The effect of the choice of feedforward controllers on the accuracy of low gain controlled robots

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

***Abstract*—High feedback gains cannot be used on all robots due to sensor noise, time delays or interaction with humans. The problem with low feedback gain controlled robots is that the accuracy of the task execution is potentially low. In this paper we investigate if trajectory optimization of feedback-feedforward controlled robots improves their accuracy. For rest-to-rest motions, we find the optimal trajectory indirectly by numerically optimizing the corresponding feedforward controller for accuracy. A new performance measure called the Manipulation Sensitivity Norm (MSN) is introduced that determines the accuracy under most disturbances and modeling errors. We tested this method on a two DOF robotic arm in the horizontal plane. The results show that for all feedback gains we tested, the choice for the trajectory has a significant influence on the accuracy of the arm (viz. position errors being reduced from 2.5 cm to 0.3 cm). Moreover, to study which features of feedforward controllers cause high or low accuracy, four more feedforward controllers were tested. Results from those experiments indicate that a trajectory that is smooth or quickly approaches the goal position will be accurate.**

I. Introduction

High precision in robotics is usually achieved with high feedback gains. However, there are applications in which such high gains are undesirable or infeasible. For instance, in the presence of sensor noise or time delays, high feedback gains will make the robot unstable. A second example of robots in which high gains are undesirable are robots that interact with humans. In such robots, high feedback gains increase the risk of injury. These examples show that it is important to develop alternative techniques to obtain high precisions that work even on robots with limited feedback.

To that end, multiple researchers have taken the idea of feedback limitations to an extremum and have focused on executing tasks with robots without any feedback. A first example is the concept of passive dynamic walking, as introduced by McGeer [1]. Those walkers do not have motors and therefore no feedback control, and still walk with a stable gait. These gaits do not rely on the motion being stable at each point in time, rather they work due to the existence of stable cyclic motions, called limit cycles. Such cycles were later on combined with feedback control in so called limit cycle walkers [2–4]. Mombaur et al. [5, 6] 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.

Control without feedback has also been applied on robotic arms. A well-known example is the work of Schaal and Atkeson [7], 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. In previous work, we showed that it is possible to perform open loop motions with robotic arms that are insensitive to model inaccuracies [8, 9] and to perform open loop stable cycles in which state errors vanish without any feedback [10, 11]. In [11], we optimized trajectories for open loop stability, and used an initial online learning approach to improve the precision of the purely feedforward controlled robot. In these studies, the trajectory itself was effected by the choice of feedforward controller, which was optimized. In the rest of this paper, we consider the feedforward controller and the trajectory to contain the same information, since they can be translated into each other using the model of the robot.

Previous examples show that the most common technique to stabilize robots without feedback is to optimize trajectories for their open loop stability [5, 6, 10, 9]. In practice, a certain amount of feedback will always be available, and therefore the advantages of both control paradigms should be exploited

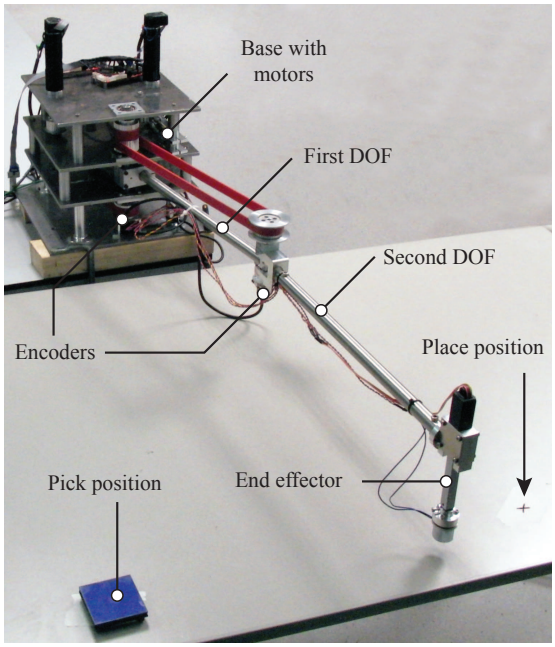

Figure 1. A photograph of the robotic arm we use to test our method: a two DOF SCARA type arm, which has to move between the pick and the place positions.

* These authors contributed equally to this paper

to achieve higher precision [12]. However, it is unclear if trajectory optimization is useful for robots with at least a little feedback.

Therefore, the question we will answer in this paper is: *does the choice of the feedforward controller influence the accuracy of systems with (limited) feedback?* We will answer this question by studying a two degree of freedom (DOF) SCARA type robotic arm in the following way. First, in section II we explain the methods that we used, including the setup and task we study. Then, in section III we introduce the Manipulation Sensitivity Norm (MSN), which we use to estimate the lower and upper bounds of the accuracy of the arm given a certain feedback controller. Next, in section IV we show the results of four alternative controllers that indicate that smooth and goal directed motions result in high accuracy. These results are discussed in section V. And finally in section VI, we will conclude the paper.

II. Methods

In this section we explain the methods we used. First, we explain the systems under consideration, including the controller. Second, we describe the robotic arm that is used as a test case. Third, we discuss the specific task that is studied. And finally, we discuss the feedforward term in the controller.

A. System description

The type of system considered in this paper is a serial chain robotic arm moving in the horizontal plane. The equation of motion of such a system is described by a second order differential equation:

$$\ddot{q} = f(q, \dot{q}) + M^{-1}(q)\tau \tag{1}$$

with q the absolute angles of the links of the robot, $\dot{q}$ and $\ddot{q}$ the angular velocities and accelerations, and τ the motor torques, which are used as control inputs. Note that this system is non-linear due to the Coriolis and centrifugal terms $f(q, \dot{q})$, and the configuration dependent mass matrix M. To control the robot, both a feedback and a feedforward term are used, hence $\tau = \tau_{fb} + \tau_{ff}$.

Because the goal of this paper is to investigate the effect of feedback gain limitations, we structure the feedback controller in such a way that it depends on only one parameter, namely ω, which is the desired natural frequency of the controlled system. For the purpose of constructing the feedback controller from this ω, the system is simplified by neglecting $f(q, \dot{q})$, and decoupling the resulting system by considering only the diagonal entries of the mass matrix at position $q = 0$. In other words, only considering the simplified system

$$\text{diag}(M(0))\ddot{q} = \tau \tag{2}$$

Note that this simplified system is only used to construct the feedback controllers and that the system we study is the non-linear system in Eq. (1). By using diagonal gain matrices K and C, we obtain the following, second order linear differential equations:

$$\text{diag}(M(0))\ddot{q}_i = -Kq - C\dot{q} \tag{3}$$

Because all matrices are diagonal, the differential equations are decoupled, which means that they can be solved separately. Finally, we choose the gains such that the natural frequency of all decoupled parts are set to a desired value (ω), and the damping ratio is set to 1, i.e. critically damped. Therefore, the controller gains are set by solving:

$$\sqrt{\frac{k}{m}} = \omega \tag{4}$$

$$\frac{c}{2\sqrt{km}} = 1 \tag{5}$$

for each part. The natural frequency ω is then used as the parameter to vary the gains. These feedback gains are then used to stabilize the robotic arm around a desired trajectory:

$$\tau_{fb} = -K(q - q_{des}) - C(\dot{q} - \dot{q}_{des}) \tag{6}$$

The feedforward term is simply a torque as function of time: $\tau_{ff}(t)$. This term leads to the desired trajectory: $\ddot{q}_{des} = f(q_{des}, \dot{q}_{des}) + M^{-1}(q_{des})\tau_{ff}(t)$. Because of this relation between feedforward controller and trajectory, we use the two terms interchangeably in this paper.

The goal of this paper is to study the effect of the feedforward controller on the accuracy of a rest-to-rest motion under disturbance. Our approach to studying the effect is to optimize the feedforward controller to minimize or maximize this disturbed accuracy. The optimization is done using single shooting, a basic optimal control approach.

B. Hardware setup

To test our approach, we use a two DOF SCARA type arm [13] (see Fig. 1). This type of arm was chosen as it is the simplest that can perform industrially relevant tasks. The arm consists of two 18x1.5mm stainless steel tubes, connected with two revolute joints. An end effector is connected to the end of the second tube. The motors are placed on a housing and AT3-gen III 16mm timing belts are used to transfer torques within the housing. The joints are actuated by Maxon 60W RE30 motors with gearbox ratios of respectively 66:1 and 18:1. The timing belts provide an additional transfer ratios of 5:4 on both joints. Because the second joint is connected to its motor via a parallelogram mechanism (see [13]), the angle of the second arm is taken as the absolute angle, i.e., relative to the world frame. The end effector acts in the vertical plane and thus its motions do not influence the dynamics of the first and second DOF. The mass of the end effector is incorporated in the inertial terms of the second DOF. The arm is controlled through xPC-target in MATLAB at a frequency of 1 kHz. The parameters of this robotic arm are listed in Table I.

Table I
THE MODEL PARAMETERS OF THE TWO DOF ARM.

Parameter	Symbol	Value	Unit
Damping	μ_{v1}, μ_{v2}	0.2, 0.2	Nms/rad
Inertia	J_1, J_2	0.0233, 0.0312	kgm^2
Mass	m_1, m_2	0.809, 0.784	kg
Length	l_1, l_2	0.410, 0.450	m
Position of COM	l_{g1}, l_{g2}	0.070, 0.195	m
Motor constant	k_{t1}, k_{t2}	25.9, 25.9	mNm/A
Gearbox ratio	g_1, g_2	82.5:1, 22.5:1	rad/rad

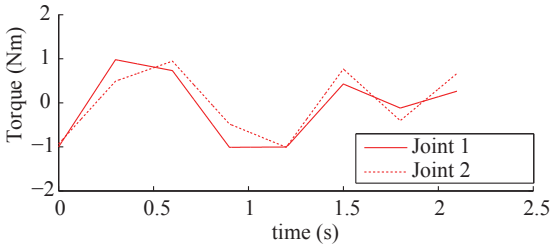

Figure 2. An example of a feedforward controller. The controller is parameterized as a piecewise linear function with the length of every part being 0.3 s. These torques over time are used as decision variables in the optimizations.

C. Task description

We let the manipulator perform a cyclic pick and place motion, with pick and place positions at [-0.2, -0.3] rad and [0.2, 0.4] rad respectively. The time to move between the pick and the place position is 1.05 s. Hence, the total time of one cycle is 2.1 s.

D. The feedforward term in the controller

We test the accuracy resulting from different feedforward controllers: minimization and maximization of the novel Manipulation Sensitivity Norm (see section III) and both smooth and time optimal trajectories (see section IV). The feedforward controller has to be parameterized in order to be able to optimize it.

The optimization schemes parameterize the torque signals for both joints as a piecewise linear signal, with the length of every piecewise part being 0.3 s. The space of such signals is called $\mathcal{U}$. The system states are constrained to be on the pick and place motions at the pick and place times. This also ensures that the motion is cyclic. Finally, the absolute value of both torque signals is bounded by $\tau_{max} = 1$ Nm in order to prevent reaching the actuator limits when when feedback is needed. An example of such a feedforward control signal is shown in Fig. 2.

In the remainder of the paper, the piecewise linear torque signals are optimized for various goals. These goals are expressed as a function $C(\tau_f f)$, which is either maximized or minimized. Combined with the task description, this leads to the following optimization problem:

$$\begin{aligned} \underset{\tau_{ff}(t)\in\mathcal{U}}{\text{minimize}} \quad & C(\tau_{ff}) \\ \text{subject to} \quad & |\tau(t)| \leq \tau_{max} \qquad \forall t \\ & q(0) = q_{pick} \\ & q(1.05) = q_{place} \\ & q(2.1) = q_{pick} \\ & \dot{q}(0) = \dot{q}(1.05) = \dot{q}(2.1) = [0, 0] \end{aligned} \tag{7}$$

III. OPTIMALITY STUDY

In this section we estimate the lower bound and upper bound of the accuracy of the arm, given a certain feedback gain ω. First, we introduce a new measure for disturbance and modeling error rejection, called the Manipulation Sensitivity Norm (MSN). Then, we use this measure to optimize feedforward controllers in simulation, both minimizing and maximizing the MSN. Finally, we apply these controllers on the hardware setup, to test their accuracy.

A. The Manipulation Sensitivity Norm

To judge the quality of a feedforward signal, a measure is needed that quantifies the feasibility of performing a manipulation task when there are disturbances or modeling errors. This section explains the novel Manipulation Sensitivity Norm (MSN), which is inspired by the gait sensitivity norm used for bipedal walking robots [14]. This inspiration comes from the insight that a pick and place task can be seen as a repetitive motion and can therefore be analyzed using limit cycles [10]. The effect of disturbances on limit cycles on bipedal robots can be captured by the gait sensitivity norm, which analyses the system based on an estimation of an input-output relation that is defined once per step. This means that the effect of a realistic disturbance profile during one step is taken as input, and a performance measure as it occurs during that step is the output. In walking robots, a possible output is the step time. A slight modification of the gait sensitivity norm can be used to analyze the performance of manipulation tasks. This modification is the MSN, and requires four steps to compute.

1. Defining output indicators
2. Defining a set of realistic disturbances as input signals
3. Obtaining the input to output relation
4. Computing the appropriate system norm of the input output relation.

The first step is to define output indicators. For pick and place tasks, output indicators are a measure of the distance of the arm to the desired path. To make the analysis as clear as possible, we use the error in the absolute angles of the links at the pick position, which is the initial position of the cycle. The MSN will compute the gain from a set of realistic disturbances to this output measure and is therefore a measure of accuracy when moving under real world disturbances.

The next step is to define the disturbances, which are used as inputs. For our analysis, we use three disturbances: a torque on the first link during the first 0.15 s of the cycle,

a varying end-effector mass that represents a product that has a different weight than expected and a varying viscous friction coefficient. These inputs have a nominal value of 0, and their value is allowed to change every cycle. Note that the last two inputs, mass and friction, are typically seen as parameter variations. For this linearized analysis, there is no mathematical distinction between such a parameter variation and a more traditional disturbance such as the torque. This justifies treating parameter variations and disturbances in the same way. Do note however, that the parameter uncertainty is lasting, which should be reflected when computing the input-output gain.

In the third step, the input-output relation in Eqs. 8-9 are obtained using a finite difference scheme. At the beginning of every cycle, very small (10^{-5}) initial condition disturbances are used to obtain a Jacobian matrix A, by comparing the initial state to the state after exactly one cycle. Then, small values (10^{-5}) of the inputs are used to obtain the input to state Jacobian B. Finally, the relation to the output is linearized, again both for initial error and inputs, obtaining C and D respectively. The procedure is described in more detail in [14]. We now obtain a state space system S of the form:

$$x(n+1) = Ax(n) + Bu(n) \tag{8}$$
$$y(n) = Cx(n) + Du(n) \tag{9}$$

where x is a vector containing the errors in the state after each cycle and y is a vector containing the errors in the positions at the pick position. In this linearized discrete system, the matrix A depends on the system dynamics and the chosen trajectory and B depends on how the inputs influence the state error. Note that for our choice, the output indicators are already linear in the state and $D = 0$.

The last step is to compute an appropriate norm, which is the main difference between the gait sensitivity norm and the MSN. For walking, it is important that the walking motion recovers to normal after the disturbance stops. For manipulation, the disturbances tend to last, meaning we are interested in the maximum error in the situation where the disturbance continues to exists, The appropriate norm is thus the induced $\mathcal{L}_\infty$ norm:

$$\|S\|_{\mathcal{L}_\infty} = \sup_{u \neq 0} \frac{\|y(u)\|_\infty}{\|u\|_\infty} \tag{10}$$

What remains is to compute $\|S\|_{\mathcal{L}_\infty}$. The $\mathcal{L}_\infty$-induced norm is the same as the $\mathcal{L}_1$ norm of the impulse response [15, 16]. Rather than computing the complete $\mathcal{L}_1$ norm, we approximate it by taking the sum of the first N steps, with $N = 100$, chosen sufficiently large. Take $g_{ij}(n)$ as the impulse response from input j to output i. Now the Manipulation Sensitivity Norm can be written as:

$$MSN = \|S\|_{msn} = \max_i \sum_{n=0}^{N} \sum_j |g_{ij}(n)| \tag{11}$$

The MSN is the amount of error given a unit input and therefore could have radians as unit. However, the specific input for which the error is largest differs per cycle. Therefore, we chose to not use a unit for the MSN. The calculation of the MSN is visualized in Fig. 3 and the overall procedure is summarized in Algorithm 1.

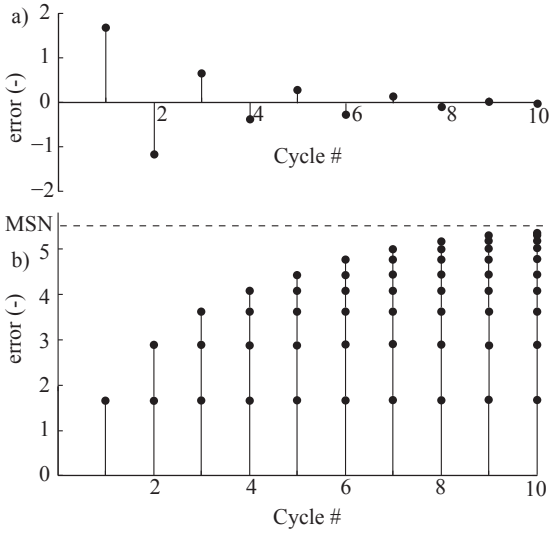

Figure 3. A visualization of the calculation of the MSN. a) The impulse response of the system. b) The sum of the absolute values of the impulse response. The MSN can be interpreted as the amount of error given a unit input, which would have radians as unit. Since the input for which the error is largest differs per cycle, we chose to not use a unit for the MSN.

When computing the MSN, we should scale the size of the inputs in order to take into account the difference in the effect they have and the realistic sizes of those inputs. Since the expected disturbances depend heavily on the system under consideration, we choose a different approach, which highlights the capability of the MSN to take into account multiple disturbance sources at the same time. The input sizes are scaled such that the MSN of each of the three inputs considered separately is 1, when feedback gains specified by $\omega = 1$ s^{-1} are used for the MSN minimization.

Algorithm 1 Calculating the MSN

procedure MSN($\tau(t), \omega, N$)
  Determine $[q(t), \dot{q}(t)]$ ▷ Eq. (1)
  Determine K ▷ Eq. (4)
  Determine C ▷ Eq. (5)
  Determine A ▷ Finite difference on q_0
  Determine B ▷ Finite difference on u
  for $j = 1..J$ **do**
    for $n = 1..N$ **do**
      Determine $g_{ij}(n)$ ▷ Eqs. 8-9
    end for
  end for
  Determine MSN ▷ Eq. 11
  return MSN
end procedure

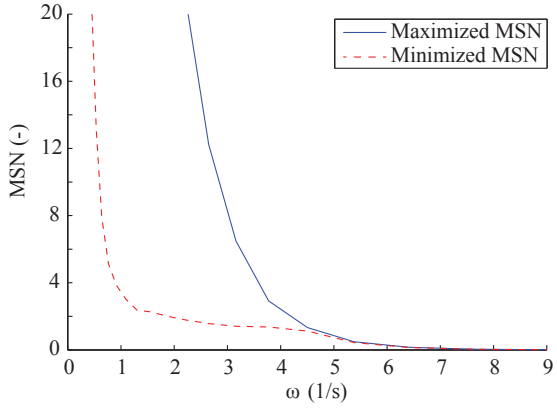

Figure 4. The minimized and maximized MSN as function ω, as found in simulation.

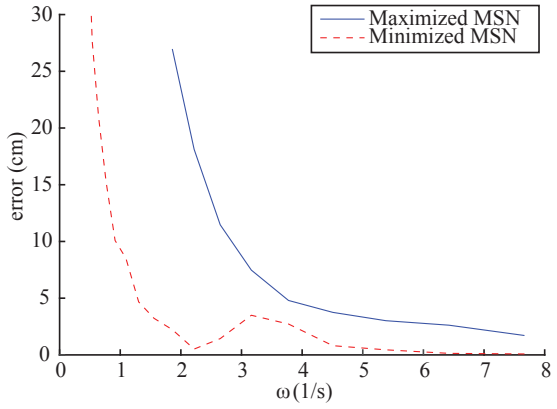

Figure 5. The errors at the pick position as function of ω, as found in hardware experiments. The errors are shown for trajectories with a minimized MSN and a maximized MSN.

The overall optimization for minimizing/maximizing the MSN is described by eq. (7), with as cost function

$$C(\tau_{ff} = \pm MSN(\tau_{ff}) \tag{12}$$

We used `fmincon` with 20 initial conditions determined by `multistart` in MATLAB® to solve the optimizations.

B. Simulation results

Fig. 4 shows the minimum and maximum MSN that were obtained as functions of the natural frequency parameter ω. This figure shows what was to be expected: increasing the gains results in a decrease of the MSN. The figure shows both the maximum and the minimum MSN that were obtained by optimization. At low gains, the maximization does not result in stable controllers, meaning that the MSN is infinitely large. This instability shows that at these gains the unstabilizing dynamical effects are larger than the stabilizing effects of the feedback controller. For $\omega > 5$ s^{-1}, the difference between the cycles with minimized and maximized MSN is negligible.

The red lines in Fig. 6a and 6b correspond to two optimized cycles. The cycle in Fig. 6a was obtained by maximizing the MSN and the cycle in Fig. 6b was obtained by minimizing the MSN. These cycles were obtained for ω = 2.7 s^{-1}. The corresponding values for the MSN are 12.2 and 1.6 respectively. Finally, Fig. 7 shows the time evolution of the torque signals, the feedforward term of which was obtained in simulation.

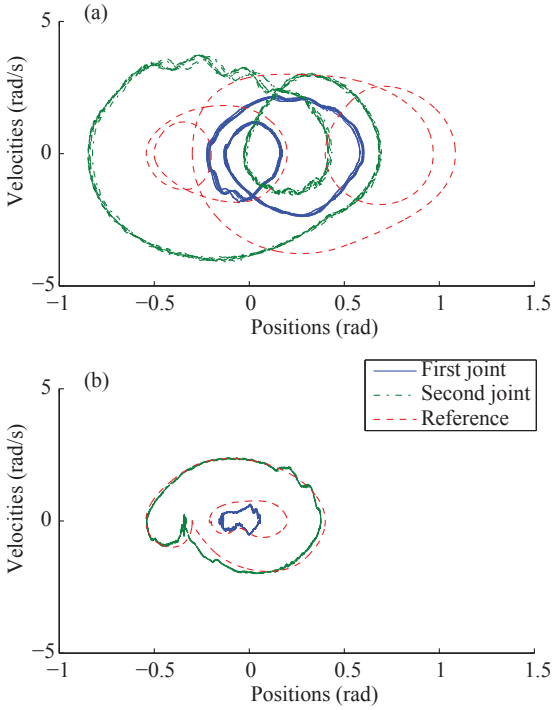

Figure 6. State space plots of the optimization and hardware results for ω =2.7 s^{-1}. a) The cycle with a maximized MSN. b) The cycle with a minimized MSN.

C. Hardware results

Fig. 5 shows the position error of the end effector at the pick position during hardware experiments as function of the natural frequency parameter ω. These errors are the average error over 10 cycles after letting the robotic arm converge for 2 cycles initially. The standard deviation over these 10 cycles is negligible. Logically, the errors decrease when the feedback gains are increased.

At the pick position, the error of the MSN minimizing trajectory is 0.3 cm at ω = 2.1 s^{-1}, in between ω = 2.1 s^{-1} and ω = 4.5 s^{-1}, the error of that trajectory is approximately 2.5 cm and for ω >4.5 s^{-1}, the error drops to approximately 0.3 cm again. The error of the maximized-MSN-trajectory is larger than 2.5 cm, for almost the whole range of gains. Only, for ω = 7.7 s^{-1}, the error becomes 1.8 cm. This significant difference between the errors of these two trajectories means that the choice for the feedforward controller is important for the accuracy of the task execution.

Fig. 6 shows two typical sets of hardware results. These results were obtained for a controller with ω = 2.7 s^{-1}. The plots show the state space trajectories for a maximized MSN and a minimized MSN. They show that the three trajectories differ significantly: the trajectory with maximized MSN covers a larger part of state space than with minimized MSN. The measurements on the prototype show the 10 cycles used in determining the errors used in Fig. 5. It can be seen

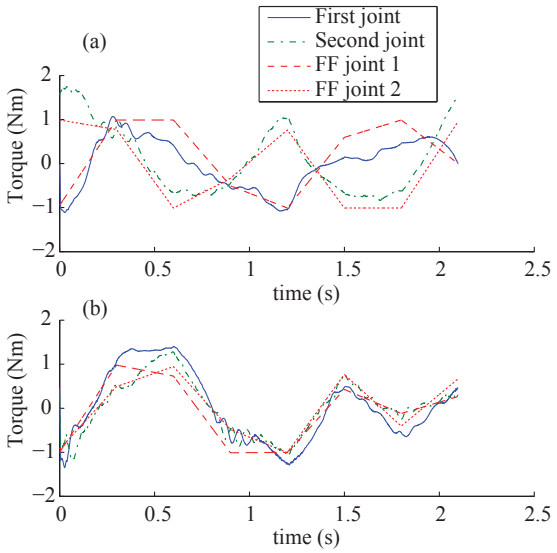

Figure 7. The torques as functions of time for the results with ω =2.7 s^{-1}. a) Torques for the cycle with a maximized MSN. b) Torques for the cycle with a minimized MSN.

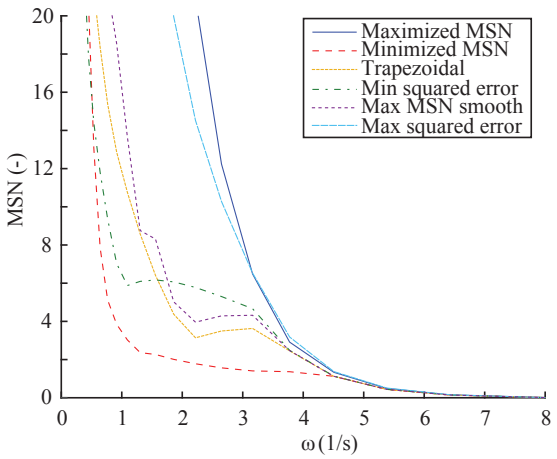

Figure 8. The MSN as found in simulation experiments. The errors are shown for six types of trajectories.

that the arm has converged, and only very little variation between cycles occurs.

Fig. 7 shows the torques for the two different feedforward controllers with $\omega = 2.7$ s^{-1}. The plots show both the feedforward torque and the actual torque. The difference between the two is due to the feedback controller. The plots clearly show that the feedback control effort is larger when following the trajectory with maximized MSN (Fig. 7a), than when following the trajectory with minimized MSN (Fig. 7b).

IV. Alternative motion profiles

In the previous section, the feedforward controllers under study were determined by minimizing or maximizing the MSN. To further study which feedforward controllers lead to accurate motions, four more methods to generate a feedforward controller will now be compared. The simulation and hardware results for these controllers are found in Fig. 8 and 9 respectively.

The first of the controllers is used to compare the minimized and maximized MSN trajectories to a trajectory that is standard in industry: *a trapezoidal velocity profile*. The trapezoidal velocity profile is created by dividing the time to move between pick and place position in three equal parts: one part each for acceleration, constant velocity and deceleration. The same procedure is used to move from place to pick position. Both the MSN in simulation, and the position error on the hardware show that this trapezoidal trajectory has accuracy closer to the minimized MSN trajectory than to the maximized MSN trajectory. In hardware results, the error of the trajectories with minimized MSN and trapezoidal velocity profile are not even significantly different.

So, why does this standard controller perform as accurate as the optimally accurate one? There are two potentially beneficial aspects to this trapezoidal trajectory. First, it is relatively smooth, without large accelerations back and forth. Second, it approaches the goal directly, and is already close to the goal position for the last part of the motion. To test if these two effects are indeed beneficial, we compare feedforward controllers with these two specific aspects.

If smooth controllers lead to accurate motions, it should be impossible to make an inaccurate motion with a smooth controller. To test this, we performed *the MSN-maximization with a smoothness constraint*. Here we took the smoothness as a maximal torque time derivative of $t_f/4$ Nm/s. This rate allows the torque to go from maximum to minimum and back in one cycle. As can be seen in Fig.9, this maximization with constraint has similar accuracy as the minimization in hardware results. This indicates that smoothness is indeed beneficial for accuracy.

To test whether a quick motion towards the goal leads to low errors, we optimized a cost function that squares the error with the goal position. This new optimization is otherwise the same, but minimizes the following cost function:

$$C(\tau_{ff}) = \int_0^{t_f} (q(\tau_{ff}) - q_g(t))^T (q(\tau_{ff}) - q_g(t)) dt \quad (13)$$

With $q_g(t)$ being the way-point position when $t < t_f/2$, and the initial position otherwise. Again, the results show that *this squared error minimization* gives accuracy close to the minimized MSN trajectory, as expected.

In order to confirm these results, we also tested the motion with a *maximize the squared error* function? Because this would lead to a motion that moves away from the target, and only reaches the target at the very last moment, this motion is expected to result in relatively large errors. Furthermore, moving away, and then rapidly towards the target is not very smooth, which is also likely to affect the accuracy adversely. Figs. 8 and 9 show that this prediction is indeed true. The trajectory performs worse than the other trajectories, although still not as poorly as the maximized MSN trajectory.

V. Discussion

In this section, we discuss the results as presented in this paper. As can especially be seen in Fig. 5, the choice

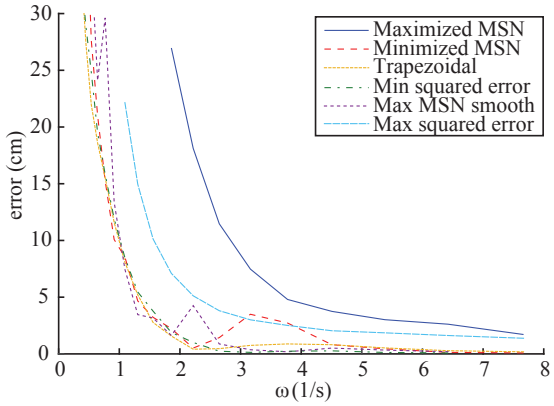

Figure 9. The errors at the pick position as function of ω, as found in hardware experiments. The errors are shown for six types of trajectories.

for a certain trajectory is important for the accuracy that is achieved. For high gains ($\omega > 3.2$ s^{-1}), this choice makes the difference between a negligible position error and an error of multiple centimeters. For medium gains (1.9 s$^{-1} < \omega < 3.2$ s^{-1}), this choice makes the difference between negligible position errors and errors between 5 and 27 cm. And for low gains ($\omega < 1.9$ s^{-1}) this choice makes the difference between stability and instability.

The simulation and hardware results match well. Particularly, the shape of Figs. 4 and 5 are similar. There are however two differences. First, the difference in error between the trajectories does not converge to 0 when the gains increase in hardware experiments, whereas the difference in MSN does convergence in simulation. The second difference is the hump in error and MSN that occurs around $\omega = 3$ rad/s. These differences are small and are caused by unmodeled dynamics that were not taken into account in our choice for disturbances in the MSN-computation. The most likely effects are elasticity in the timing belts and backlash. Because the simulation and hardware results are so similar, the MSN is a good approximation for accuracy, and can be used to find feedforward controllers in cases when feedback gains are low, yet accuracy is important.

The shape of Figs. 8 and 9 are similar, but there are clear differences. The first and most important difference is that only two control strategies lead to errors that are significantly larger than the minimum error. Those two control strategies are the maximized MSN and the maximized squared error. These results suggest that there are two principles that lead to small errors: smoothness and being close to the goal position before the end of the motion. Therefore, we expect that other common profiles such as minimal energy, minimal torque, minimal acceleration and minimal jerk will also perform well. The second difference is that in the hardware results, the accuracy of the MSN-optimized feedforward controllers only give an estimate of the range of possible accuracies. This can be seen in Fig. 9, in which the minimized MSN controller is not the most accurate one for 2.1 s$^{-1} < \omega < 4.5$ s^{-1}.

In simulation, there are two points where one of the comparison trajectories has an MSN that is outside the range given by the MSN minimization and maximization. Specifically, this occurs for the In both these instances, the difference is small, and caused by the choice of step size in simulation. In optimization, a sampling time of 0.01 s is used to save computation time. For Fig. 8, a sampling time of 0.001 s was used, because this aligns with the robot hardware.

As ω goes to zero, both the MSN and the error in hardware results get very large. This is logical since the system without feedback is not stable. Fig. 5 shows that for feedback with $\omega < 1.8$ s^{-1}, the error is larger than 2.5 cm and therefore picking up objects of reasonable size will be difficult. If the feedback gains cannot be increased, more mechanical feedback has to be implemented. The most straightforward approach is to place springs at the joints. In our previous work [10, 11], we showed that with a spring at the first joint, tasks can be performed stably even when $\omega = 0$.

The results from this paper can be improved by incorporating the feedback in a Repetitive Control (RC) scheme [17]. In an RC scheme, the feedforward controller is adjusted based on the state error in the previous cycle. In the most simple form, the feedforward controller in the current cycle is equal to that in the previous cycle plus the feedback that was applied. Such an RC scheme was used before on robotic arms to learn open loop stable trajectories [11].

There is an interesting parallel between the controller we use in this paper and human movement control. Similar to our controller, humans also exploit the advantages of both feedforward and feedback in order to optimize their performance [18]. For fast motions, humans cannot rely on feedback at all, due to the large time delays (typically 150 ms for humans [19, 20]). Therefore, they have to rely on feedforward, in which control signals are generated based on the prediction of an internal model [21]. In slower motions, more feedback is used to correct for inaccuracies in the internal model and external disturbances.

Another interesting parallel with human motion control is the fact that smooth motions perform well. In human motion control, there is an ongoing debate about the cost function humans use optimize their motions. Suggested cost functions are the maximum jerk [22, 23], change of torque [24] and sensitivity to motor noise [25]. Other researchers suggest that humans perform some kind of stochastic optimal control in which variability in task irrelevant directions is ignored [26]. The problem in this debate is that all cost functions result in approximately the same smooth motions. Similarly, we expect that all smooth motions that result from such cost functions will perform well in terms of accuracy.

VI. Conclusion

In this paper we focused on the question 'does the choice of the feedforward controller influence the accuracy of systems with (limited) feedback?'. The answer to this question is: 'yes, the choice for a certain feedforward controller makes the difference between an accurate and an inaccurate task execution.' The feedforward controller was tuned using the novel Manipulation Sensitivity Norm, which measures the

accuracy while taking into account disturbances and model errors. Feedforward controllers that were either minimized or maximized for this norm were implemented on our robotic arm. Results show that for a large range of feedback gains, the error varies between 0.3 and 2.5 cm, depending on the choice for the feedforward controller. Further experiments with alternative feedforward controllers indicated that a trajectory that is either smooth, or approaches the goal position quickly, will be accurate. Therefore, the commonly used trajectory with a trapezoidal velocity profile performs well and is a good choice in terms of accuracy.

ACKNOWLEDGEMENT

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

REFERENCES

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

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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), IEEE/RSJ International Conference on*, 2013, pp. 3382–3389.

[9] M. Plooij, W. Wolfslag, and M. Wisse, "Robust feedforward control of robotic arms with friction model uncertainty," *Robotics and Autonomous Systems*, vol. 70, no. 0, pp. 83 – 91, 2015.

[10] M. Plooij, W. Wolfslag, and M. Wisse, "Open loop stable control in repetitive manipulation tasks," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2014.

[11] W. Wolfslag, M. Plooij, R. Babuska, and M. Wisse, "Learning robustly stable open-loop motions for robotic manipulation," *Robotics and Autonomous Systems*, vol. Available Online, 2015.

[12] 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.

[13] 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.

[14] D. G. E. 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, Dec 2007.

[15] B. Picasso and P. Colaneri, "A factorization approach for the l_∞-gain of discrete-time linear systems," in *Proc. of the 17-th IFAC world congress, Seoul, ROK*, 2008, pp. 1299–1304.

[16] S. P. Boyd, C. H. Barratt, S. P. Boyd, and S. P. Boyd, *Linear controller design: limits of performance*. Prentice Hall Englewood Cliffs, NJ, 1991.

[17] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *International Journal of Control*, vol. 73, no. 10, pp. 930–954, 2000.

[18] 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.

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

[20] 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.

[21] M. Kawato, "Internal models for motor control and trajectory planning," *Current Opinion in Neurobiology*, vol. 9, no. 6, pp. 718 – 727, 1999.

[22] N. Hogan, "An organizing principle for a class of voluntary movements," *The Journal of Neuroscience*, vol. 4, no. 11, pp. 2745–2754, 1984.

[23] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *The journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[24] Y. Uno, M. Kawato, and R. Suzuki, "Formation and control of optimal trajectory in human multijoint arm movement," *Biological cybernetics*, vol. 61, no. 2, pp. 89–101, 1989.

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

[26] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nature neuroscience*, vol. 5, no. 11, pp. 1226–1235, 2002.