
in_op3	57	45	174
in_op4	21	14	27
in_op5	24	19	56
inLib	154	93	332
out_op1	41	18	81
out_op2	46	17	86
out_op3	49	24	93
out_op4	52	20	90
out_op5	41	21	63
outLib	229	77	413
fullLib	383	157	745

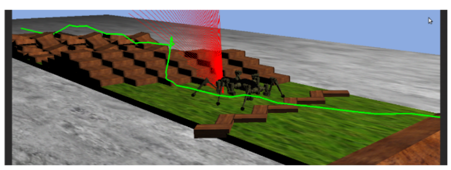
Methodology

- Learning from demonstration by collecting experiences while operators control SpaceClimber through obstacle courses
- Evaluation of operator skills
  - Path following capability [0.5, 0.1] m
  - Energy efficiency [1.4, 0.14] Wh/m
  - Stability [0, 45]°

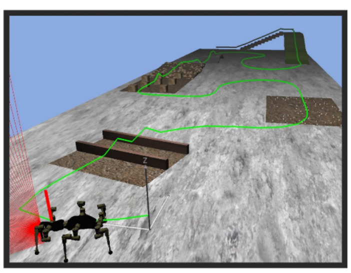


Test Tracks

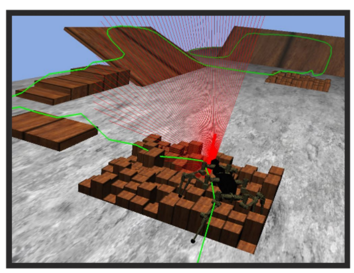
DRC mobility test track replicate for operator selection



Outdoor test track

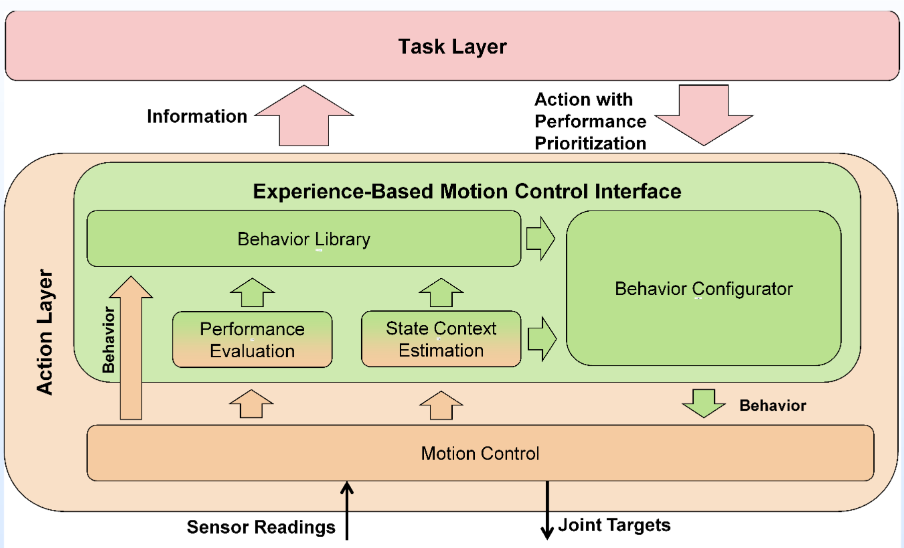


Indoor test track



Resulting Behavior Libraries

library name	behaviors	contexts	total evaluations
in_op1	26	15	36
in_op2	26	31	39
in_op3	57	45	174
in_op4	21	14	27
in_op5	24	19	56
inLib	154	93	332
out_op1	41	18	81
out_op2	46	17	86
out_op3	49	24	93
out_op4	52	20	90
out_op5	41	21	63
outLib	229	77	413
fullLib	383	157	745



Case-Based Selection

- Input query in form of two vectors
  - Current state context features
  - Current desired action described by action performance features
  - Meta performance features constant at optimum
- Additional weight vectors to manipulate feature importance

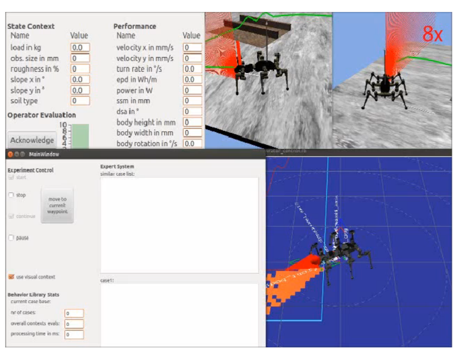
for each $behavior\_eval$ in $behavior\_library$ **do**
  for each $context\_eval$ in $behavior\_eval$ **do**
    $Sim_e^{State} = getStateContextSimilarity()$
  end for
  $Sim^{State} = getMaxStateSimilarity()$
  $e_{max} = getMostSimilarContextEvaluation()$
  $Sim^{Action} = getActionSimilarity()$
  $Sim = getBehaviorSimilarity()$
end for
$applyMostSimilarBehavior(blend\_time)$

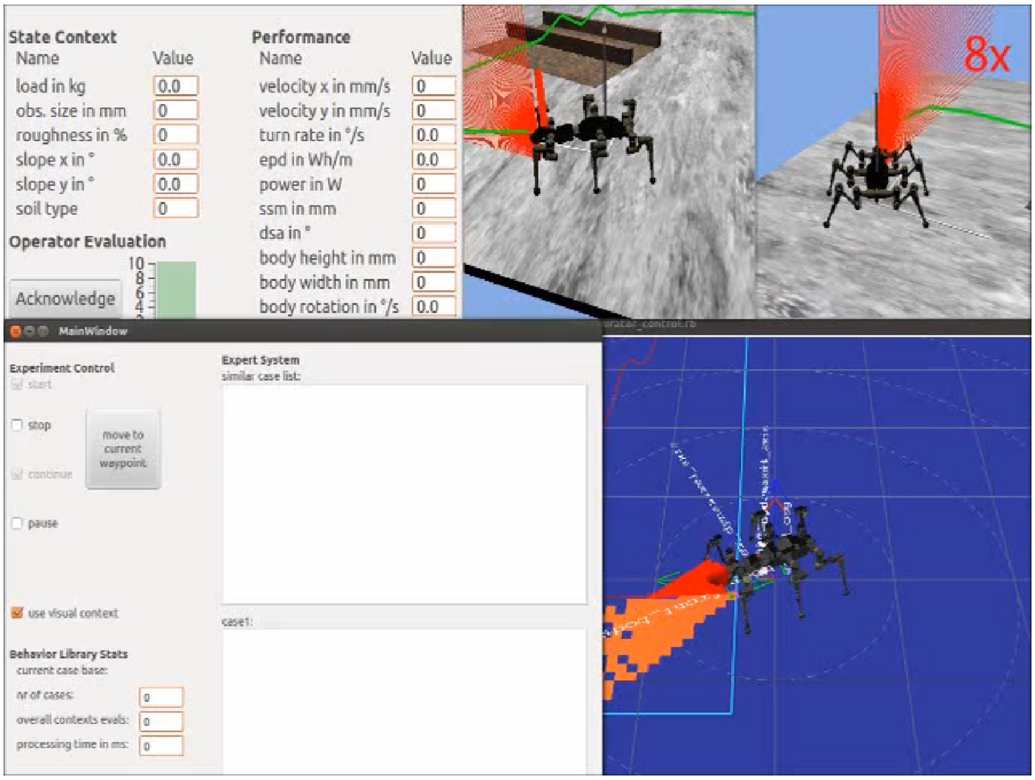
Autonomous Control

- Operator replaced by
  - Trajectory follower to generate motion commands
    - velocity x from 0 to 0.15 m/s
    - turn rate from -10°/s to 10°/s

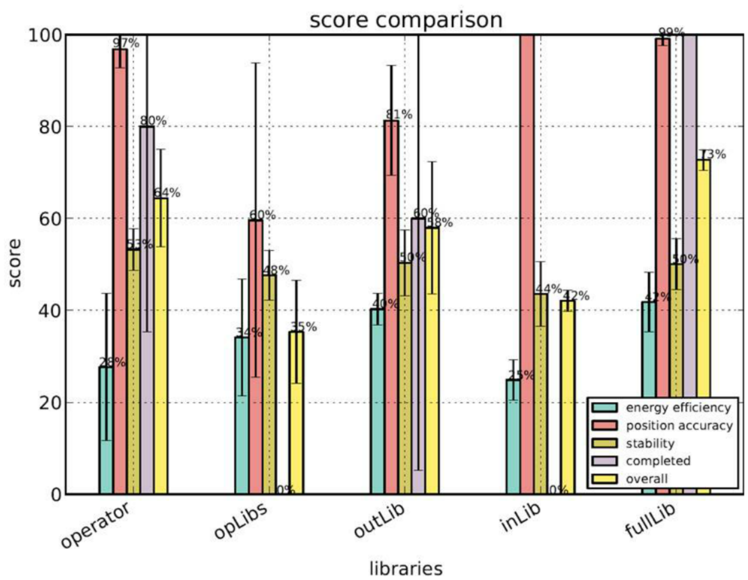
    depending on orientation error
  - Behavior configurator for autonomous behavior adaptation
    - 2 s blend time between behaviors

feature	weight
velocity x	0.8
velocity y	0.2
turn rate	1.0
body height	0.0
body width	0.0
ssm	0.0
dsa	0.1
power	0.0
epd	0.1
vibration	0.1





Results on Outdoor Obstacle Course



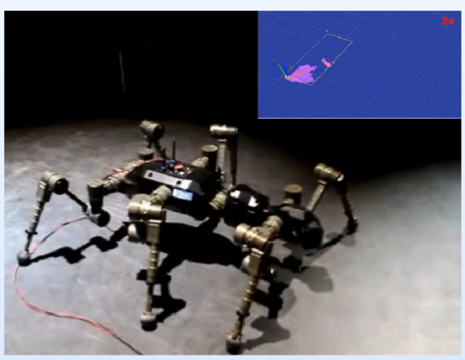
Conclusion & Outlook

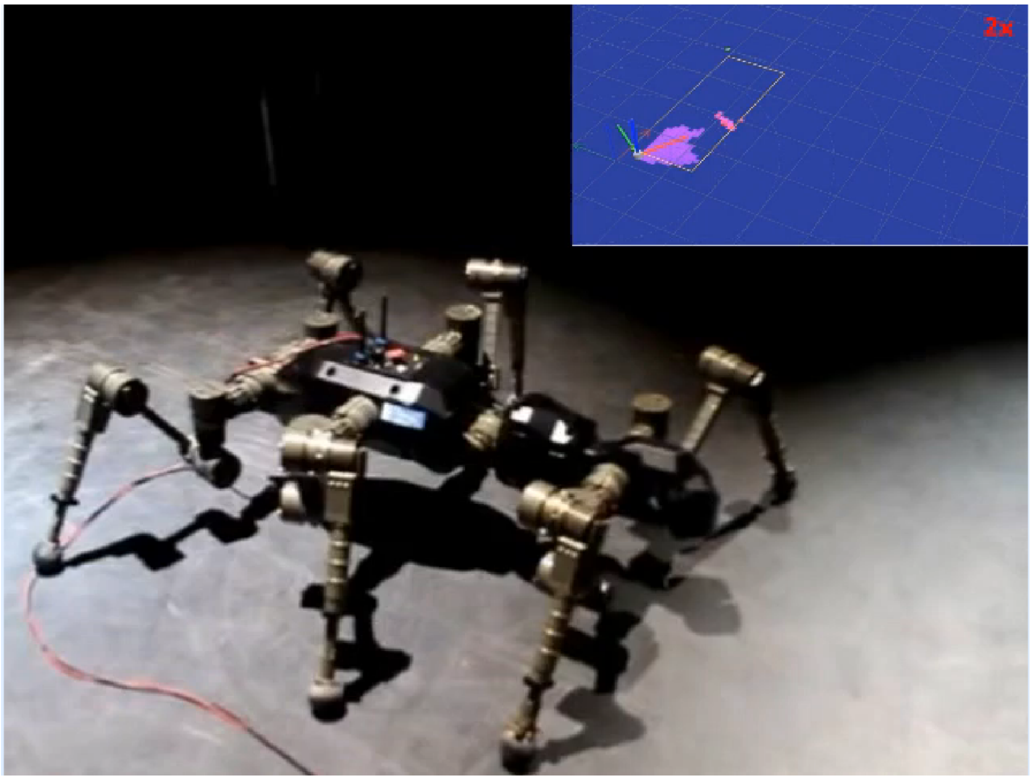
- Storing of behavior performance for a diverse state contexts
- Utilization of experiences for autonomous behavior adaptation
- Performance of autonomous control increases with maturity of behavior library

- Shortcomings of case-based selection
  - No generalization
  - No exploration
  - No forgetting strategy

- Next Steps
  - Improving case-based reasoning approach
    - Case adaption, forgetting strategy
  - Testing LWPR, GMR, or SOGP

Thank you for your attention!




2x