Two-Degree-of-Freedom Control for Trajectory Tracking and Perturbation Recovery during Execution of Dynamical Movement Primitives

Martin Karlsson   **Fredrik Bagge Carlson**
Anders Robertsson   **Rolf Johansson**

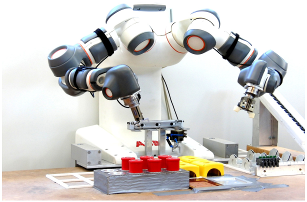
Dept. Automatic Control, Lund University

Introduction

Aim: Continue a robot motion, generated by a dynamical movement primitive (DMP), after an unforeseen event.

- Physical contact with a human
- Pause until certain condition is fulfilled
- Obstacle avoidance

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, no. 2, pp. 328–373, 2013.



Presentation Outline

- Previous research
  - Introduction to DMPs
  - DMP perturbation recovery
- Proposed method
- Simulations
- Experiments
- Results
- Discussion and Future Work
- Conclusion

Introduction to DMPs

Dynamical movement primitives, used to model (robot) motion

Three main building blocks:

1. Goal (attractor), g
2. Shape, f
3. Time scale, τ

Introduction to DMPs

Damped-spring system:

$$\tau^2 \ddot{y} = \alpha_z(\beta_z(g - y) - \tau \dot{y}) + f(x) \tag{1}$$

where

$$\tau \dot{x} = -\alpha_x x \tag{2}$$

$$f(x) = \frac{\sum_{i=1}^{N_b} \Psi_i(x) w_i}{\sum_{i=1}^{N_b} \Psi_i(x)} x \cdot (g - y_0) \tag{3}$$

Introduction to DMPs

- A DMP can be used to generate a robot trajectory
- A DMP can be determined given a demonstrated trajectory
- Allows for replanning without recomputation of DMP parameters

Previous Research

- Aim: The robot should be able to recover from a perturbation, and continue the planned trajectory.
- Problem: In the original DMP formulation, the time evolution of x would be unaffected by any perturbation.

$$\tau^2 \ddot{y} = \alpha_z(\beta_z(g - y) - \tau \dot{y}) + f(x) \tag{4}$$

where x evolves as

$$\tau \dot{x} = -\alpha_x x \tag{5}$$

Previous Research

General idea presented previously:

- In case of a tracking error, slow down the time evolution of the DMP.
- Use a PD controller to drive the actual trajectory, y_a, toward a coupled trajectory generated by the DMP, y_c.

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, no. 2, pp. 328–373, 2013.

Previous Research

Coupling terms:

$$\dot{e} = \alpha_e(y_a - y_c) - \alpha_e e \tag{6}$$

$$C_t = k_t e \tag{7}$$

$$\tau_a = 1 + k_c e^2 \tag{8}$$

Coupled DMP:

$$\tau_a \dot{z} = \alpha_z(\beta_z(g - y_c) - z) + f(x) + C_t \tag{9}$$

$$\tau_a \dot{y}_c = z \tag{10}$$

$$\tau_a \dot{x} = -\alpha_x x \tag{11}$$

PD controller:

$$\ddot{y}_r = K_p(y_c - y_a) + K_v(\dot{y}_c - \dot{y}_a) \tag{12}$$

Previous Research – Evaluation

- Problem: Small tracking errors result in slower evolution, even without perturbation
- This has been mitigated in simulations by using very large gains

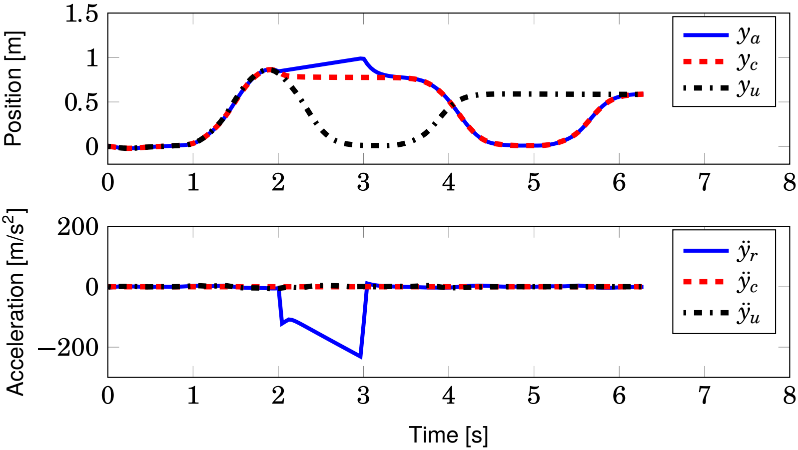
$$\ddot{y}_r = K_p(y_c - y_a) + K_v(\dot{y}_c - \dot{y}_a) \quad (13)$$

$$K_p = 1000 \quad (14)$$

$$K_v = 125 \quad (15)$$

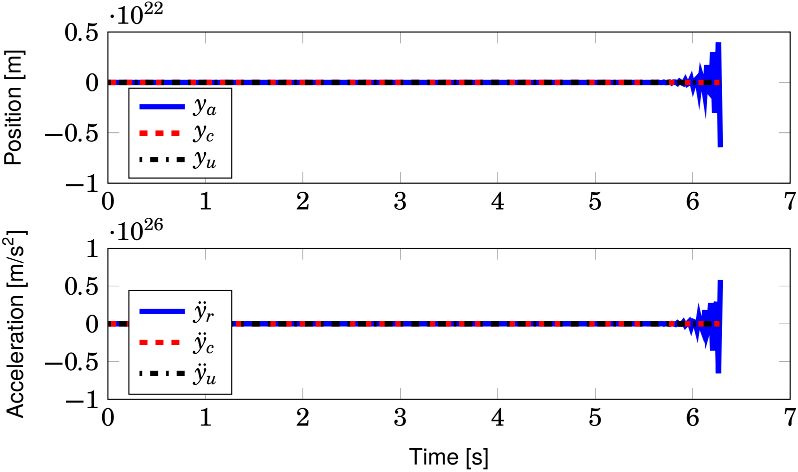
Previous Research – Evaluation

Large gains ⇒ prohibitively large acceleration reference



Previous Research – Evaluation

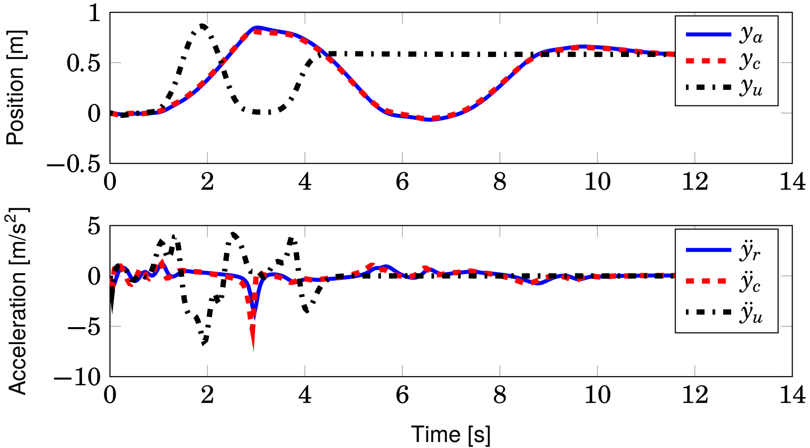
Large gains ⇒ delay margin of 12 ms



Previous Research – Evaluation

Moderate gains ⇒ small tracking error ⇒ slow evolution due to temporal coupling

$$K_p = 10, K_v = 25 \tag{16}$$

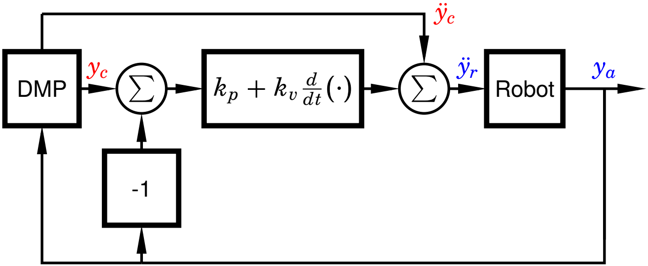


Method

That was an evaluation of previous research. Temporal coupling seems promising, but the controller is not practically realizable.

Could a feedforward term improve the tracking,
so that moderate gains could be used?

Method



$$\ddot{y}_r = k_p(y_c - y_a) + k_v(\dot{y}_c - \dot{y}_a) + \ddot{y}_c \quad (17)$$

$$k_p = 10 \quad (18)$$

$$k_v = 25 \quad (19)$$

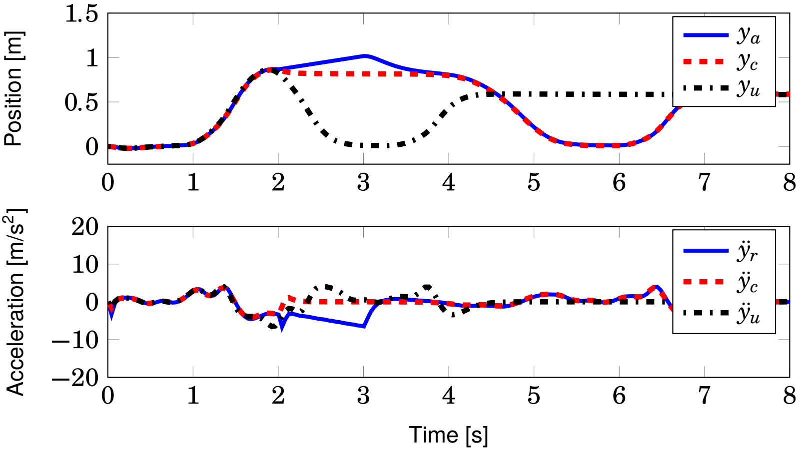
Note: $\ddot{y}_c$ does not include the time-derivative of any measured signal.

Method

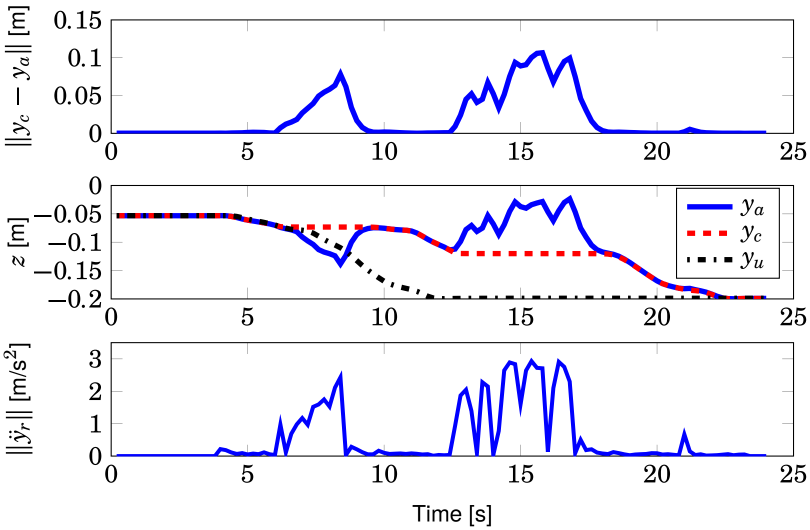
$$\ddot{y}_c = \frac{d}{dt}(\dot{y}_c) = \frac{d}{dt}\left(\frac{z}{\tau_a}\right) = \frac{\dot{z}\tau_a - z\dot{\tau}_a}{\tau_a^2} = \frac{\dot{z}\tau_a - 2\tau k_c z e \dot{e}}{\tau_a^2} \quad (20)$$

Simulations

Moving perturbation



Results – Scenario A



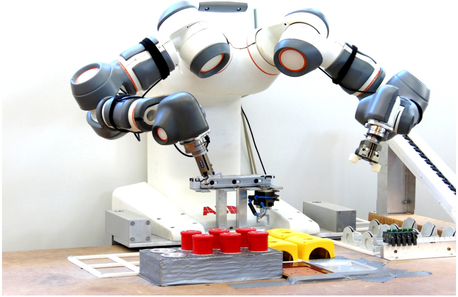
Discussion and Future Work

- Convergence to goal state
- Incorporation of trajectory-based learning and sensor data

Conclusion

- Extension of the DMP framework to enable perturbation recovery
- Feedforward control was used to track the reference trajectory generated by a DMP
- Feedback control with moderate gains to suppress deviations
- Temporal coupling
- Practically realizable control structure

Thank You!



The research leading to these results has received funding from the European Commission's Framework Programme Horizon 2020 – under grant agreement No 644938 – SARAFun.