Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Master's Thesis

Lazy Robot Control by Relaxation of Motion and Force Constraints

Author:
Djordje Vukcevic

Supervisors:
Prof. Dr. Paul G. Plöger
Prof. Dr. Herman Bruyninckx
M.Sc. Sven Schneider

February 11, 2020

Motivation

Figure 1: Human pouring water with the help of surface support

- Humans **adapt** their accuracy and precision
  → Constantly to everyday requirements [1]
- Focus on **energy** and **stress** reduction
  → In contrast to traditional robots [2]
- **Exploit** existing natural forces and constraints
  → E.g. gravity, surface support...
- While performing actions, humans make mistakes
  → Constantly performing **corrections** [2], [3]

This **lazy** strategy still allows high performance

Motivation

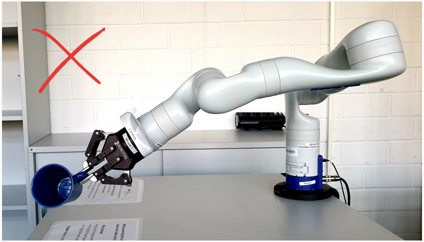


Figure 2: Robot pouring water by avoiding the surface contact

- **Model** and **parameters** of robots & environment: never completely accurate!
- In **sensors** noise always exist!

Motivation

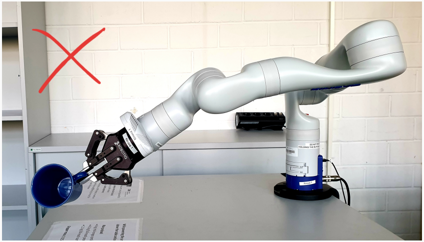


Figure 2: Robot pouring water by avoiding the surface contact

- **Model** and **parameters** of robots & environment: never completely accurate!
- In **sensors** noise always exist!

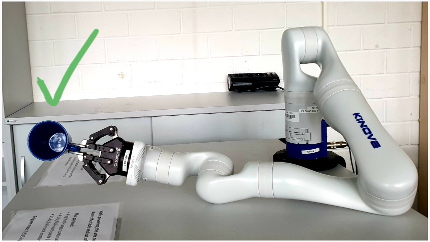


Figure 3: Robot pouring water with the help of surface support

Relax force and motion constraints

Solve it with the help of **environment**

Don't look for **optimal** control solutions
→ Go for the ones that are **good enough**

Problem Formulation

Traditional control approaches do not fully exploit **prior knowledge** about the task, environment and robot system [4]–[9]

Problem Formulation

Traditional control approaches do not fully exploit **prior knowledge** about the task, environment and robot system [4]–[9]

- Concentrated on planning **robot trajectories**
- Motions **fully constrained** by position, velocity and acceleration parameters
- **Tight tolerances** imposed for the controller
- Expensive **optimization techniques** often employed to generate motion plans

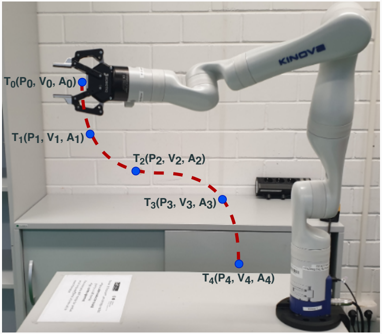


Figure 4: An example of the traditional robot trajectory

Problem Formulation

- Many strategies result in a **setpoint tracking** control [10]–[12]
  → **Not required** in most applications

- Perform **expensive predictions** of future robot states [4]–[6]
  → **Whole dynamics** is evaluated in each step of prediction computations

- **Existing information** (by-product of control & dynamics computations)
  → Not fully exploited for the load **estimation** and **compensation** [13], [14]

Problem Formulation

- Many strategies result in a **setpoint tracking** control [10]–[12]
  → **Not required** in most applications

- Perform **expensive predictions** of future robot states [4]–[6]
  → **Whole dynamics** is evaluated in each step of prediction computations

- **Existing information** (by-product of control & dynamics computations)
  → Not fully exploited for the load **estimation** and **compensation** [13], [14]

Task

Relax these **aspects** of robot control

Problem Formulation

- Many strategies result in a **setpoint tracking** control [10]–[12]
  $\rightarrow$ **Not required** in most applications

- Perform **expensive predictions** of future robot states [4]–[6]
  $\rightarrow$ **Whole dynamics** is evaluated in each step of prediction computations

- **Existing information** (by-product of control & dynamics computations)
  $\rightarrow$ Not fully exploited for the load **estimation** and **compensation** [13], [14]

Task

Relax these **aspects** of robot control

How?

Exploit prior **knowledge** about the task, robots and environment

Approach (Part 1): Task Specification

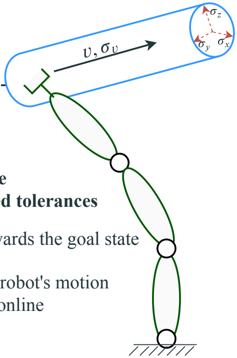
Create a **less constraining** task specification

Create a **less constraining** task specification

- Consider **area** error instead of the **setpoint** error
  - For both, **motion** and **force** variables
- **Eliminate** generation & optimization of robot **trajectories**

Task

Allow for possibility of leaving some of task DOFs **unspecified**
-> **Unconstrained** task directions can "**emerge naturally**" [9]



- Constrain robot motion via **tube**
  - Area (volume) with **relaxed tolerances**
- **Guide** robot through a tube, towards the goal state
- Trajectories are **outcome** of the robot's motion
  - Not planned in advance or online

Approach (Part 2): Resolving Motion and Force Constraints

- **Adimensional & Adaptive**
- Enables possibility of limiting control output
  - **Wind-up** situation avoided
- Good **predictability** & **stability** of the controller's behaviour

Task

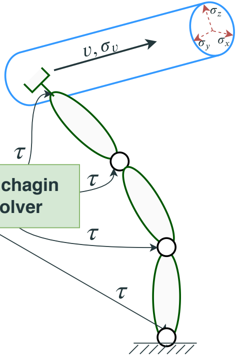
$X_d, \dot{X}_d, F_d$

Adaptive Bias Adaptive Gain (ABAG) Controller

$\ddot{X}, F$

Popov-Vereshchagin Dynamics Solver

τ

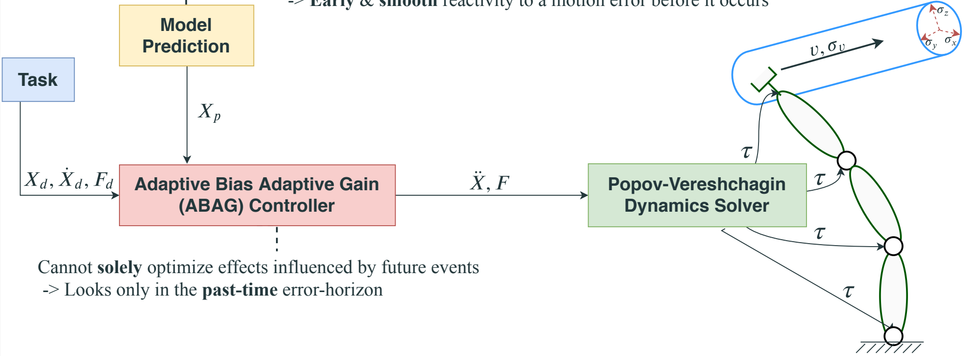


- Exposes **tree** different task (input) interfaces
  - (Partial) Cartesian Acceleration Constraints
  - Cartesian External Forces
  - Feed-forward Joint Torques
- Resolves dynamics **analytically** with linear time-complexity
  - Even with **partial** task specifications (some of DOFs left unspecified)
  - **Redundancy** resolved based on Gauss' *criterion of least constraints*"
  - Acceleration energy **optimal** solution

Approach (Part 3): Predicting Future Robot States

- Method for estimating **future** robot states (poses)
- **Relax** prediction computations
  - Make them **inexpensive**
  - **Integrate** end-effector Cartesian **velocities** over finite time-interval

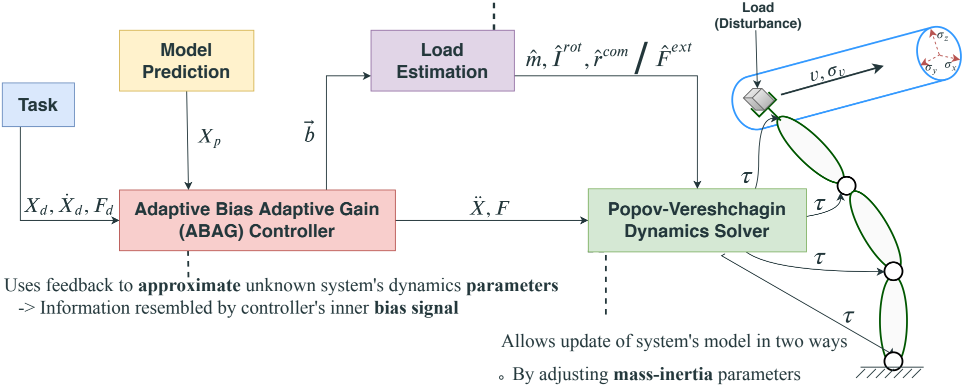
Benefit of combining predictions with the ABAG
-> **Early & smooth** reactivity to a motion error before it occurs



Cannot **solely** optimize effects influenced by future events
-> Looks only in the **past-time** error-horizon

Approach (Part 4): Estimating and Compensating Unknown Load

- Method for estimating unknown **load effects** and **parameters**
- **Relax** compensation procedure
  - **Take advantage** of already existing information
  - **Infer** load parameters from **by-product** of control computation
- **Update** system's model parameters accordingly
  - Improve **feedforward** of the control architecture



Uses feedback to **approximate** unknown system's dynamics **parameters**
-> Information resembled by controller's inner **bias signal**

Allows update of system's model in two ways

- By adjusting **mass-inertia** parameters
- By specifying a **Cartesian force** on the end-effector

Evaluation: Scenario 1 - Setup

- *Goal*: **Validate** approach in a **simulation** environment
- *Robot System*: 7-DOF KUKA Lightweight Robot 4 [15]
- *Use Case*: Performing a "**pre-grasp**" motion
- *Command*:
  - Move towards pre-defined goal area
  - Keep the end-effector within tube bounds
  - Maintain predefined speed throughout the tube
- *Initial Condition*:
  Initial end-effector's position is outside of tube bounds

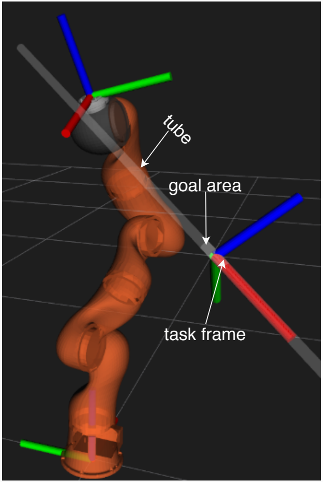


Figure 5: KUKA LWR 4 model, its initial configuration & a virtual tube (based on the task-defined bonds)

Evaluation: Scenario 1 - Task Specification

move compliantly { // with task frame directions

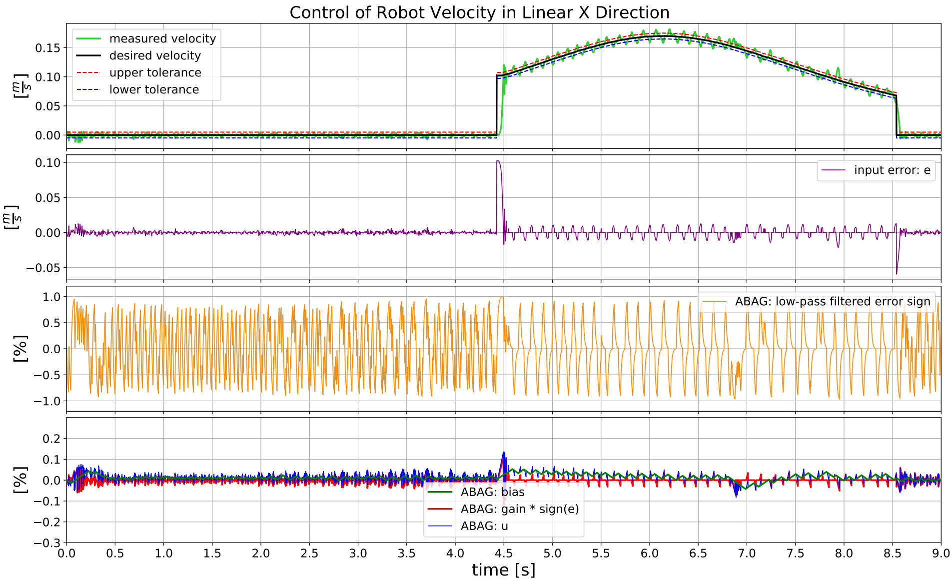
^{T}X: s-curve velocity profile[1] [m/s], velocity-tolerance 0.005 [m/s];
^{T}Y: 0 [m], position-tube 0.01 [m];
^{T}Z: 0 [m], position-tube 0.01 [m];
$\left.\begin{array}{l} ^{T}aX: \\ ^{T}aY: \\ ^{T}aZ: \end{array}\right\}$ no specification;

} until:

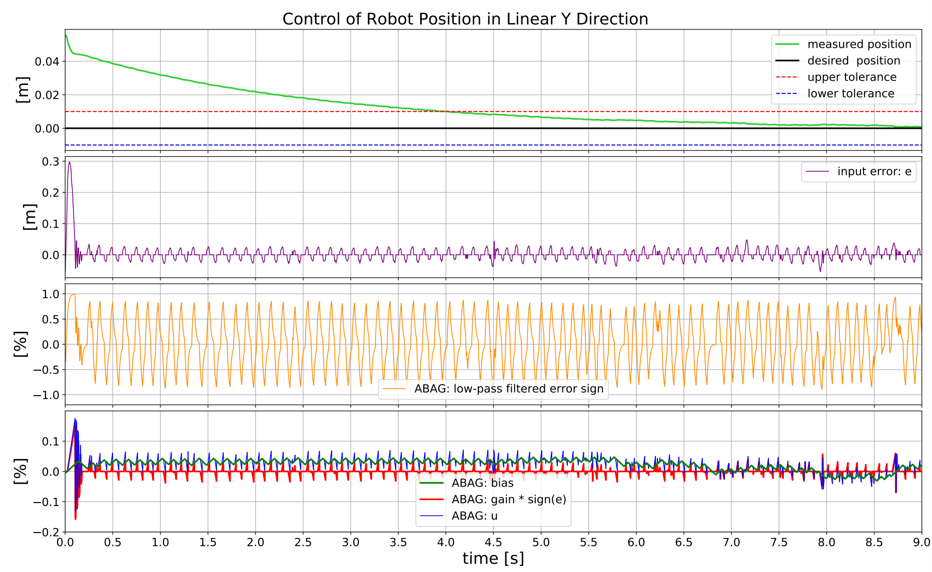
goal area reached, tolerance 0.03 [m];
$^{T}F_X$ / $^{T}F_Y$/ $^{T}F_Z > 0.5$ [N];
task time > 9 [s];

[1] S-Curve Velocity Profile: $0.05 + 0.12 \cdot \sin(\text{current\_robot\_position\_x\_axis} \cdot 5.0)$ [m/s]

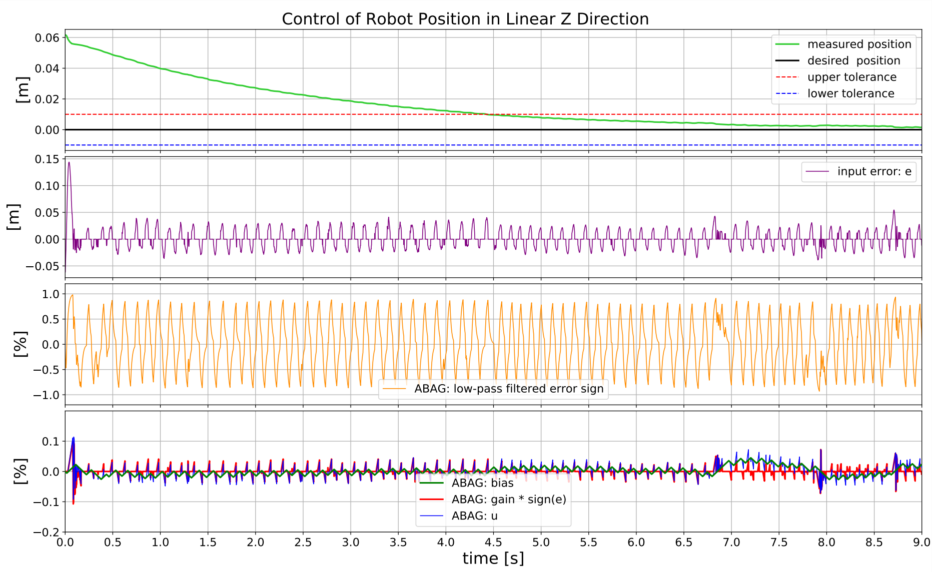
Evaluation: Scenario 1 - Results



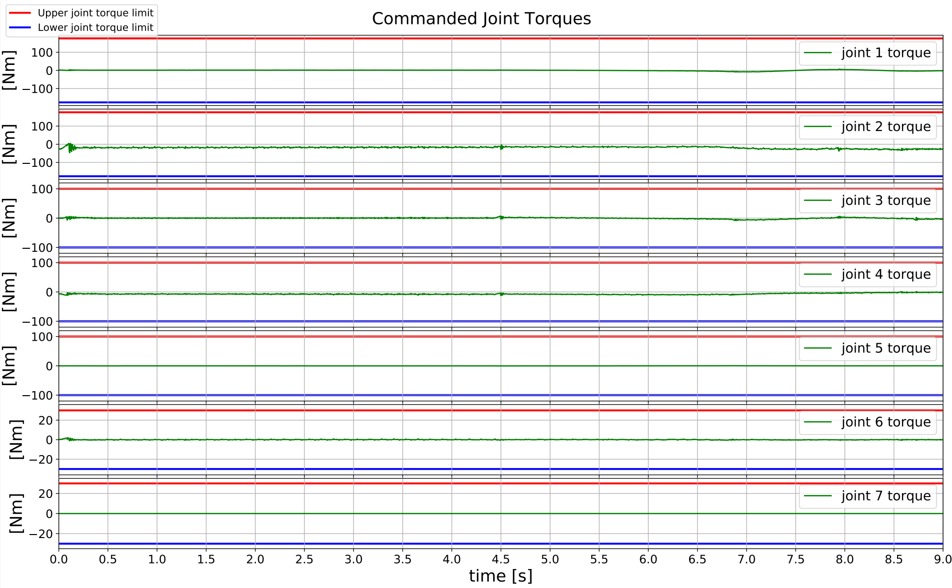
Evaluation: Scenario 1 - Results



Evaluation: Scenario 1 - Results



Evaluation: Scenario 1 - Results



- *Goal*: **Validate** approach on a **real robot** platform
- *Robot System*: 5-DOF KUKA youBot [16]
- *Use Case*: Performing a "**pre-grasp**" motion
- *Command*:
  - Move towards pre-defined goal area
  - Keep the end-effector within tube bounds
  - Maintain predefined speed throughout the tube
- *Initial Condition*:
  Initial end-effector's position is outside of tube bounds

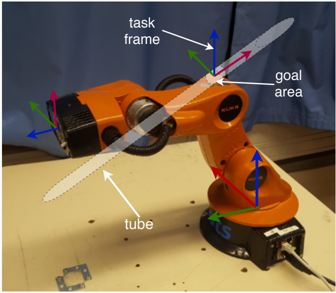


Figure 6: 5-DOF KUKA youBot, its initial configuration & a virtual tube (manually added in the image)

Evaluation: Scenario 2 - Task Specification

move compliantly { // with task frame directions

^{T}X: s-curve velocity profile [2] [m/s], velocity-tolerance 0.003 [m/s];
^{T}Y: 0 [m], position-tube 0.015 [m];
^{T}Z: 0 [m], position-tube 0.015 [m];

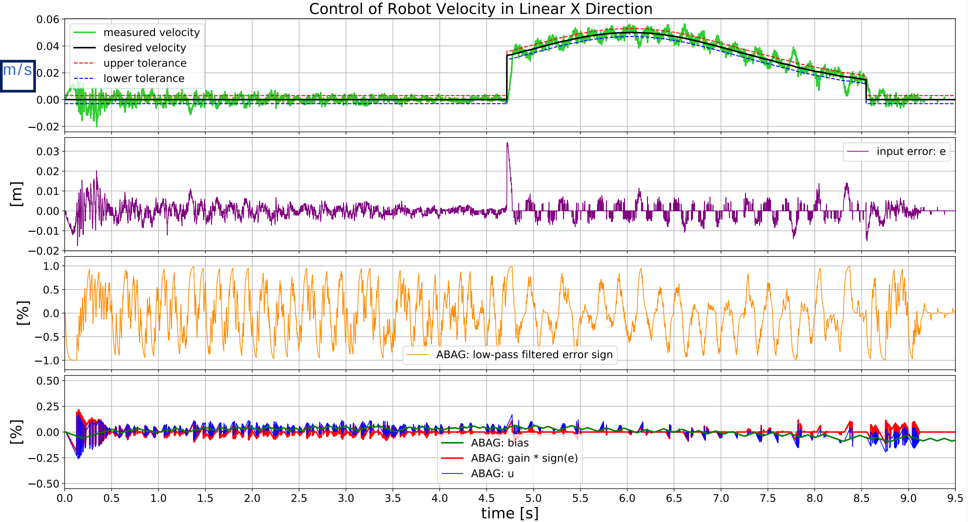
$$\left.\begin{array}{l} {}^{T}aX: \\ {}^{T}aY: \\ {}^{T}aZ: \end{array}\right\} \text{no specification;}$$

} until:

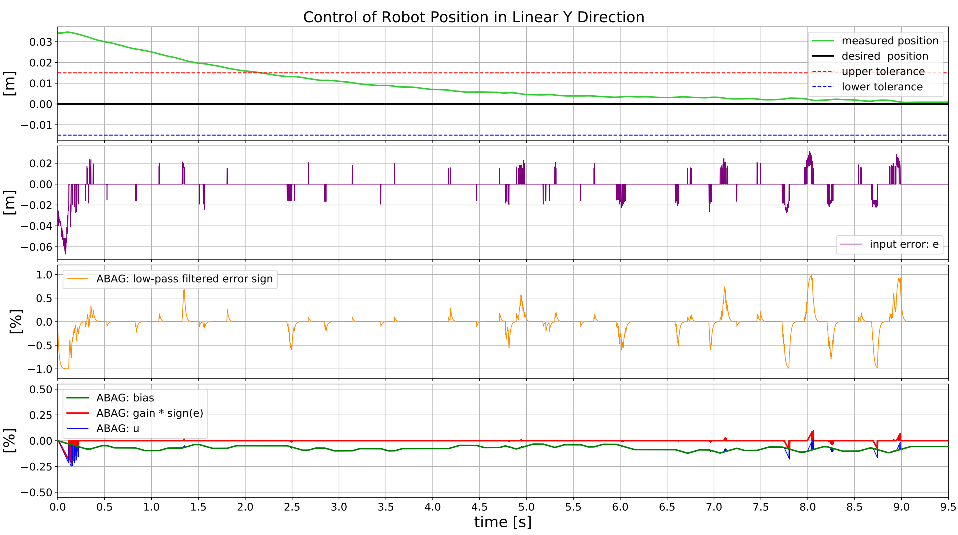
goal area reached, tolerance 0.02 [m];
task time > 9.5 [s];

[2] S-Curve Velocity Profile: $0.05 \cdot \sin(\text{current\_robot\_position\_x\_axis} \cdot 15.0)$ [m/s]

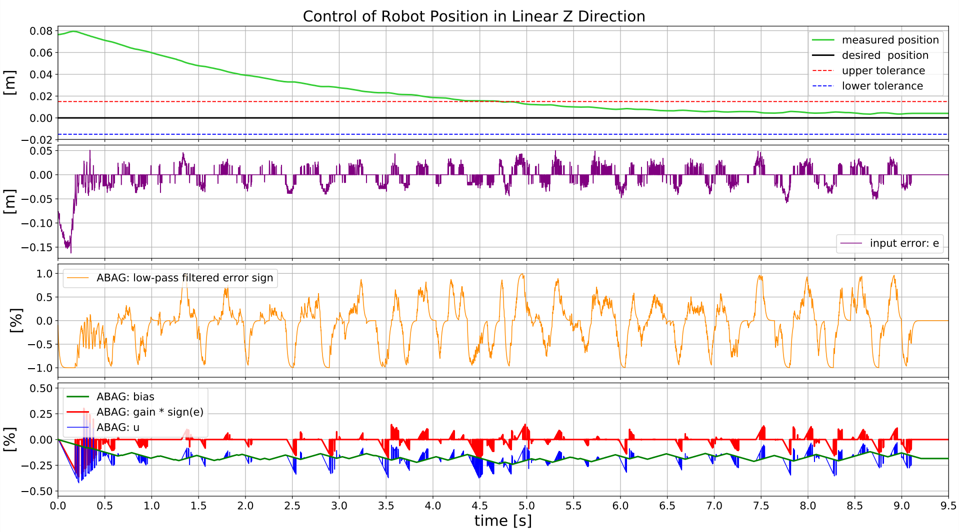
Evaluation: Scenario 2 - Results



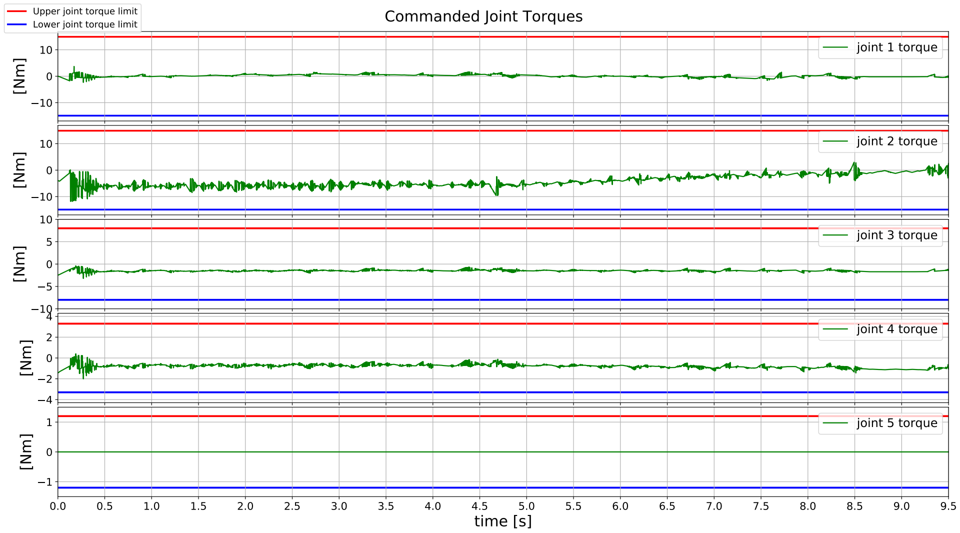
Evaluation: Scenario 2 - Results



Evaluation: Scenario 2 - Results



Evaluation: Scenario 2 - Results



Conclusion

A **lazy** robot control strategy that **relax** several control aspects has been developed

- Task specification models
- Control architectures

Features of the Popov-Vereshchagin dynamics solver & ABAG controller **exploited** for deriving control architectures

Additional components have been developed:

- **Prediction** units that provide estimates of future robot poses
- **Estimator** unit that serves for estimating an unknown load

An experimental **evaluation** has been performed

- In a **simulation environment**
- On a **real** robot **platform**

Feasibility of the proposed lazy control approach was demonstrated

Future Work

Extending future state estimations:

- Future **Cartesian velocities** can be **simply derived** from the P-V solver's output
  $\rightarrow$ Integrate resultant spatial **segment-accelerations** $\ddot{X}$
  $\rightarrow$ Improve **even more** by fusing with **IMU data**, if available

Extending task specification model:

- Only one *task frame* considered, for both position and orientation constraints
- Consider **two** task frames, one for **position** & one for **orientation** constraints
- In this way, task specification model is **even less** constraining

Extending evaluation and experiment scenarios:

- Evaluate control of robot's orientation
- Evaluate a contact-force control (stabilization) on a real robot platform
- Evaluate load estimation and compensation capabilities

Thank you!

Questions?

Appendix - Approach: Energy Consumption

- Leave some task objectives to the system's **natural dynamics**
  - **Take advantage** of the existing forces acting on the system
  - E.g. gravity, contact forces, friction in the motors, etc.
- Popov-Vereshchagin hybrid dynamics solver:
  - While resolving **instantaneous** dynamics
  - Based on the **known** model parameters [18]
- ABAG controller:
  - While computing the **non-instantaneous** control
  - Based on the approximation of **non-modelled** parameters [19]

Combined effort improves energy consumption of the system's actuators

Appendix: Following a Pre-defined Path Use Case

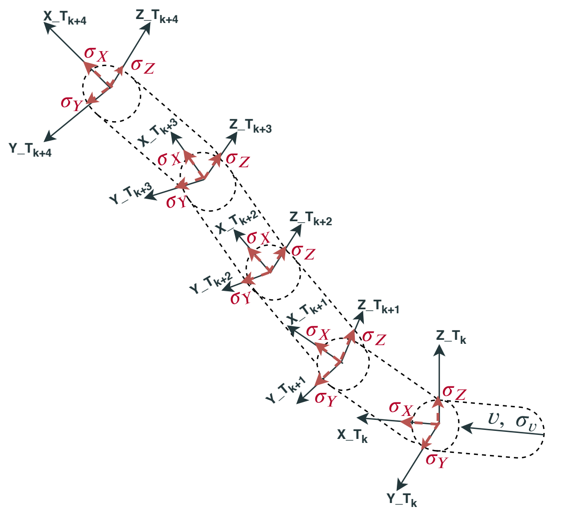


Figure 7: Illustration of an example path, defined for realizing the task in the second use case

Appendix: Cleaning a Surface Use Case

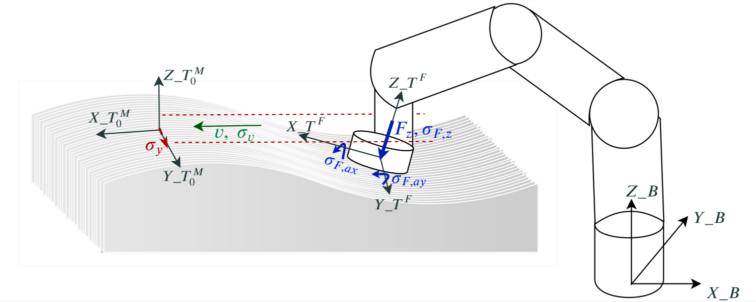


Figure 8: Illustration of the robot action performed in the third use case. In this example, only one goal area is defined. Nevertheless, if required, the task specification can be extended with a pre-defined Cartesian path, in the same manner as in the second use case.

References

[1] E. Todorov, "Optimality principles in sensorimotor control," *Nature neuroscience*, vol. 7, p. 907, 2004.

[2] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nature neuroscience*, vol. 5, p. 1226, 2002.

[3] F. R. Hogan and A. Rodriguez, "Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics," *ArXiv preprint arXiv:1611.08268*, 2016.

[4] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the HRP-2 humanoid," in *IEEE International Conference on Intelligent Robots and Systems*, 2015.

[5] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," *IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017.

[6] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[7] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural computation*, vol. 25, pp. 328–373, 2013.

[8] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1996.

[9] R. J. Griffin, G. Wiedebach, S. Bertrand, A. Leonessa, and J. Pratt, "Straight-leg walking through underconstrained whole-body control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[10] G. Borghesan and J. De Schutter, "Constraint-based specification of hybrid position-impedance-force tasks," in *IEEE International Conference on Robotics and Automation*, 2014.

[11] C. Schindlbeck and S. Haddadin, "Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[12] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[13] S. Farsoni, C. T. Landi, F. Ferraguti, C. Secchi, and M. Bonf, "Compensation of load dynamics for admittance controlled interactive industrial robots using a quaternion-based kalman filter," *IEEE Robotics and Automation Letters*, vol. 2, 2017.

[14] D. Kubus, T. Kroger, and F. M. Wahl, "Improving force control performance by computational elimination of non-contact forces/torques," in *IEEE International Conference on Robotics and Automation*, 2008.

[15] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger, "The KUKA-DLR Lightweight Robot arm-a new reference platform for robotics research and manufacturing," in *ISR (International Symposium on Robotics) and ROBOTIK (6th German Conference on Robotics)*, 2010.

[16] R. Bischoff, U. Huggenberger, and E. Prassler, "KUKA youBot - a mobile manipulator for research and education," in *2011 IEEE International Conference on Robotics and Automation*, 2011.

[17] T. Lefebvre, H. Bruyninckx, and J. De Schutter, *Nonlinear Kalman filtering for force-controlled robot tasks*. Springer, 2005.

[18] D. Vukcevic, S. Schneider, and P. G. Plöger, "Extending a constrained hybrid dynamics solver for energy-optimal robot motions in the presence of static friction," Hochschule Bonn-Rhein-Sieg, Department of Computer Science, Tech. Rep. 03-2018, 2018, p. 66.

[19] A. Franchi and A. Mallet, "Adaptive closed-loop speed control of BLDC motors with applications to multi-rotor aerial vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.