[20.12.2020]

[Policy-Regularized Model Predictive Control
to Stabilize Diverse Quadrupedal Gaits for the MIT Cheetah]

Summary

A new policy-regularized model-predictive control approach to automatically generate and stabilize different quadrupedal gaits.

Stabilize means we are actually pushing the robot, that’s how it is tested.

Policy-regularized is because penalize on the u term, the control, the penalize the deviation from the simple policies.

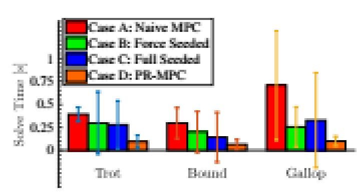
Some key points of this paper:

a) introducing heuristic reference policy to the cost function of MPC, the reference alone is not enough/correct.

   1. What would be the difference then? Because MPC anyway will have a desired trajectory and should try to follow that, then you are balancing between the desired one and a common-sense heuristic?

      Theoretically, it is like adding regularization to an optimization problem, for a MPC, adding regularization would 1) better conditioning for computational issues, 2) bias towards the heuristic, reduce the problem free degree, (a assumption)

b) computational time comparison: four kinds of experiments are implemented, <u>Here seeded means if we use reference forces/foot positions for MPC. Heuristics. PM means if the R in MPC is zero.</u>

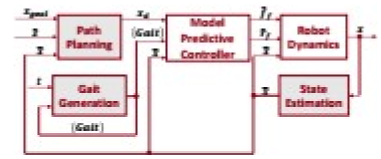


c) Same, four kinds are compared with their performance in conducting MPC. A fails or go into clear local optimum,

d) Pushing it, the performance differs, although when pushing it, the only-yaw assumption breaks, but it still manage the job.

Major Analysis and Comparison

1) Something different from what I did in my thesis: The desired trajectory is computed every time new observation/estimation is obtained, but in TOWR, the whole trajectory is first computed and no longer changes.
    - <u>Therefore, the prediction in TOWR is not as expected I think. Because the trajectory is done, it cannot handle huge disturbance actually. Therefore, it cannot predict the future steps, what it does is to connect to low level control, and predict how the actuators would behave.</u>



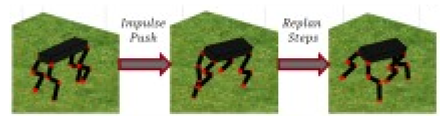


Fig. 8. Footstep planning with MPC allows the robot to choose new footstep locations autonomously to recover from disturbances.

2) Simplified model, simplified a lot, assume the robot only rotates around the z-axis, therefore only consider the yaw angle.

Thoughts

1) high level controller for path planning? What are they
2) computational issues for that, what kind of optimization is required?
3)