2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Open Loop Stable Control in Repetitive Manipulation Tasks

Michiel Plooij*, Wouter Wolfslag* and Martijn Wisse
Delft University of Technology

Abstract—**Most conventional robotic arms depend on sensory feedback to perform their tasks. When feedback is inaccurate, slow or otherwise unreliable, robots should behave more like humans: rely on feedforward instead. This paper presents an approach to perform repetitive tasks with robotic arms, without the need for feedback (i.e. the control is open loop). The cyclic motions of the repetitive tasks are analyzed using an approach similar to limit cycle theory. We optimize open loop control signals that result in open loop stable motions. This approach to manipulator control was implemented on a two DOF arm in the horizontal plane with a spring on the first DOF, of which we show simulation and hardware results. The results show that both in simulation and in hardware experiments, it is possible to create open loop stable cycles. However, the two resulting cycles are different due to model inaccuracies. We also show simulation and hardware results for an inverted pendulum, of which we have a more accurate model. These results show stable cycles that are the same in simulation and hardware experiments.**

I. Introduction

The vast majority of robotic manipulators require sensory feedback in order to perform their tasks. Humans, on the other hand, use both feedback and feedforward (or open loop) control when controlling their body [1]. Using feedforward allows humans to control their body despite having large time delays (typically 150 ms [2, 3]). Although robots have faster feedback loops, feedforward still has advantages. First, feedforward can anticipate on future states of the system, second, is can offer cheaper control when the cost is critical, and third, it is suitable for systems with slow and imprecise feedback, such as camera based feedback.

In a previous study, we showed the remarkable result that sensitivity to some modeling inaccuracies can be eliminated by choosing the right feedforward controller [4]. This result was similar to observations in humans, who minimize the influence of uncertainty on the final position of feedforward controlled movements [5]. Where the previous study only considered short motions (one second), this paper takes those ideas a step further by considering long term stability of open loop controlled robotic arms (see Fig. 1), inspired by (human) stable walking motions.

One of the commonly cited disadvantages of open loop controllers is that they cannot directly compensate for perturbations, since those perturbations are not fed back into the controller. Since most tasks of robotic arms are repetitive, we propose to view them as cycles and consider the inherent stability of those cycles. If the trajectories we perform are

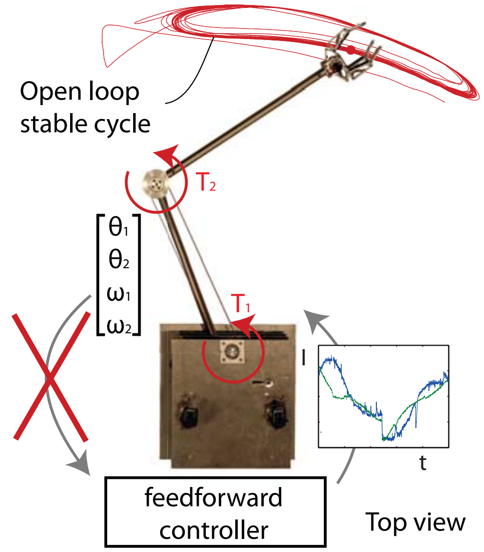
Open loop
stable cycle
T_2
$\begin{bmatrix}\theta_1\\ \theta_2\\ \omega_1\\ \omega_2\end{bmatrix}$
T_1
I
t
feedforward
controller
Top view

Fig. 1. This figure shows the top view of the concept of open loop stable manipulation for our two DOF robot setup. Since there is no feedback available, the torque signal is a function of time only. Using numerical optimizations, we find torque signals that result in open loop stable cycles and allow the robot to perform repetitive tasks. Both the path displayed and the current signals are obtained from hardware experiments.

stable, perturbations will simply decay over time, without the need for sensory feedback.

Other researchers have already used the inherent stability of specific motions in order to create functional robots. Those robots can be split into two groups: robots with only limited state feedback and robots without any state feedback.

A well known example of robots with limited state feedback is given by Schaal and Atkeson [6], who studied open loop stable juggling with a robotic arm. In their case, open loop means that the state of the ball is not used as an input for the controller, but the arm itself is position controlled. Other examples include a timed position controlled swing leg retraction to stabilize running [7] and rope turning without measuring the state of the rope [8].

The group that is more related to this study is the group of robots without any state feedback. The most striking result in this group was obtained by McGeer, who introduced the concept of passive dynamic walking [9].Those walkers do not have motors and thus they do not use any feedback control, while their walking motion is stable. The stable walking motions do not rely on the motion being stable at each point

[*] These authors contributed equally to this paper

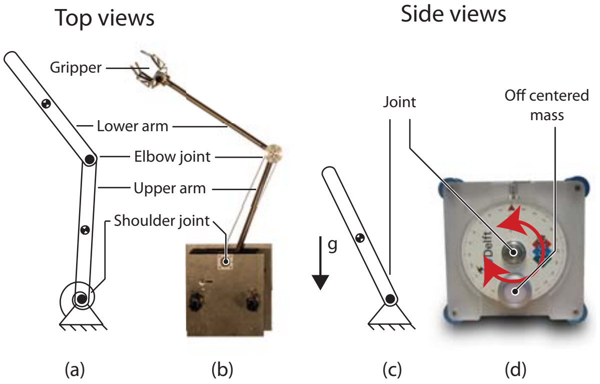

Fig. 2. The four configurations we studied: (a) a simulation model of a two DOF robotic arm, (b) a two DOF robotic arm, (c) a simulation model of an inverted pendulum and (d) an inverted pendulum.

in time, rather they work due to the existence of stable cyclic motions, called limit cycles. Such cycles were later on used in so called limit cycle walkers in combination with feedback control [10, 11, 12]. The work most strongly related to this paper is that by Mombaur et al. [13, 14]. They found stable open loop controllers for walking and running robots by optimizing the open loop controllers for both stability of the motion and energy consumption. These results on a variety of systems indicate that open loop stable control can be an effective approach for robotic arms as well.

The goal of this paper is to introduce a new approach in robot manipulator control: *open loop stable manipulation* and to show the first results using this approach. We show results both in simulation studies and in hardware experiments.

The remainder of this paper is structured as follows. Section II explains the configurations we studied and the optimization method we used. Section III shows the results for the two DOF robotic arm, including a description of the simulation model, the simulation results and the hardware results. Section IV shows the results for the inverted pendulum, including a description of the simulation model, the simulation results and the hardware results. Finally, the paper ends with a discussion in Section V and a conclusion in Section VI, where we will conclude that it is possible to perform manipulation tasks with an open loop controller by performing open loop stable cycles.

II. Methods

We studied open loop stable task execution of robotic arms by optimizing the open loop controller such that the task is performed by a stable cyclic motion. In this section, we discuss the configurations we studied and the optimization method.

A. Configurations

We studied four configurations, which are also shown in Fig. 2:

(a) **A simulation model of a two DOF robotic arm**. On this model we optimized the open loop controller such that the arm makes stable cyclic motions while performing the specified task.

(b) **A prototype of a two DOF robotic arm**. We implemented the controller obtained in (a) on a two DOF robotic arm to test how the controller performs on a real system. The results will show that the open loop controller generates stable cycles on the robot, but does not converge to the same trajectory as in (a). In Section V-B, we will argue that this is caused by a bending of the second DOF due to gravity.

(c) **A simulation model of an inverted pendulum**. We used the same techniques as used in (a) to obtain open loop stable motions of an inverted pendulum. We show that on this system, the results have an intuitive explanation.

(d) **A prototype of an inverted pendulum**. We implemented the controller obtained in (c) on an inverted pendulum to test how the controller performs on a real system for which an accurate model is known.

B. Optimization method

The stability of limit cycles can be assessed with a number of different measures, such as Lyapunov stability [15] or contraction analysis [16, 17]. We used the classic notion of Poincaré maps [15] to find the stability of the open loop controlled motions, which we will now explain.

Consider a non-linear system described by the following differential equation:

$$\dot{x} = f(x, u(t)) \tag{1}$$

Since we use open loop control, we can consider the time as an extra state. Using transverse coordinates [18, 19, 20] with the time as phase variable in this appended state space is the same as using error dynamics in the original state space. This means that we can set a Poincaré section at $t = t_f$. We calculated the error dynamics along the trajectory $x^*(t)$ that results from the input $u^*(t)$. Since the controller is open loop, the error dynamics are simply given as the difference between the current state and the trajectory state, both at the same time:

$$\dot{x}^* = f(x^*, u^*(t)) \tag{2}$$

$$\delta x = x - x^* \tag{3}$$

$$\dot{\delta x} = f(x, u^*(t)) - f(x^*, u^*(t)) \tag{4}$$

Linearizing the dynamics along the trajectory results in:

$$\dot{\delta x} = \frac{\partial f(x^*, u^*(t))}{\partial x}\delta x \tag{5}$$

$$= A^*(t)\delta x \tag{6}$$

Where A^* is the linearized system matrix, and δx the error-state. We constrained the motions to be cyclic with period t_f (see below), which results in a cyclic $A^*(t)$ with the same period. Since system (6) is linear, we can take the state transition matrix Ψ from $t = t_0 = 0$ to $t = t_f$. Because we know $A^*(t)$ analytically, we can find Ψ by numerically computing the solution to the following initial value problem:

$$\dot{\Psi} = A^*(t)\Psi, \ \Psi(0) = I \tag{7}$$

Similar to the monodromy matrix in Poincaré map analysis of limit cycles, the motion is stable if the eigenvalues of Ψ have an absolute value smaller than one:

$$|\lambda\left(\Psi(t_f)\right)|_{max} < 1 \tag{8}$$

To find a trajectory, we now use an optimal control approach similar to [13]. We used the above condition as a constraint rather than to minimize the left hand side of it, because of two reasons. First, the above condition specifies the convergence rate of the limit cycle, and does not specify other stability related factors, such as basin of attraction [21] and robustness against model uncertainties. Second, in practice other performance issues are also of concern, such as energy consumption and speed. We chose to use the integral of the squared input as objective, resulting in the following optimization:

$$\underset{u(t)}{\text{minimize}} \quad \int_{t_0}^{t_f} u(t)^2 dt \tag{9}$$

$$\begin{aligned} \text{subject to} |\lambda\left(\Psi(t_f)\right)|_{max} &< 0.9 \\ |u(t)| &< u_{max} \\ x(0) &= x_{pick} \\ x(t_1) &= x_{place} \\ x(t_f) &= x_{pick} = x(0) \end{aligned} \tag{10}$$

Where $u(t)$ is the input, u_{max} is the maximum input and x_{pick} and x_{place} are the pick and place states (see Section III-B). The constraint that the final state is equal to the initial state causes the resultant motion to be cyclic.

In case of a setup with multiple motors, we used the integral of the sum of squared inputs as cost function. We used this cost function because it is often used in other control applications. Furthermore, the resulting controllers are relatively smooth, whereas a time-optimal controller would be bang-bang. Such a bang-bang controller is undesirable in robot experiments, because it is more likely to be affected by unmodeled effects such as backlash. The optimization is performed using the optimal control package GPOPS [22] in Matlab.

III. Two DOF manipulator

We implemented open loop stable manipulation on a SCARA type arm with two DOFs: two revolute joints moving two links in the horizontal plane (see Fig. 2b and 3). Since there is no spring present on the second joint of the arm, an open loop controlled motion must depend on dynamic effects for stabilization. Specifically, the zero input does not lead to a stable result. Since the dynamic effects are highly non-linear, it is not obvious whether stable motions exist. This section starts with a system description, including a description of the robotic arm and the simulation model. Next, we explain the task the arm has to perform, followed by simulation results and hardware results.

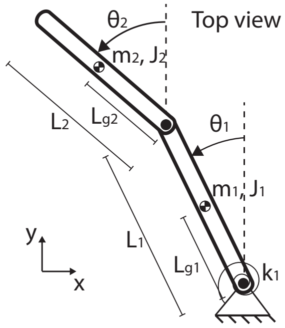

Fig. 3. Top view of the two DOF system with a linear spring on the first joint. The second joint is actuated through a parallel mechanism (not shown in this figure), such that the angle of the second arm is an absolute angle. The friction acts on the absolute angles of the joints.

TABLE I
The model parameters of the two DOF arm.

Parameter	Symbol	Value	Unit
Coulomb friction	μ_{c1}, μ_{c2}	0.0481, 0.0218	Nm
Viscous friction	μ_{v1}, μ_{v2}	0.03, 0.03	Nms/rad
Torque dependent friction	μ_{tf1}, μ_{tf2}	21.87, 31.91	%
Inertia	J_1, J_2	0.0233, 0.0871	kgm^2
Mass	m_1, m_2	0.809, 1.599	kg
Length	l_1, l_2	0.410, 0.450	m
Position of COM	l_{g1}, l_{g2}	0.070, 0.325	m
Motor constant	k_{t1}, k_{t2}	26.7, 28.1	mNm/A
Gearbox ratio	n_1, n_2	1:54, 1:110	rad/rad
Spring stiffness	k_1	1.6	Nm/rad

A. System description

Fig. 1 shows a picture of the two DOF robotic arm [23]. The DOFs are created by two 18x1.5mm stainless steel tubes, connected with two revolute joints, with a spring on the first joint. A mass of 1 kg is connected to the end of the second tube, which represents the weight of a gripper with product. The motors are placed on a housing and AT3-gen III 16mm timing belts transfer the torques within the housing. The joints are actuated by Maxon 60W RE30 motors with gearbox ratios of respectively 18:1 and 1:66. The timing belts provide an additional transfer ratio of 3:1 on the first joint and 5:3 on the second joint.

We used the TMT method [24] to obtain the equations of motion of the simulation model of this arm, which are too long to include in this paper. The model includes 19 parameters, which are listed in Table I. We included three types of frictional losses: Coulomb friction, viscous friction and torque dependent gearbox friction. Torque dependent gearbox friction is less commonly used than the other two, however, from the parameter values obtained through a system identification of the motor it is clear that this type of friction is not negligible (see Table I). The way we implemented it is similar to the force dependent friction term in [25]. The friction in a joint is equal to:

$$T_f = -\mu_v \cdot \omega - \text{sign}(\omega) \cdot (\mu_c + \mu_t \cdot |T|) \tag{11}$$

for $\omega \neq 0$

$$T_f = -\min(\mu_c + \mu_t \cdot |T|; |T|) \cdot \text{sign}(T) \tag{12}$$

for $\omega = 0$.

ω is the velocity of the joint, T is the torque exerted by the motor on the joint, and μ_v, μ_c and μ_t are the coefficients of viscous, Coulomb and torque dependent friction respectively.

The simulation model includes a DC motor. The torque applied by the DC motor on the joint is equal to:

$$T = n \cdot k_t \cdot I \tag{13}$$

Where k_t is the motor constant, n is the gearbox ratio and I is the current through the motor. The current through the motor is constrained to 1 A (see eq. 10).

Coulomb friction and torque dependent friction introduce discontinuities in the equations of motion, which are difficult for GPOPS to handle. Therefore, we optimized the open loop controller on a model with only viscous friction, and added a torque afterwards to compensate for the Coulomb friction and torque dependent friction. Such a compensation can only be done when it does not effect the stability of the cycle, so when A^* is independent of both the compensated friction and the input. Both the Coulomb friction and the torque dependent friction depend on the state through a sign function, which is a piecewise constant function. Since A^* is the result of linearizing along the state, A^* does not depend on those friction terms (neglecting the always stabilizing effect of the discontinuity in the sign function). To make A^* independent of the input, we consider momenta instead of velocities. The equations of motion then become:

$$\dot{x} = f(x) + Bu(t) \tag{14}$$

With x, the state consisting of positions and momenta, and B a constant matrix. Following the definition in eq. (6), we see that A^* is therefore independent of u. Such a transformation is possible for many mechanical systems. Although such a transformation is not necessary, our specific optimization was faster with the transformation. For easier interpretation, we will show the velocities in the results.

B. Task

The robotic arm has to perform a pick-and-place task. The important task parameters are the pick state, the place state and the time per stroke. We show the results of the optimization for a motion which starts at $t = 0$ at the pick state x_{pick}, goes to the place state x_{place} at t_1 and then returns to the pick state at t_f with

$$x_{pick} = \begin{bmatrix} -0.7 \text{ rad} \\ -0.85 \text{ rad} \\ 0 \text{ rad/s} \\ 0 \text{ rad/s} \end{bmatrix}; x_{place} = \begin{bmatrix} 0.7 \text{ rad} \\ -0.3 \text{ rad} \\ 0 \text{ rad/s} \\ 0 \text{ rad/s} \end{bmatrix} \tag{15}$$

Where x is the vector consisting of the positions of the first and second arm, and their respective angular velocities. t_1 and t_f are free parameters in the optimization, but bounded as follows:

$$0.1 \text{ s} \leq t_1 \leq 1.2 \text{ s} \tag{16}$$

$$0.1 \text{ s} \leq t_f - t_1 \leq 1.2 \text{ s} \tag{17}$$

The goal is to find a path that statisfies the task constraints (that also include a stability-enforcing constraint) according to eq. (10), and then minimize the integral of the square of the current (see eq. (9)). Although we only show the result of one specific task, similar results were obtained using other task constraints. In none of these cases multi-starts or tuned initial conditions were needed, even though the optimization is non-convex.

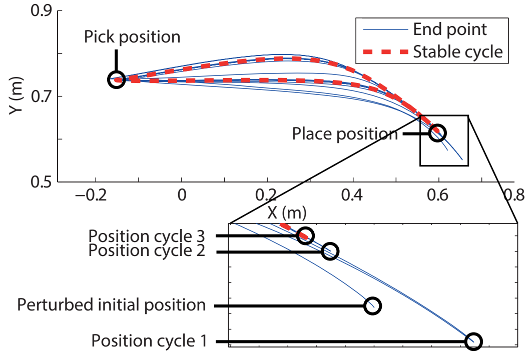

Fig. 4. This figure shows the plot of the end point of the arm in simulation. The plot shows the stable cycle from simulation (thick dotted red line) and the motion of the robotic arm that starts from a perturbed position and converges to the stable cycle (thin blue line).

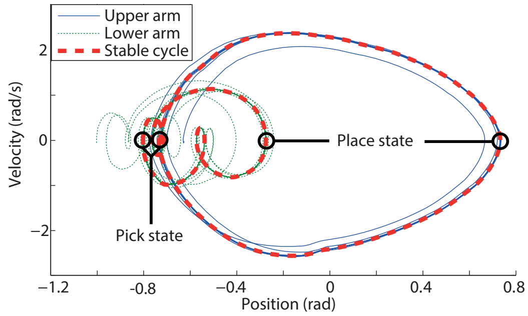

Fig. 5. This figure shows the state space plot of the simulation data for the two DOF robotic arm. The plot shows the stable cycle (thick dotted red lines) and a motion that starts from a perturbed position and converges to the open loop stable cycle (thin lines).

C. Simulation results

Fig. 4 shows the position of the gripper in the workspace. The gripper does not start at the stable cycle, but converges to it. The sharp corners in the motion are the pick and place positions. Similar results were obtained for different pick and place positions.

Fig. 5 shows an example motion converging to the stable cycle in state space. The motion starts at a distance from the pick state $x_{start} = x_{pick} + [0.07; -0.15; 0; 0]$ and converges to the open loop stable cycle.

Fig. 6 shows the motion and current profile that result from the optimization. In Fig. 6a and 6b, we see that the motion starts and ends at the pick state, while at t $\approx$ 1.2 s, the arm is at the place state.

D. Hardware results

Fig. 7 shows the position of the gripper in the workspace. It clearly shows that the gripper does not start at the stable cycle, but converges to the cycle. Comparing the hardware results (Fig. 7) with the simulation results (Fig. 5) leads to the

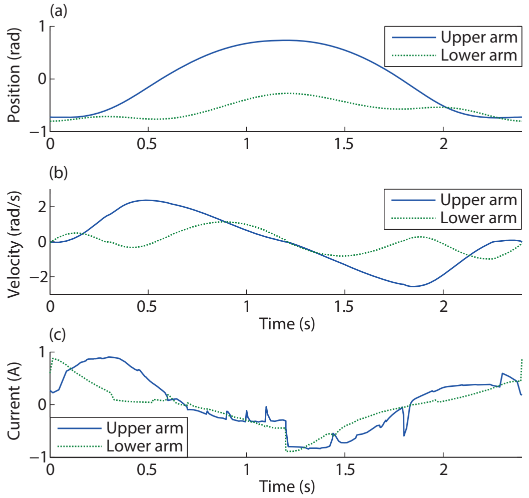

Fig. 6. This figure shows the simulation data for the two DOF robotic arm as function of time. The figure shows the positions of the joints (a), the velocities of the joints (b) and the torques about the joints (c).

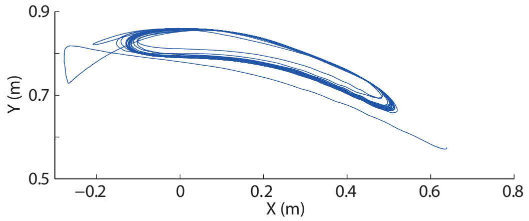

Fig. 7. This figure shows the plot of the gripper in hardware experiments. The plot shows the motion of the robotic arm that starts from a perturbed position and converges to the stable cycle.

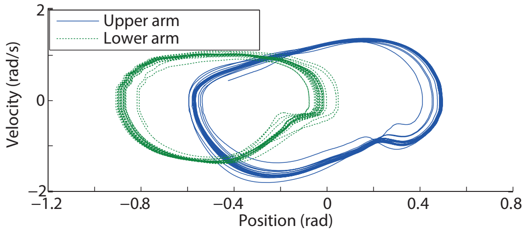

Fig. 8. This figure shows the state space plot of the hardware experiments on the two DOF robotic arm. The plot shows the motion of the robotic arm that starts from a perturbed state and converges to the open loop stable cycle. In order to obtain a smooth graph, the velocity data is filtered with a fifth order Butterworth filter with the cutoff frequency at 10 Hz.

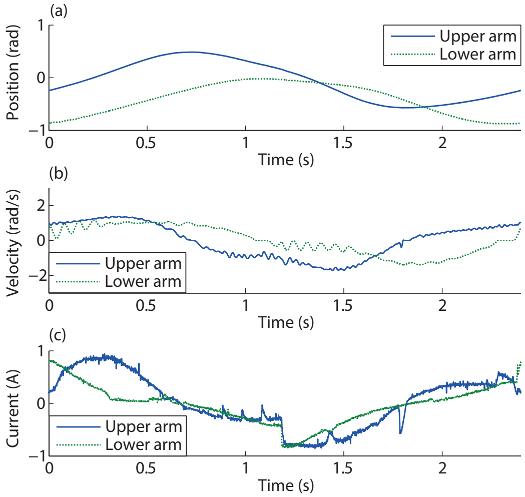

Fig. 9. This figure shows the data of a typical motion of the hardware experiments on the two DOF robotic arm as function of time. The figure shows the positions of the joints (a), the velocities of the joints (b) and the current through the motors (c).

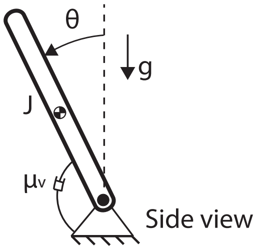

Fig. 10. Side view of the one DOF system with linear viscous friction and gravity.

conclusion that although both the simulation and hardware results show convergence to a stable cycle, the stable cycles themselves are different. In Section V-B we will argue that the difference is probably caused by a bending of the second DOF due to gravity. In Section IV, we will show that for a simpler system (i.e. an inverted pendulum), our model is accurate enough to predict the exact cycle.

Fig. 8 shows the motion of the arm in state space. We see that the motion converges to a stable cycle in state space. When comparing Fig. 8 with Fig. 5, we see that range of positions of the upper arm is smaller in hardware experiments, and that the range of positions of the lower arm is larger in hardware experiments.

Fig. 9 shows the time series of a typical cycle of the robotic arm in hardware experiments. Again we see that the hardware results differ from the simulation results. In Fig. 9b, we notice that the velocity signals show a vibration at approximately 10 Hz. This vibration is caused by the elasticity of the timing belts between the motors (with encoders) and the joints.

The accompanying video shows a demonstration of the disturbance recovery of the two DOF arm. This video also indicates that the basin of attraction is large. In Section V-B, we will show this basin of attraction in more detail.

IV. Inverted pendulum

In the previous section, we showed the results of a two DOF SCARA type arm. These results show that pick and place motions can be performed in an open loop stable manner. However, there was a difference between simulation and hardware results due to model inaccuracies. In this Section we move to a simpler system, allowing us to both give an intuitive explanation for how the stabilization works, and indicate that with a more accurate model hardware experiments and simulation can be made to match. For this purpose an inverted pendulum is used (see Fig. 10).

A. System description

Fig. 2d shows a picture of the inverted pendulum setup we used. The pendulum consists of a disk with an off-centered mass, which is connected to a DC motor without a gearbox

TABLE II
THE MODEL PARAMETERS OF THE TWO DOF ARM. THE VALUES ARE OBTAINED THROUGH A SYSTEM IDENTIFICATION OF THE INVERTED PENDULUM.

Parameter	Symbol	Value	Unit
Gravitational term	c_1	112.9	s^{-2}
Motor parameters term	c_2	28.0	$V^{-1}s^{-2}$
Damping term	c_3	-1.8	s^{-1}

in between. This direct drive actuation results in low friction, which makes the system easier to model. The DC motor is voltage controlled with a maximum voltage of 5 V (see eq. 10).

The differential equation for this system is

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \omega \\ c_1 \sin(\theta) + c_2 U + c_3 \omega \end{bmatrix} \tag{18}$$

Where, U is the input voltage, and c_1, c_2 and c_3 are the model parameters, which we identified through a system identification and are listed in Table II. In this model, c_1 can be seen as the gravitational term, c_2 as the motor parameters term and c_3 as the damping term, which includes the back-emf term of the motor. All these terms have the inertia of the pendulum included.

We again used GPOPS to optimize an open loop controller such that we obtain an open loop stable motion for the task described below.

B. Task

The task we will look at is a motion with the initial and final position of the arm both at $\theta = 1/5\pi$ (nearly upright position). Simply not moving will not result in stability, since by itself this is an unstable position. So at first, it seems impossible to find an open loop stable solution. However, this is possible when we allow the pendulum to swing to the region between $^1/_2\pi$ and $^3/_2\pi$ (around the lower equilibrium) during the motion. The goal is to find a path that is stable according to (8), and then minimize the integral of the square of the voltage (see eq. (9)). We limit the duration of the motion to 1.2 seconds.

C. Simulation results

Fig. 11 shows the result of the optimization, which can be explained intuitively. Since eq. (18) is linear in the velocity and the input, the A^*-matrix of the pendulum only depends on the position. That is, being in a state with θ between $-\pi/2$ and $\pi/2$ has a destabilizing effect, and being in a state with θ outside that range has a stabilizing effect. Loosely speaking, a stable motion requires that the stabilizing effects compensate the destabilizing ones. This means that the system should spend enough time in sufficiently stabilizing positions to counter the time it spends in the destabilizing positions. In Fig. 11, we see that the pendulum moves directly from the destabilizing initial position to stabilizing positions, where it spends most of the time before moving back at the end of the motion. We use the condition in eq. (8) to evaluate if the stabilizing effects are compensating the destabilizing effects.

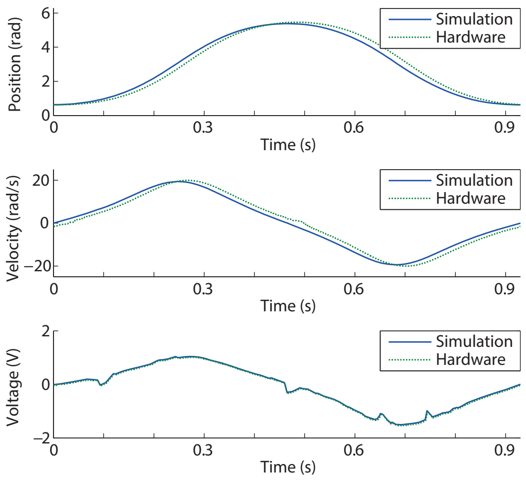

Fig. 11. This figure shows the data for the single pendulum in the vertical plane as function of time. The figure shows the position of the joint (a), the velocity of the joint (b), and the torque about the joint (c) for both the simulation as the hardware experiments.

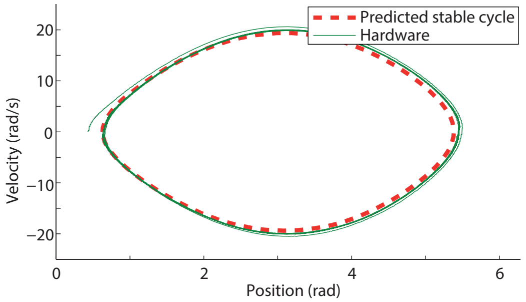

Fig. 12. This figure shows the state space trajectory for the single pendulum in the vertical plane. The plot shows the stable cycle as determined in simulation (thick dotted line) and a motion on the hardware setup that starts from a perturbed position and converges to the stable cycle (thin line).

D. Hardware results

Fig. 12 shows the results of applying the input signal found in simulation on the hardware setup. It shows a motion being initialized in a perturbed position, after which it converges towards a cycle close to the one predicted by simulation. Fig. 11 shows that the motion over time after convergence is the same in simulation and hardware experiments. This shows that when an accurate model is available, the method we used finds a controller that performs the desired task on the real system. Extending the method to allow task performance even when accurate models are not readily available is an important next step in the research on open loop stable manipulation.

V. DISCUSSION

A. Model mismatch

For the two DOF robotic arm, the results from simulation (see Fig. 4, 5 and 6) are clearly different from those of the hardware experiments (see Fig. 7, 8 and 9). However, both simulation and hardware experiments show convergence to a stable cycle. These results show that the specific cycle the system converges to is sensitive to unmodeled behavior. For the inverted pendulum, we showed the results from simulation and hardware experiments are the same. This is due to the fact that the inverted pendulum is easier to model accurately.

There are three possible ways to reduce errors due to unmodeled behavior and thereby improve the control performance. First, the model of the system can be extended to include more of the unmodeled dynamics. Since modeling inaccuracies will always exist, a second approach would be to include the sensitivity to modeling inaccuracies in the optimization as in [4]. However, since sensitivity to modeling inaccuracies cannot always be reduced [4], we expect the best results from the third approach: tuning or learning open loop stable cycles online.

B. Basin of Attraction

In this paper we focused on the calculation of open loop stability, which is a minimal requirement for making open loop manipulation work, but it gives little information about the rejection of realistic (i.e. finite) disturbances. Therefore, we simulated the arm using a grid of initial positions in order to see if they converge to the intended stable cycle. Fig. 13 shows this data, which is a slice of the 4D basin of attraction. This shows that the majority of initial positions (within the mechanical limits of the arm) result in convergence to the cycle. This result means that precise initial positioning of the arm is not required for converging to the intended cycle.

Interestingly, the basin of attraction analysis shows that all initial positions converge to the same cycle, although in some cycles, the second joint has rotated for exactly one or multiple revolutions. While implementing several feedforward controllers on the robotic arm, we found that most feedforward controllers (including the one shown in this paper) result in two stable cycles: one with negative θ_2 and one with positive θ_2. We expect that this is caused by a bending of the arm due to gravity, which is larger around $\theta_2 = \pm\frac{\pi}{2}$ rad than around $\theta_2 = 0$ rad. We suspect that this difference between the model and the arm is the main cause for the mismatch of the cycles in simulation and in hardware experiments. In order to improve the prediction of the cycle by the model, a stiffer arm can be used or this bending can be modeled. A second cause of the model mismatch could be the friction, which is hard to model in general.

C. Implications

The idea of using open loop stability of periodic motions to analyze repetitive manipulation tasks is a new approach in robot manipulator control. It allows for robotic arms to be controlled when feedback is too slow (e.g. using camera feedback or control over great distance), too imprecise (e.g. due to cheap, noisy sensors [26]) or even impossible (e.g. in micro-scale applications or due to radiation), or when the input is limited and planning is required.

There is no fundamental reason why stable cycles would not exist in robotic arms with more DOFs, or for more complex tasks (e.g. obstacles or interactions with the environment). However, we expect that it will be more difficult to find such cycles and maybe impossible to find cycles that include the pick- and place positions. Finding trajectories can be made easier by tuning the dynamics of the system, such that all trajectories are stable, i.e. the system is contractive. Such tuning could be done by adding springs on all joints or changing the mass distribution.

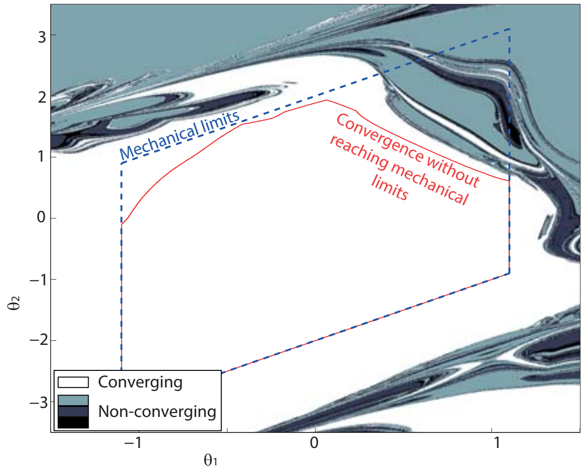

Fig. 13. This figure shows the basin of attraction of the open loop stable cycle for different initial positions. The white region depicts all initial positions that result in convergence to the cycle. The blue dotted line depicts the mechanical limits of the robotic arm. All initial positions within the red line result in convergence while staying within the mechanical limits. The non-converging initial positions are divided into three groups. First, light blue depicts convergence to the intended cycle with an error in θ_2 of 2π rad; second, dark blue depicts an error in θ_2 of 4π rad and third, black depicts an error in θ_2 of 6π rad or more. So all initial positions converge to the same cycle, although in some cycles, the second joint has rotated for exactly one or multiple revolutions.

D. Applicability

The concept of open loop stable manipulation as presented in this paper is not fully applicable yet, since it consists of finding one stable cycle. In practice, tasks consist of moving between multiple positions in a certain (not necessarily predefined) order. Since the computations are too complex to perform online, we see two approaches to make open loop stable manipulation fully applicable in the future.

The first approach is to move in a stable periodic motion which covers most of the task space. Whenever necessary, the controller can make open loop controlled deviations from this cycle to the specified positions after which the manipulator returns to the main cycle.

The second approach is to divide the space into a grid and build a library of feedforward controllers for moving between the grid positions. The controller can then create various stable cycles by combining multiple trajectories from this library into a stable cycle that tracks the specified positions.

VI. Conclusions

In this paper we introduced an approach to control robotic arms, without the need for state feedback. We conclude that this approach is promising for robotic arms that perform repetitive position tasks without feedback. Small disturbances on the state of the system will decay over time and the robotic arm will asymptotically return to its original trajectory. Both simulation and hardware experiments show convergence to an open loop stable cycle, but the cycles they converge to are not the same, probably caused by a bending of the second DOF due to gravity. We expect that this problem can be solved by online learning of the stable cycles.

ACKNOWLEDGEMENT

This work is part of the research programme STW, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

REFERENCES

[1] M. Desmurget and S. Grafton, "Forward modeling allows feedback control for fast reaching movements," *Trends in cognitive sciences*, vol. 4, no. 11, pp. 423–431, 2000.

[2] S. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system," *Nature*, vol. 381, no. 6582, pp. 520–522, 1996.

[3] P. Cordo, L. Carlton, L. Bevan, M. Carlton, and G. K. Kerr, "Proprioceptive coordination of movement sequences: role of velocity and position information," *Journal of Neurophysiology*, vol. 71, no. 5, pp. 1848–1861, 1994.

[4] M. Plooij, M. de Vries, W. Wolfslag, and M. Wisse, "Optimization of feedforward controllers to minimize sensitivity to model inaccuracies," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, pp. 3382–3389.

[5] C. M. Harris and D. M. Wolpert, "Signal-dependent noise determines motor planning," *Nature*, vol. 394, no. 6695, pp. 780–784, 1998.

[6] S. Schaal and C. Atkeson, "Open loop stable control strategies for robot juggling," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, may 1993, pp. 913 –918 vol.3.

[7] A. Seyfarth, H. Geyer, and H. Herr, "Swing-leg retraction: a simple control model for stable running," *Journal of Experimental Biology*, vol. 206, no. 15, pp. 2547–2555, 2003.

[8] C. H. Kim, K. Yonekura, H. Tsujino, and S. Sugano, "Physical control of the rotation of a flexible objectrope turning with a humanoid robot," *Advanced robotics*, vol. 25, no. 3-4, pp. 491–506, 2011.

[9] T. McGeer, "Passive dynamic walking," *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.

[10] Y. Hürmüzlü and G. Moskowitz, "The role of impact in the stability of bipedal locomotion," *Dynamics and Stability of Systems*, vol. 1, no. 3, pp. 217–234, 1986.

[11] A. Goswami, B. Espiau, and A. Keramane, "Limit cycles and their stability in a passive bipedal gait," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 246–251.

[12] D. Hobbelen and M. Wisse, "Limit cycle walking," 2007.

[13] K. D. Mombaur, R. W. Longman, H. G. Bock, and J. P. Schlöder, "Open-loop stable running," *Robotica*, vol. 23, no. 1, pp. 21–33, Jan. 2005.

[14] K. D. Mombaur, H. G. Bock, J. P. Schlöder, and R. W. Longman, "Open-loop stable solutions of periodic optimal control problems in robotics," *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 85, no. 7, pp. 499–515, 2005.

[15] S. Strogatz, "Nonlinear dynamics and chaos: with applications to physics, biology, chemistry and engineering," 2001.

[16] W. Lohmiller and J. Slotine, "Contraction analysis of nonlinear systems," Ph.D. dissertation, Massachusetts Institute of Technology, Dept. of Mechanical Engineering, 1999.

[17] I. Manchester and J. Slotine, "Contraction criteria for existence, stability, and robustness of a limit cycle," *arXiv preprint arXiv:1209.4433*, 2012.

[18] A. Banaszuk and J. Hauser, "Feedback linearization of transverse dynamics for periodic orbits," *Systems & control letters*, vol. 26, no. 2, pp. 95–105, 1995.

[19] A. Shiriaev, L. Freidovich, and S. Gusev, "Transverse linearization for controlled mechanical systems with several passive degrees of freedom," *Automatic Control, IEEE Transactions on*, vol. 55, no. 4, pp. 893–906, 2010.

[20] I. Manchester, "Transverse dynamics and regions of stability for nonlinear hybrid limit cycles," *arXiv preprint arXiv:1010.2241*, 2010.

[21] D. Hobbelen and M. Wisse, "A disturbance rejection measure for limit cycle walkers: The gait sensitivity norm," *Robotics, IEEE Transactions on*, vol. 23, no. 6, pp. 1213–1224, 2007.

[22] A. Rao, D. Benson, C. Darby, M. Patterson, C. Francolin, I. Sanders, and G. Huntington, "Algorithm 902: Gpops, a matlab software for solving multiple-phase optimal control problems using the gauss pseudospectral method," *ACM Transactions on Mathematical Software*, vol. 37, no. 2, pp. 1–39, 2010.

[23] M. Plooij and M. Wisse, "A novel spring mechanism to reduce energy consumption of robotic arms," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, oct. 2012, pp. 2901 –2908.

[24] R. van der Linde and A. Schwab, "Lecture notes on multibody dynamics b, wb1413," *Delft University of Technology*, 1997/1998.

[25] P. E. Dupont, "The effect of coulomb friction on the existence and uniqueness of the forward dynamics problem," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*. IEEE, 1992, pp. 1442–1447.

[26] A. Kuo, "The relative roles of feedforward and feedback in the control of rhythmic movements." *Motor control*, vol. 6, no. 2, pp. 129–145, 2002.