3.6 'An Experience-Based Interface for Abstracting the Motion Control of Kinematically Complex Robots' (LM-P-02)

Alexander Dettmann[(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

In order to provide higher mobility and to assist humans in building up infrastructure in future extraterrestrial space missions, kinematically complex robots are needed. One key challenge which needs to be addressed is to handle their complex motion control and to make use of their high potential. Utilizing the possibility to achieve various actions even in different ways by tuning manually numerous parameters of the motion control can be very demanding and even unmanageable when also taking communication delay into account.

Thus, the proposed experience-based interface is encapsulating the motion control of complex robots by autonomously mapping application-specific action parameters to robot-specific motion control parameters depending on the current context. Therefore, the robot is using experiences collected from previously executed behaviors. Apart from acquiring experiences during operation of the real robot, they can also be collected in simulation. The possibility to test in low gravity environments makes the latter a valuable tool for increasing the robot's knowledge base for space missions.

The experiments in this paper show that reconfiguring the motion control can be beneficial and that in simulation optimized behaviors can easily be integrated in the experience-based control interface to improve the performance of a robot. In addition, the transferability from simulation to the real system is shown.

Please note, that the corresponding paper is published in:
An experience-based interface for abstracting the motion control of kinematically complex robots; A. Dettmann, S. Bartsch, and F. Kirchner; In Proceedings of ASTRA 2015.

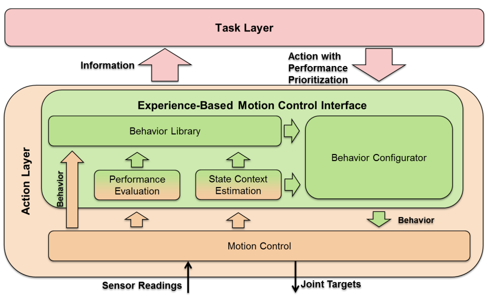
German Research Center for Artificial Intelligence

An Experience-Based Interface for Abstracting the Motion Control of Kinematically Complex Robots

Alexander Dettmann, Sebastian Bartsch, and Frank Kirchner

Introduction

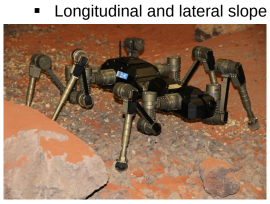
- Future space mission require higher mobility to reach locations of scientific or ecological interest
- Kinematically complex robots
  - Capable of realizing numerous tasks and adapting to varying contexts
  - Require sophisticated motion control which needs suitable parameterization to produce desired behavior
  - Same action can be realized by numerous behaviors with different behaviors
  - High control effort resulting in high operator load
- Autonomous mapping between scenario-specific action and robot-specific parameters needed which also incorporates current context



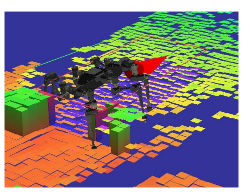
Experience-based motion control interface

Performance and State Context Features for Locomotion

- Performance features characterize robot's behavior
  - Action performance features characterize action execution
    - Longitudinal and lateral velocity
    - Turn Rate
  - Meta performance features characterize
    - Stability (static stability measure, dynamic stability angle)
    - Efficiency (power, energy per distance, body vibration)
- State context features characterize environment
  - Step Hazard
  - Roughness
  - Longitudinal and lateral slope



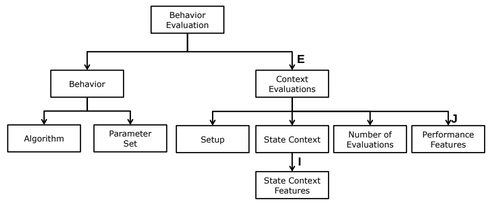
SpaceClimber in ESA's Mars Yard (ESTEC)



Generated map and region of interest for state context estimation

Behavior Library

- Behaviors (Algorithm + Parameterization)
- State Contexts
- Behavior Evaluations



Experiences stored in behavior evaluations

Behavior Configurator

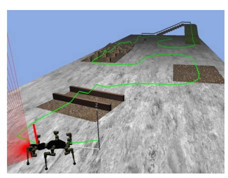
```
normalize(cur_state_context_features)
normalize(des_action_features)
for each behavior_eval in behavior_library do
    for each context_eval in behavior_eval do
        Sim_e^State = getStateContextSimilarity()
    end for
    Sim^State = getMaxStateSimilarity()
    e_max = getMostSimilarContextEvaluation()
    Sim^Action = getActionSimilarity()
    Sim = getBehaviorSimilarity()
end for
applyMostSimilarBehavior(blend_time)
```

$$Sim_e^{State} = 1 - \frac{\sum_{i=1}^{I}(s_i^{cur} - s_i^{ref,e})^2 \cdot w_i^S}{\sum_{i=1}^{I} w_i^S}$$

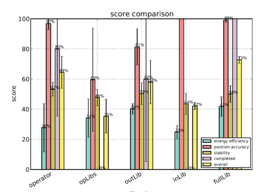
$$Sim^{State} = \max_{e \in E}(Sim_e^{State}) \quad e_{max} = \operatorname{argmax}_{e \in E}(Sim_e^{State})$$

$$Sim^{Action} = 1 - \frac{\sum_{j=1}^{J}(p_j^{cur} - p_j^{ref,e_{max}})^2 \cdot w_j^P}{\sum_{j=1}^{J} w_j^P}$$

$$Sim = Sim^{State} \cdot Sim^{Action}$$



Obstacle course to collect experiences



Score comparison between manual and autonomous control utilizing different behavior libraries

Conclusion

- Motion control abstracted
  - Action-specific instead of robot-specific interface
  - Performance prioritization possible
  - Autonomous configuration of control layer
- Constantly growing behavior library
  - Gaining confidence during operation
  - Incorporating system wearout
  - Storing and utilizing real and simulated experiences possible

Supported by:

DLR

The presented work was carried out in the project LIMES, a collaboration between the DFKI Robotics Innovation Center and the University of Bremen, funded by the German Space Agency (DLR, Grant numbers: 50RA1218, 50RA1219) with federal funds of the Federal Ministry of Economics and Technology (BMWi) in accordance with the parliamentary resolution of the German Parliament.

Universität Bremen

Contact:
DFKI Bremen & University of Bremen
Robotics Innovation Center

Director: Prof. Dr. Frank Kirchner
E-mail: robotics@dfki.de
Website: www.dfki.de/robotics