

A Direct Sampled-data Design Approach for Robot Stabilization

A. M. Pertew ^{*,1} H. J. Marquez ^{**,1} Q. Zhao ^{***,1}

* *pertew@ece.ualberta.ca*

** *marquez@ece.ualberta.ca*

*** *qingzhao@ece.ualberta.ca*

Abstract: In this paper, a discrete-time output feedback controller that stabilizes the class of nonlinear Lipschitz systems is presented. The applicability of the controller design in the robotics field is thoroughly discussed from both the modeling and design perspectives. It is shown that the objective of stabilizing the robot manipulator continuously using a discrete-time controller can be achieved using a direct sampled-data design approach in which neither the plant model nor the controller need to be discretized a priori, and where the robot measurements are only available at discrete points of time. The proposed design is shown to have important advantages over the classical emulation approach that has been used to solve similar problems. Simulation results on a nonlinear model of the single link flexible joint robot are presented.

Keywords: Sampled-data Control, Robot Manipulator, Lipschitz Systems, H_∞ .

1. INTRODUCTION

Nonlinear control design continues to be a challenge. One of the main challenges, especially in practice, is the Sampled-data (SD) nonlinear control problem, motivated by the use of digital computers in most modern control setups. In this case, a nonlinear continuous-time plant is controlled using a discrete-time controller, where sampled signals from the plant are processed with a digital computer, and where the plant input is obtained by using a hold device on the resulting discrete-time signal generated by the computer. Three standard approaches have been applied to design nonlinear SD controllers in this case [13]: (i) Continuous-time “*Emulation*” design, (ii) Discrete-time “*Discretization*” design, and (iii) Sampled-data “*Direct*” design.

In the *emulation method*, a continuous-time controller is designed based on the continuous-time plant model, then the discrete-time controller is obtained through discretization. The majority of the work done for nonlinear SD control uses this technique, see for example [2, 4, 5]. These results use the principle that the discretized controller recovers the performance of the continuous-time controller as the sampling frequency is sufficiently large, or try to show the relationship between the sampling time and the domain of attraction. In general, these methods assume the existence of a continuous-time stabilizing controller and do not focus on the controller design problem. The focus is rather on the sampling fre-

quency, and fast sampling rates are usually needed to achieve stability in this case. In the *discretization method*, a digital controller is designed based on the exact or approximate discrete-time plant model. For results on nonlinear SD control that use this approach, see [7, 9, 10]. Most of these results pursue controller design for sampled-data nonlinear systems based on the approximate discrete-time model of the system when sampling is sufficiently fast. It is important in the discretization approach to consider approximate discrete-time models simply because exact models are impossible or hard to obtain in most nonlinear systems of practical interest. As with the emulation method, a limitation of these results is that they do not consider the design problem (see for example [13]) since their main objective is to provide sufficient conditions to guarantee the stability of the exact discrete-time plant if the given digital controller stabilizes the approximate discrete-time plant. Another limitation is the fact that stability is only considered in the discrete-time domain, and that the intersample response is not considered in this case. In the *direct SD design approach*, no approximations are made and a discrete-time controller is designed based on the sampled-data model of the plant which must include exact modeling of the sample and hold devices, and therefore constitutes a modeling challenge in nonlinear systems. Due to the complexity of the problem, results on the direct SD design method are scarce (see [3] for direct design of sampled-data linear time-invariant systems). The main advantage of this approach is that it makes no approximations, intrinsic in the discretization process, and that the design takes into account the inter-sample behavior.

¹ Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada, T6G 2V4.

Another important advantage is that the controller design problem plays an important role in the stability analysis, and is not considered as an independent problem unlike the emulation and discretization designs. This alleviates the need for fast sampling rates, and hence puts less restrictions on the control hardware.

In this paper, we focus on the latter approach proposing a new direct design technique for solving the SD stabilization problem of the general class of nonlinear Lipschitz continuous systems. Due to its importance, the class of systems considered in this paper has seen much attention in the literature, especially for the observer and state estimation design problems (see for example [1, 14, 17, 20]). Our approach to solve the SD stabilization problem is as follows: we first show that the SD stabilization problem is equivalent to a standard discrete-time H_∞ problem solvable using commercially available software packages (this can therefore be seen as an extension of the works in [14, 15] where the H_∞ approach was used for solving the observer design and sensor fault estimation problems in the continuous-time domain). We then prove that the discrete-time H_∞ problem is sufficient for exact stabilization, presenting a parameterization for the set of “direct design” SD controllers in this case. The proposed design is shown to have important advantages over the classical emulation approach that has been used to solve similar problems. This is illustrated through simulations on a nonlinear model of the single link flexible joint robot. The general application in the robotics field is thoroughly discussed from both the modeling and design perspectives.

2. PROBLEM DEFINITION

In the sequel we consider the nonlinear system Σ of the form:

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + \Phi(x, u, t) \\ y(t) = Cx(t) \end{cases} \quad (1)$$

satisfying the following conditions

- (1) $\Phi(0, u, t) = 0 \quad \forall u \in \mathbb{R}^m, \forall t \in \mathbb{R}$.
- (2) $\Phi(x, u, t)$ satisfies a uniform Lipschitz condition globally in x , i.e.,
$$\|\Phi(x_1, u, t) - \Phi(x_2, u, t)\| \leq \alpha \|x_1 - x_2\| \quad (2)$$
 $\forall u \in \mathbb{R}^m, t \in \mathbb{R}$ and $\forall x_1$ and $x_2 \in \mathbb{R}^n$. Here $\alpha \in \mathbb{R}^+$ is referred to as the *Lipschitz constant* and is independent of x, u and t .
- (3) (A, B) stabilizable and (A, C) detectable.

We will discuss in section 4 how the Euler-Lagrange model of robot manipulators can also be put in this form. The third condition (stabilizability and detectability of the pairs (A, B) and (A, C) respectively) is needed to guarantee that the standard assumptions in the H_∞ optimization theory are satisfied. In this paper, we consider the output feedback stabilization problem of the system Σ in the following sense:

Definition 1. The system Σ is said to be globally exactly stabilized by the output feedback control input $u = g(y)$, if the equilibrium point of Σ at the origin is rendered globally asymptotically stable by u (i.e, if the system $\dot{x} = Ax + Bg(y) + \Phi(x, g(y))$ has a globally asymptotically stable equilibrium point at the origin, in the sense that $x(t) \rightarrow 0$ as $t \rightarrow \infty, \forall x \in \mathbb{R}^n$).

We focus on the sampled-data (SD) case, in which the output y is sampled at discrete time instants (as shown in Figure 1²) and the input $u(t) = Hu(k)$ where H is the hold operator, and where the sample and hold devices are both operating with the speed $1/h$ (h being the sampling time).

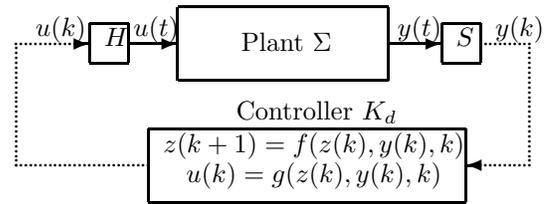


Fig. 1. The SD stabilization problem.

Therefore, in our problem definition we consider that: (i) not all the system states are measurable, (ii) the measurements are only available every “ h ” seconds, and our objective is to design the discrete time controller that achieve exact time stabilization according to Definition 1.

Throughout the paper, we will make use of the following notation: Consider a system $G : \mathcal{L}_2 \rightarrow \mathcal{L}_2$. We will represent by $\gamma(G)$ the \mathcal{L}_2 gain of G defined by $\gamma(G) = \sup_u \frac{\|Gu\|_{\mathcal{L}_2}}{\|u\|_{\mathcal{L}_2}}$. It is well known that, for a linear system $G : \mathcal{L}_2 \rightarrow \mathcal{L}_2$ with a transfer matrix $\hat{G}(s)$, $\gamma(G)$ is equivalent to the H-infinity norm of $\hat{G}(s)$ defined as follows: $\gamma(G) \equiv \|\hat{G}(s)\|_\infty \triangleq \sup_{\omega \in \mathbb{R}} \sigma_{\max}(\hat{G}(j\omega))$ where σ_{\max} represents the maximum singular value of $\hat{G}(j\omega)$. The corresponding \mathcal{L}_2 space and H_∞ norm in the discrete-time case will also be used (see [3] for their exact definition). The matrices $I_n, 0_n$ and 0_{nm} represent the identity matrix of order n , the zero square matrix of order n and the zero n by m matrix respectively. The symbol \hat{T}_{yu} represents the transfer matrix from input u to output y . The partitioned matrix $K = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}$ (when used as an operator from $y(k)$ to $u(k)$, i.e, $u(k) = Ky(k)$) represents the discrete-time state space representation ($z(k+1) = A_K z(k) + B_K y(k)$, $u(k) = C_K z(k) + D_K y(k)$) and when used as a mapping between the continuous-time signals

² In the figures of this paper, solid and dotted lines represent continuous-time and discrete-time signals respectively.

$u(t)$ and $y(t)$, it represents the continuous-time version of this state space representation. The operators S and H will be used to represent the sample and zero-order hold operators with sampling time h .

3. DIRECT SD STABILIZATION

In this section, we consider the SD output feedback stabilization problem represented by Figure 1. Our goal is to solve this problem using a direct design approach where no approximate discretizations of the plant or the controller are used, and where the objective is to design a *discrete-time controller* K_d that achieves global exact stabilization of the *continuous-time plant* Σ according to Definition 1. Our main result is presented in the form of a theorem showing that the “direct design” controller can be obtained by solving a standard discrete-time H_∞ problem (without the need to discretize the plant a priori nor to design a continuous-time controller then discretize it as is the case with the emulation method) as follows:

Theorem 1. There exists a dynamic discrete-time controller $K_d = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}$ that stabilizes the system Σ according to Definition 1, if there exists a solution to the following discrete-time H_∞ problem: “Find K_d such that $\|\hat{T}_{\zeta\omega}(z)\|_\infty < \frac{1}{\alpha}$, for the standard setup in Figure 2(a), where $\hat{G}(z)$ is a discrete-time system that is function of both the Lipschitz constant α and the sampling time h (see [16] for details about the computation of $\hat{G}(z)$).”

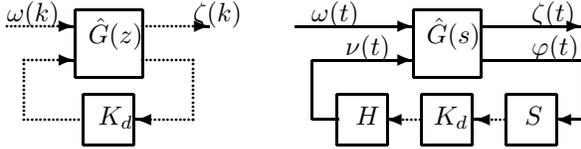


Fig. 2. (a) Discrete-time setup. (b) SD setup.

Proof: The proof is divided into three parts:

Part 1:

Using the first two Lipschitz conditions in section 2, the function $\Phi(x, u, t) : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^n$ satisfies:

$$\|\Phi(x, u, t)\| \leq \alpha \|x\|; \quad \forall x \in \mathbb{R}^n. \quad (3)$$

Then, looking at $\Phi(x, u, t)$ as a nonlinear uncertainty term Δ , and proposing a dynamic discrete-time controller $K_d = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}$, the SD feedback stabilization problem in Figure 1 can be represented by Figure 3 where

$$\|\Delta x\| \leq \alpha \|x\| \quad (4)$$

This can be seen as the feedback interconnection of system P (the black box shown in Figure 3) and the nonlinear uncertainty Δ . By defining the following variables:

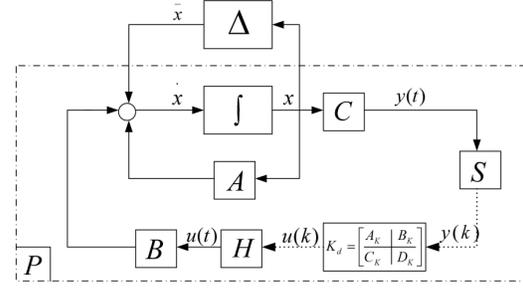


Fig. 3. Feedback interconnection.

$$\begin{aligned} \omega &= \bar{x} \\ \zeta &= x \\ \varphi &= y \\ \nu &= u \end{aligned} \quad (5)$$

the system P can be represented by the sampled-data setup in Figure 2(b), where $\hat{G}(s)$ has the continuous-time state space representation in (6) with the matrices defined in (7)-(8).

$$\hat{G}(s) = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \quad (6)$$

$$\dot{z} = [A] z + [I_n \ B] \begin{bmatrix} \omega \\ \nu \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} \zeta \\ \varphi \end{bmatrix} = \begin{bmatrix} I_n \\ C \end{bmatrix} z + \begin{bmatrix} 0_n & 0_{nm} \\ 0_{pn} & 0_{pm} \end{bmatrix} \begin{bmatrix} \omega \\ \nu \end{bmatrix} \quad (8)$$

Part 2:

If the controller K_d satisfies $\|\hat{T}_{\zeta\omega}(z)\|_\infty < \frac{1}{\alpha}$ for the standard setup in Figure 2(a), then the \mathcal{L}_2 gain of the system P (i.e., the \mathcal{L}_2 gain from $\omega(t)$ to $\zeta(t)$ in Figure 2(b)) is less than $\frac{1}{\alpha}$. More details about the relation between the two setups in Figure 2 can be found in [3]³.

Part 3:

The uncertainty Δ is a static nonlinearity (no internal states) with finite \mathcal{L}_2 gain $\leq \alpha$ due to (4). The mappings $T_{x\bar{x}} : \bar{x} \rightarrow x$ and $\Delta : x \rightarrow \bar{x}$, therefore, satisfy $\gamma(T_{x\bar{x}}) \cdot \gamma(\Delta) < 1$. Using a dissipativity argument for the feedback interconnection of Figure 3, $T_{x\bar{x}}$ and Δ are dissipative with respect to the supply rates $s_1 = -x^T x + \gamma(T_{x\bar{x}})^2 \bar{x}^T \bar{x}$ and $s_2 = -\bar{x}^T \bar{x} + \alpha^2 x^T x$ respectively. Denoting by V_1 and V_2 the storage functions associated with these supply rates, it follows from Corollary 1 in [8] that $V_1 + \epsilon V_2$, $\epsilon > 0$ is a Lyapunov function for the system, and that the equilibrium point is globally asymptotically stable in the sense of Definition 1, and the proof is complete. \triangle

Comment:

The importance of the result in Theorem 1 is that it

³ P is a hybrid system (mix of continuous-time and discrete-time signals) and has an ill-defined transfer function. However, the equivalence between the \mathcal{L}_2 gain of these two setups has been proven. See [3], Chapter 13, for more details.

gives an exact stability result (i.e, exact stability of the equilibrium point is proven and no convergence to stability sets is involved) for the output feedback stabilization problem of SD nonlinear Lipschitz systems. The design is a discrete-time H_∞ problem that can be solved using commercially available software packages. Most importantly, the proposed method is a direct design approach for this SD stabilization problem, where no approximate discretization of the plant Σ or of the controller is used to obtain the solution.

3.1 H_∞ Direct Design

Now, we introduce a design algorithm that is based on the result obtained from theorem 1.

Design procedure:

Step 0. Set the sampling time $h > 0$ and the Lipschitz constant α .

Step 1. Calculate the entries of $\hat{G}(s)$ where $\hat{G}(s)$ is given from (6) with the matrices defined by (7)-(8).

Step 2. Compute the discrete equivalent $\hat{G}(z)$ which is function from $\hat{G}(s)$, h and α (See [16] for more details).

Step 3. Test solvability of the discrete-time H_∞ problem related to the setup in Figure 2(a) and defined as: “Find S_d , the set of stabilizing controllers K_d such that $\|\hat{T}_{\zeta\omega}(z)\|_\infty < \frac{1}{\alpha}$ ”. If test fails then go to Step 4 ; otherwise solve the problem (using available software packages) and any $K_d \in S_d$ is a solution for the SD stabilization problem shown in Figure 1.

Step 4. Decrease α . If $\alpha <$ certain threshold then *stop the algorithm*; otherwise go to Step 1.

4. APPLICATION IN ROBOTICS

4.1 Robot Manipulators as Lipschitz Systems

The dynamic equations of a robot manipulator can be represented by the following structure [18]:

$$u = M(\theta) \ddot{\theta} + V(\theta, \dot{\theta}) \quad (9)$$

where θ , $\dot{\theta}$ and $\ddot{\theta} \in \mathfrak{R}^n$ are vectors representing the position, velocity and acceleration of the n robot joints respectively, $u \in \mathfrak{R}^n$ represent the actuator torques applied at these joints, $M(\theta)$ is referred to as the inertia matrix, and $V(\theta, \dot{\theta})$ is a vector representing the centrifugal, coriolis, gravity and friction terms. In our formulation, sensors may not be available to measure all the joints positions and velocities. Therefore, θ and $\dot{\theta}$ are not assumed measurable for control purposes. By defining the state variables as $x_1 = \theta$, $x_2 = \dot{\theta}$ (note that here x_1 , x_2 represent n -dimensional vectors) we get the state space model:

$$\dot{x} = f(x) + g(x)u \quad (10)$$

where $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $f(x) = \begin{bmatrix} x_2 \\ -M^{-1}(x_1)V(x_1, x_2) \end{bmatrix}$ and $g(x) = \begin{bmatrix} 0_n \\ M^{-1}(x_1) \end{bmatrix}$. This can also be represented as:

$$\dot{x} = Ax + \Phi(x, u) \quad (11)$$

where:

$$A = \begin{bmatrix} 0_n & I_n \\ 0_n & 0_n \end{bmatrix}, \Phi = \begin{bmatrix} 0_n \\ M^{-1}(x_1)u - M^{-1}(x_1)V(x_1, x_2) \end{bmatrix}.$$

It is important to note that the nonlinear terms in Φ are mainly trigonometric terms which are locally Lipschitz, and an upper bound on the Lipschitz constant can be found by computing $\|\Phi(x, u)\|$ over the operating range. Another representation of (10) around an operating point x^* of interest is:

$$\dot{x} = Ax + Bu + \Phi(x, u) \quad (12)$$

where $A = \left(\frac{\partial f}{\partial x}\right)|_{x^*}$, $B = \left(\frac{\partial g}{\partial x}\right)|_{x^*}$, and where $\Phi = (f(x) - Ax + g(x)u - Bu)$.

It is also important to note that (11) and (12) are both exact models of (10). By neglecting the terms in Φ in (12), one gets the well known approximate linearized model around the operating point x^* , i.e:

$$\dot{x} = Ax + Bu \quad (13)$$

where $A = \left(\frac{\partial f}{\partial x}\right)|_{x^*}$, $B = \left(\frac{\partial g}{\partial x}\right)|_{x^*}$, which is an approximate model of (10). In our formulation, however, no approximation is needed and one can directly use the exact Lipschitz model in (12) for the stabilization purpose. Note that (12) is a special case of the general form in (1) and that the system output can be assumed as any “sampled” linear combination of the states as long as the pair (A, C) is detectable. It is also important to note that, as long as the operating point x^* is not a singular point of the manipulator inertia matrix, the pair (A, B) in (12) will always be stabilizable.

4.2 Simulations on a Flexible Joint Robot

In this section, simulation experiments are used to illustrate the performance of the stabilizing controller proposed in section 3. A 4th order model of a single-link flexible joint robot is considered in these experiments [17]. The robot is represented schematically in Figure 4, and its parameters can be found in [17]. The system variables are θ_m (rad), ω_m (rad/sec), θ_l (rad) and ω_l (rad/sec), which represent the motor angular rotation, the motor angular velocity, the link angular rotation and the link angular velocity respectively. The control input is the DC voltage u (V).

The state space representation of the system is as follows:

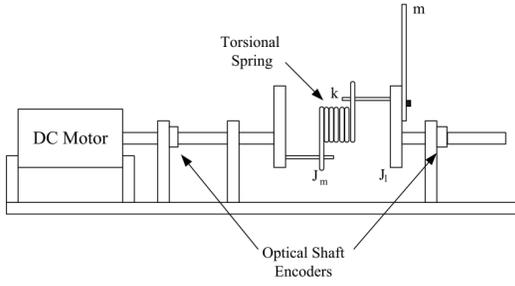


Fig. 4. Schematic of the single link flexible joint robot.

$$\dot{\theta}_m = \omega_m \quad (14)$$

$$\dot{\omega}_m = \frac{k}{J_m} (\theta_l - \theta_m) - \frac{B}{J_m} \omega_m + \frac{K_T}{J_m} u \quad (15)$$

$$\dot{\theta}_l = \omega_l \quad (16)$$

$$\dot{\omega}_l = -\frac{k}{J_l} (\theta_l - \theta_m) - \frac{mgl}{J_l} \sin(\theta_l) \quad (17)$$

By assuming that only the motor position and velocity are measurable, and using the state as $x = [x_1 \ x_2 \ x_3 \ x_4]^T$ where x_1, x_2, x_3 and x_4 represent $\theta_m, \omega_m, \theta_l, \omega_l$ respectively, the model can be represented by:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + \Phi(x, u) \\ y(t) &= Cx(t) \end{aligned}$$

with:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -48.6 & 48.6 & -1.25 & 0 \\ 19.5 & -19.5 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 21.6 \\ 0 \end{bmatrix},$$

$$\Phi(x, u) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -3.33 \sin(x_2) \end{bmatrix}, \quad \text{and } C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The model is in the form of (12), and the Lipschitz constant $\alpha = 3.33$. The stabilization problem considered here is trying to bring x to rest starting from $x(0) = [0 \ 0 \ 3 \ 0]^T$. The simulation time is taken as 10 seconds. The result of continuous-time stabilization is shown in Figure 5 where the controller K is obtained by using a continuous-time H_∞ technique⁴ and is given in (18). The four states of the system converge to zero (only the first two are shown in Figure 5) with $\|x\|_{\mathcal{L}_2} = 13.9361$.

⁴ The continuous-time controller is to be discretized later on. Its matrices are denoted by the subscript "c". It is included here for the sake of comparison between the different SD designs.

$$A_{K_c} = \begin{bmatrix} -1.718 & 0.1178 & 0.1433 & 0.003 \\ -315.8339 & -44.2023 & -6.0686 & -743.6544 \\ -6.8772 & -9.1322 & 0.7208 & 40.0273 \\ 1.8055 & -0.5313 & -0.4602 & 0.0044 \end{bmatrix},$$

$$B_{K_c} = \begin{bmatrix} 0.37 & 0.19 \\ 0.19 & 2.27 \\ 1.48 & 1.96 \\ 0.03 & 0.11 \end{bmatrix}, \quad C_{K_c} = \begin{bmatrix} -58.62 \\ -7.13 \\ -4.05 \\ -163.67 \end{bmatrix}^T, \quad D_{K_c} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}^T \quad (18)$$

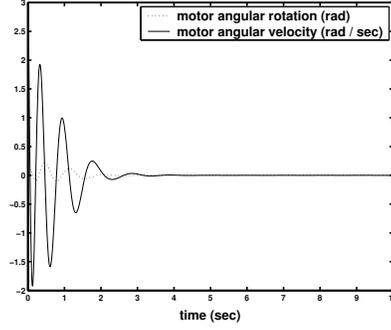


Fig. 5. H_∞ Continuous stabilization.

To simulate the SD stabilization, two design techniques are considered for the discrete-time controller K_d : (i) The first design is by using the emulation method where K_d is obtained by discretization of the continuous controller in (18) (two discretization methods are used here: the *zoh* and the *bilinear* "Tustin" discretizations), (ii) The second design is the direct H_∞ design technique proposed in section 3. Table 1 below shows the trend of the \mathcal{L}_2 norm of x for these two techniques and for different selections of the sampling time h .

Table 1. \mathcal{L}_2 norm of x for different SD stabilization methods

	$h = 0.001 \text{ sec}$	$h = 0.01 \text{ sec}$	$h = 0.1 \text{ sec}$
<i>Zoh</i>	14.0371	15.5256	∞
<i>Bilinear</i>	13.9795	14.4688	∞
<i>Direct H_∞</i>	17.9455	18.4124	20.8695

As seen from Table 1, for large sampling time ($h = 0.1$), the direct H_∞ design method is the only one that converges and the response for this case is shown in Figure 6 where the discrete-time controller obtained using the design algorithm suggested in section 3 is given in (19).

$$A_K = \begin{bmatrix} -0.1342 & 0.0351 & 0.0358 & -0.3946 \\ -2.8096 & -0.7561 & 0.6552 & -6.6464 \\ -4.0948 & -1.5639 & 0.9109 & 3.8428 \\ -0.2761 & -0.0867 & -0.0442 & 0.8769 \end{bmatrix},$$

$$B_K = \begin{bmatrix} 0.82 & 0.07 \\ -3.8 & 0.5 \\ 4.6 & 1.6 \\ 0.46 & 0.1 \end{bmatrix}, \quad C_K = \begin{bmatrix} 0.4 \\ 0.3 \\ -0.2 \\ -4.5 \end{bmatrix}^T, \quad D_K = \begin{bmatrix} -1.7 \\ -0.9 \end{bmatrix}^T \quad (19)$$

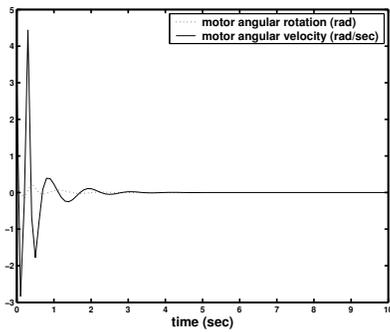


Fig. 6. H_∞ SD stabilization for $h = 0.1$ sec.

The two discretization methods fail to converge when $h = 0.1$ sec (as shown by the ∞ 's in Table 1). By trying different sampling times for this simulation example, the maximum sampling time h stabilizing the manipulator using the “*zoh*” method is $\simeq 0.02$ sec, using “*bilinear discretization*” is $\simeq 0.05$ sec and using the “ *H_∞ direct design*” approach is $\simeq 0.2$ sec (10 times the *zoh* and 4 times the *bilinear*). Finally, Figure 7 compares the response of the motor angular velocity in case of the H_∞ direct design technique with that of the bilinear discretization when the sampling time h is fixed at 0.05 sec. This figure also shows the improvement in the transient response when using the direct design approach.

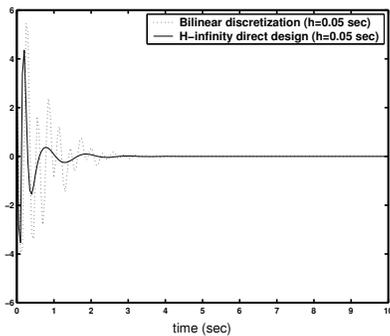


Fig. 7. (H_∞ direct design) vs (emulation design).

5. FUTURE WORK

Future works includes dealing with modeling uncertainties, and calculation of the controller in real-time for more complex systems.

REFERENCES

- [1] C. Aboky, G. Sallet, J. C. Vivalda, “Observers for Lipschitz Non-linear Systems,” *Int. J. Contr.*, Vol. 75, pp. 204-212, 2002.
- [2] B. Castillo, S. Di. Genmaro, S. Monaco, D. Normand-Cyrot, “On Regulation Under Sampling,” *IEEE Trans. on Automat. Contr.*, Vol. 42, pp. 864-868, 1997.
- [3] T. Chen, B. A. Francis, *Optimal Sampled-data Control Systems*, Springer, London, 1995.
- [4] F. H. Clarke, Y. S. Ledyaev, E. D. Sontag, A. I. subbotin, “Asymptotic Controllability Implies Feedback Stabilization,” *IEEE Trans. on Automat. Contr.*, Vol. 42, pp. 1394-1407, 1997.
- [5] A. Dabroom, H. Khalil, “Output Feedback Sampled-data Control of Nonlinear Systems Using High-gain Observers,” *IEEE Trans. on Automat. Contr.*, Vol. 46, pp. 1712-1725, Nov 2001.
- [6] M. Green, D. Limebeer, *Linear Robust Control*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [7] E. Gyurkovics, A. Elaiw, “Stabilization of Sampled-data Nonlinear Syst. by Receding Horizon Control via Discrete-time Approx.,” *Automatica*, Vol. 40, pp. 2017-2028, 2004.
- [8] D. J. Hill and P. J. Moylan, “Stability Results for Nonlinear Feedback Systems,” *Automatica*, Vol. 13, pp. 377-382, July 1977.
- [9] N. Kazantzi, C. Kravaris, “System Theoretic Properties of Sampled-data Representations of Nonlinear Systems Obtained via Taylor-lie Series,” *Int. J. Contr.*, Vol. 67, pp. 997-1020, 1997.
- [10] T. C. Lee, Z. P. Jiang, “On Uniform Global Asymptotic Stability of Nonlinear Discrete-Time Systems With Applications,” *IEEE Trans. on Automat. Contr.*, Vol. 51, No. 10, pp. 1644-1660, 2006.
- [11] R. Marino, P. Tomei, *Nonlinear Control Design: Geometric, Adaptive and Robust*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [12] M. Morari, E. Zaffari, *Robust Process Control*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [13] D. Nesić, A. Teel, P. Kokotović, “Sufficient Conditions for Stabilization of Sampled-data Nonlinear Systems via Discrete-time Approximations,” *Syst. Contr. Letters*, Vol. 38, pp. 259-270, 1999.
- [14] A. M. Pertew, H. J. Marquez, Q. Zhao, “ H_∞ Observer Design for Lipschitz Nonlinear Systems,” *IEEE Trans. on Automat. Contr.*, Vol. 51, No. 7, pp. 1211-1216, July 2006.
- [15] A. M. Pertew, H. J. Marquez, Q. Zhao, “LMI-based Sensor Fault Diagnosis for Nonlinear Lipschitz Systems,” *Automatica*, Vol. 43, No. 8, pp. 1464-1469, August 2007.
- [16] A. M. Pertew, H. J. Marquez, Q. Zhao, “Sampled-data Stabilization of a Class of Nonlinear Systems with Application in Robotics,” accepted in *ASME J. of Dynamic Systems, Measurements and Control*, 2008.
- [17] S. Raghavan, J. K. Hedrick, “Observer Design for a Class of Nonlinear Systems,” *Int. J. Control*, Vol. 59, No. 2, pp. 515-528, 1994.
- [18] L. Sciavicco, B. Siciliano, *Modeling and Control of Robot Manipulators*, McGraw Hill, 1989.
- [19] A. van der Schaft, *\mathcal{L}_2 -Gain and Passivity Techniques in Nonlinear Control*, Springer Verlag, London, UK, 1999.
- [20] S. H. Zak, “On the Stabilization and Observation of Nonlinear/Uncertain Dynamic Systems,” *IEEE Trans. on Automat. Contr.*, Vol. 35, No. 5, pp. 604-607, May 1990.