Robotics and Autonomous Systems 66 (2015) 27–34

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Learning robustly stable open-loop motions for robotic manipulation

Wouter Wolfslag [*,1,2], Michiel Plooij [1,2], Robert Babuška, Martijn Wisse [2]

Delft University of Technology, The Netherlands

HIGHLIGHTS

- Stable open-loop control of pick and place robots that handles model inaccuracies.
- Find robustly stable trajectory, then learn the tracking input online.
- Novel linear matrix inequalities based approach to determine robustness.
- Using Repetitive Control an open loop accuracy of 2.5 cm was obtained.

ARTICLE INFO

Article history:
Received 3 July 2014
Received in revised form
26 November 2014
Accepted 5 December 2014
Available online 9 January 2015

Keywords:
Feedforward control
Open-loop control
Robotic arms
Robustness
Linear matrix inequalities
Repetitive control

ABSTRACT

Robotic arms have been shown to be able to perform cyclic tasks with an open-loop stable controller. However, model errors make it hard to predict in simulation what cycle the real arm will perform. This makes it difficult to accurately perform pick and place tasks using an open-loop stable controller. This paper presents an approach to make open-loop controllers follow the desired cycles more accurately. First, we check if the desired cycle is robustly open-loop stable, meaning that it is stable even when the model is not accurate. A novel robustness test using linear matrix inequalities is introduced for this purpose. Second, using repetitive control we learn the open loop controller that tracks the desired cycle. Hardware experiments show that using this method, the accuracy of the task execution is improved to a precision of 2.5 cm, which suffices for many pick and place tasks.

1. Introduction

This research aims at future applications where sensing and feedback are undesirable due to costs or weight; or difficult due to small scale, radiation in the environment or frequent sensor faults. A recent example of such an application is the control of a swarm of nano-scale medical robots [1]. These applications inspire us to investigate an extreme case of feedback limitations: solely open-loop control on robotic arms. Control without any feedback can only be effective if two key problems are addressed: disturbances (e.g. noise and perturbations) and model inaccuracies.

The first problem, handling disturbances on an open-loop controlled robot, has mainly been addressed by creating open-loop stable cycles. The best known examples of this are passive dynamic walkers, as introduced by McGeer in 1990 [2]. Since those walkers do not have any actuators, there is no computer feedback control. The walking cycle of those walkers is stable, which means that small perturbations will decay over time. Such stable cyclic motions are called limit cycles. Limit cycle theory was later used to perform stable walking motions with active walkers, of which the closest related work is that by Mombaur et al. [3,4]. They optimized open-loop controllers for both stability and energy consumption and performed stable walking and running motions with those robots. Open-loop stable motions have also been used before to perform tasks with robotic arms. In 1993, Schaal and Atkeson showed open loop stable juggling with a robotic arm [5]. Even though their controller had no information about the position of the ball, they showed that any perturbation in this position decays over time, as long as a specific path of the robotic arm itself can be tracked. In a recent study, we showed that it is possible to perform repetitive tasks on a robotic arm with solely an open-loop current controller [6].

The second key problem with feedforward control (i.e. model inaccuracies) prevents the approach in [6] to be fully applicable:

* Correspondence to: Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands.
E-mail address: w.j.wolfslag@tudelft.nl (W.J. Wolfslag).

[1] These authors contributed equally to this work.

[2] These authors have received funding from the research programme STW (Project 11282), which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

http://dx.doi.org/10.1016/j.robot.2014.12.017
0921-8890/

it causes a difference between the motion as planned in simulation and as performed in hardware experiments. Handling model inaccuracies on open-loop controlled robots has recently become the subject of research. Singhose, Seering and Singer [7,8] researched vibration reducing input shaping of open-loop controllers while being robust to uncertainty in the natural frequency and damping of the system. Becker and Bretl [9] researched the effect of an inaccurate wheel diameter of unicycles on the performance of their open-loop velocity controller. In their case, open-loop control means that the position of the unicycle is not used as an input for the controller, but the velocity of the wheels is. In a previous paper, we showed that on a robotic arm, different open-loop current controllers have different sensitivities to model inaccuracies [10]. We found open-loop current controllers of which the end position of the motion is independent of the friction parameters. However, the motions that handle the model inaccuracy problem of feedforward control, the stability problem still exists, i.e., disturbances acting on these motions will grow over time.

Since these two problems of disturbances and model inaccuracies in open-loop control have only been addressed separately, no applicable purely open-loop control scheme has been devised. This paper shows that repetitive tasks can be performed stably by robotic arms with an open-loop voltage controller, even when an accurate model is not available.

In order to achieve this goal, the problem is split into two phases (see Fig. 1). In the first phase the robustness of the system is analyzed with a novel method based on linear matrix inequalities (LMI) [11]. In the second phase repetitive control (RC) [12] is used to learn the exact control input, such that the desired positions are reached accurately. During this learning phase, very slow feedback is allowed, this feedback can be removed after the learning has been completed.

The rest of this paper is structured as follows. Section 2 shows why the problem can be split into two phases, and explains the robustness analysis method and the repetitive control scheme. Next, Section 3 shows the experimental setup we used to test our approach. Then, Section 4 shows the results of both the numerical and the hardware experiments. Finally, the paper ends with a discussion in Section 5 and a conclusion in Section 6.

2. Methods

In this section we explain our methods. First, in Section 2.1 we discuss the basic concept of the stability analysis. Second, Section 2.2 explains our approach to perform robustly stable open-loop cycles. Then we will describe the two steps of this approach separately: a robust stability analysis (Section 2.3) and learning of an open-loop controller (Section 2.4).

2.1. Open-loop stable manipulation

A system described by the differential equation $\dot{x} = f(x, u)$ can be linearized along a trajectory x^* caused by input $u^*(t)$:

$$\frac{d\bar{x}}{dt} = A^*(t)\bar{x} \tag{1}$$

with

$$A^*(t) = \left.\frac{\partial f}{\partial x}\right|_{x^*(t),u^*(t)} \tag{2}$$

where $\bar{x}(t) = x(t) - x(t)^*$ is the state error. For the ease of notation, the time dependency of variables is occasionally dropped if it is unambiguous to do so. For example $A^*(t)$ will be written as A^*.

If both trajectory and input are cyclic with period t_f, stability can be assessed by discretizing the system using a time step t_f. To be able to draw upon the research in stability of limit cycles, note

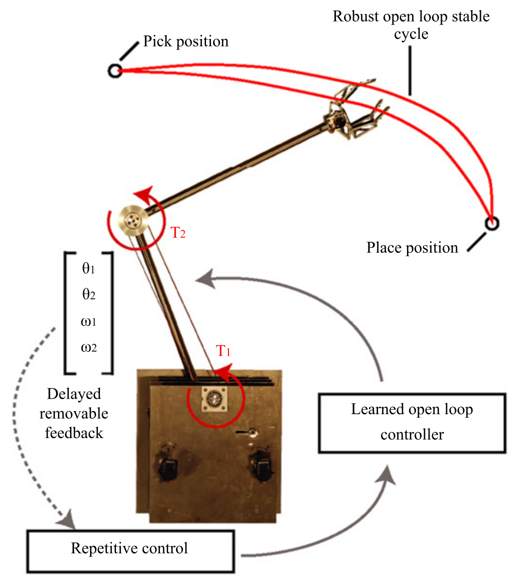

Fig. 1. This figure shows the top view of the concept of robust open-loop stable manipulation. We first optimize a cycle that stands still at the pick and place positions for open-loop stability. Next, we check the cycle for robustness to model uncertainty. Then, using repetitive control on the robotic arm, we learn an open-loop controller that tracks the cycle. After the learning, the open-loop controller performs the task without any feedback.

that such discretization is the same as a Poincaré map of the system with the time appended to the state vector. The Poincaré section is then taken as $t = t_f$, and the time is reset to 0 after crossing this section. Previously (notably in [3,6]), verifying stability was done using the eigenvalues of the linearized discrete system. But that approach does not allow incorporating model uncertainty in the stability analysis.

To obtain a method that does allow uncertain models, we use a quadratic Lyapunov function, $J = \bar{x}^T M(t)\bar{x}$, with positive definite $M(t)$. The idea is that for a stable system, an $M(t)$ can be found such that the norm J is always decreasing over time. For cyclic systems this means the following two constraints should be satisfied (cf. [13]):

$$M(t)A(t)^* + \dot{M}(t) + A(t)^{*T}M(t) \prec 0, \quad \forall t \in [t_0, t_f] \tag{C1}$$

$$M(t_f) - M(t_0) \succ 0 \tag{C2}$$

where $\prec$ and $\succ$ are used to indicate negative/positive definiteness respectively, $\dot{M}$ is the time derivative of M and the subscripts 0 and f denote initial/final time. The first of these constraints ensures that the Lyapunov function is decreasing at each time instant. The second constraint makes sure that it becomes stricter after each cycle, i.e., that having the same error ($\bar{x}$) as a cycle before means that the Lyapunov function has increased. Note that only one of the two inequalities needs to be strict in order for stability to hold.

When there are model inaccuracies, two changes occur that make the above conditions invalid. First, $A^*(x^*(t))$ is no longer accurate when in state $x^*(t)$. Second, when using a fixed open-loop controller on an uncertain system, the trajectory is not fully predictable, so in general $x(t) \neq x^*(t)$, when using the input $u^*(t)$. In the next section we will outline our approach to solve these two issues.

2.2. Robust open-loop approach

To find motions that are open-loop stable even when the model is not accurately known, we will focus on input affine systems