vibration

MDPI

Article

Chaos Stabilization and Tracking Recovery of a Faulty Humanoid Robot Arm in a Cooperative Scenario

Said G. Khan [1,2,*], Samir Bendoukha [1] and Salem Abdelmalek [3]

[1] Department of Mechanical Engineering, College of Engineering Yanbu, Taibah University KSA, Medina 42353, Saudi Arabia; sbendoukha@taibahu.edu.sa

[2] Department of Mechanical Engineering, University of Bristol, Bristol BS8 1TH, UK

[3] Department of Mathematics and Informatics, Faculty of Exact Sciences and Natural Sciences and Life, University of Larbi Tebessi, Tébessa 12000, Algeria; sallllm@gmail.com

* Correspondence: sfatehrahman@taibahu.edu.sa

Received: 31 December 2018; Accepted: 2 February 2019; Published: 6 February 2019

Abstract: Synchronised motion is an important requirement for two cooperating humanoid robot arms. In this work a cooperative scenario is considered where two humanoid robot arms (using 4DOF each, namely Shoulder Flexion Joint, Shoulder abduction Joint, Humeral rotation joint and Elbow Flexion Joint) motion are synchronized. The master robot arm is controlled by a sliding mode controller and the slave robot arm is synchronized using a basic PD plus adaptive control, employing the position and velocity errors between the master and the slave. During the operation, if a joint of the slave robot arm saturates or malfunctions (for instance, Elbow flexion joint does not respond or free swinging), consequently, slave robot arm will go into chaos (i.e., chaotic motion of the end effector). In this case, a chaos controller kicks in to recover and re-synchronize the motion of the slave robot arm end effector. This re-synchronization is extremely important to complete the task in hand to address any safety issues arising from any joint malfunction of the slave robot. Effectiveness of the scheme is tested in simulation using Bristol Robotics Laboratory Humanoid BERT II arms.

Keywords: humanoid robots; chaos control; actuator fault

1. Introduction

In recent years, robots aimed for social environments have become of great interest to the research and industrial communities alike. In such scenarios, the safety of the robot, its environment and humans or animals therein is a major concern [1]. For example, if the robot in question exerts an excessive force or collision (impact) in a cooperative task with a human being, the result may be catastrophic and the robot may pose a significant risk to the interacting agent whether it is a human or another robot. Force control and its variants such as compliance or impedance control can help to address some of the safety issues by limiting the interaction forces [2–6]. Another safety issue is the development of faults during operation, which cannot be ruled out in robots and other actuated machinery. Faulty actuators may make the system unstable or drastically degrade the tracking performance of the robot. Hence, actuator fault-tolerance is normally considered a very important aspect of the cooperative control of two robot arms. The ideal desired response when a failure is observed is to take immediate action that will ensure the robot arms continue operation as normal as possible during the cooperative task.

Dealing effectively with actuator failures is a challenging task. Over the last couple of decades, numerous studies have explored the subject. The literature related to actuator failures takes several distinct

directions. Some studies have focused on complete actuator failures whereby the actuator movement is decoupled from the input signal while others have dealt with those failures that degrade the tracking performance of the actuators due to partial failures [7]. The most common of these failures are locked joint faults, free swinging or actuator saturation and ramp actuator faults [8,9]. The design of fault-tolerant control schemes remains a hot topic. Recently, an intelligent way to deal with actuator faults was presented in [10]. Also, a fault tolerant control scheme was proposed in [11] for a differential drive mobile robot using multiple models. Another interesting control scheme is that of [12], which features a neural network based adaptive actuator fault detection algorithm for robot manipulators.

Robot–Robot cooperation or interaction is inevitable as many daily life tasks cannot be performed solely by a single robot. Cooperative schemes are mostly bio-inspired as humans and other beings are constantly cooperating to accomplish tasks that would otherwise be impossible. Cooperation is not limited to the mirrored action of two arms belonging to a single entity but in many cases also extends to multiple robots working together. The control of a cooperative robotic manipulator is currently a hot research area with interest from specialists in both the academic and industrial sectors. Many different approaches have been studied and applied as reported in the literature. For instance, [13] proposed a neural network based distributed control scheme for multiple redundant manipulators. In addition, communication networks have been employed to synchronize the positions of all manipulators, see also [14–16].

As it will be shown in this paper, once a joint becomes faulty, the robot end effector moves in a chaotic manner [17]. In general, the term chaos refers to a dynamical system following a trajectory that is neither asymptotically stable nor periodically stable but remains asymptotically bounded to a specific region of phase space. This type of dynamical systems is well known for its extreme dependence on very small variations in the initial conditions. In fact, two trajectories that start from infinitesimally close points in phase space diverge away from one another at an exponential rate, which may be quantified by means of the positive Lyapunov exponents of the system [18]. The literature related to chaos can be split into two main categories, chaos synthesis and chaos control. The first category deals with the implementation and application of chaotic systems that posses special properties in order to make use of the high level of entropy they entail [19]. For instance, in the field data and image encryption, the seemingly unpredictable nature of a chaotic system's states drives the generation of a sequence of secret encryption keys [20,21]. Chaos control, on the other hand, deals with the stabilization and synchronization of chaotic phenomena appearing in nature or in engineered systems. This paper is concerned with the latter.

In this paper, we aim to use a well established energy reduction stabilization controller first proposed in [22] to overcome the chaotic behavior observed as a result of a joint failure. The main contribution of the paper is to use computer simulations of a redundant four-joint robot arm in order to show the effectiveness of the chaos based controller [22] when an actuator failure is exhibited in the middle of a synchronized two robot arm cooperative scenario. The robot arm cooperation scenario under consideration in this work is similar to that of [1]. To the best of our knowledge, chaos based control has not been considered before to deal with actuator faults. More specifically, all of the fault identification and accommodation methods we have seen in the literature are rather complicated and do not take into consideration the chaotic nature of the system under fault. The study at hand shows how a very simple energy reduction chaos controller not only suppresses chaos in the motion of the faulty robot arm but also forces it to correctly track the desired trajectory. This paper is organized as follows. Section 2 describes the cooperative scenario where a master/slave pair is synchronized to carry out a joint task. Section 3 shows the sliding mode controller (SMC) used to drive the master arm to follow a given trajectory. The slave arm is controlled by means of a simple adaptive synchronization law as discussed in Section 4. In Section 5, we discuss the effect of a complete joint failure on the movement of the slave end-effector. In order to overcome the haphazard movement due to the failure, a chaos controller is presented in Section 6. Finally, the simulation results are discussed in Section 7 and conclusions are drawn based on the discussion in Section 8.

2. Robots Cooperation Scenario

In this paper, we consider a cooperation scenario where a master and a slave robot have synchronised motion to complete a task similar to Figure 1. Further, in this cooperative scenario where two robot arms are expected to carry out a certain task by following the exact same trajectory. For instance, the two arms may be carrying a large fragile object. In order to ensure the two end effectors are completely synchronized, we will assume a master/slave configuration where the master end effector's position X_m is controlled to follow the desired path X_{dm} and the slave end effector's position X_s is controlled through an adaptive synchronization control law. As discussed earlier in the paper, the literature related to path planning and synchronization is quite rich and well established. However, one must keep in mind that most actuated joints are prone to different types of failure, which are generally missed by the control methods leading to an erratic and chaotic behavior.

Figure 1. Two Robot arms, a cooperative task.

3. Sliding Mode Control of the Master Arm

For the control of Masters arm simple sliding mode control is employed. In this section, the controller of the master arm is described which is aimed at to follow a desired trajectory determined by the Cartesian coordinates $X_{dm} = [x_{dm}, z_{dm}]^T$, where x_{dm} is the Cartesian demand in the X direction and z_{dm} is the Cartesian demand in the Z direction. An simple and robust scheme that can be found in the literature is the sliding mode controller (SMC), which has the advantage of counteracting system uncertainties [23–27]. SMC is particularly important for mechanical systems with stiction, friction, and small negligible flexibilities. It should also be noted that SMC employed in is dynamic model free and it allows for system uncertainties. This makes it superior to other model based approaches such as feedback linearization and dynamic inversion methods.

In this paper, we employ four-joint robot arm structures whose dynamics can be modeled as

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = T \tag{1}$$

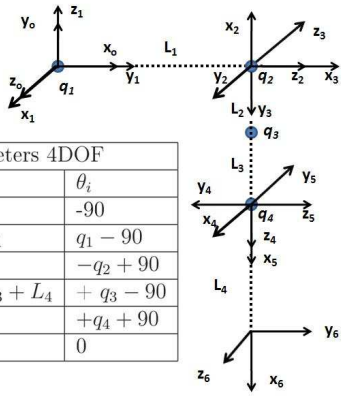
where $M \in \mathbb{R}^{4\times4}$ is the positive-definite and symmetric inertia matrix as a function of the four joint angles q, $V \in \mathbb{R}^{4\times1}$ is the coriolis/centripetal vector, which also represents viscous and nonlinear damping, and $G \in \mathbb{R}^{4\times1}$ denotes the gravity vector. Note that any practical robot is subject to friction and damping which makes it an open loop stable system. Vector $T \in \mathbb{R}^{4\times1}$ in Equation (1) represents the input torque.

A complete list of the physical parameters of the robot is provided in Table 1. These parameters are identical for both the master and slave robot arms. The structure of the BERT2 humanoid robot is depicted in Figure 2. This robot arm has seven degrees of freedom. However, for simplicity, only four are considered in this study. Figure 3 shows the DH parameters for the 4 DOFs in consideration.

Table 1. Links lengths and masses the humanoid robot(BRL BERT2) arm used.

Joint	Mass (Kg)	Length (m)	Radius (m)
1	$m_1 = 4$	$l_1 = 0.21$	$r_1 = 0.03$
2	$m_2 = 1.5$	$l_2 = 0.23$	$r_2 = 0.02$
3	$m_3 = 0.85$	$l_3 = 0.23$	$r_3 = 0.02$
4	$m_4 = 1.8$	$l_4 = 0.23$	$r_4 = 0.02$

Figure 2. Bristol Robotics Laboratory BERT II robot.



D-H Parameters 4DOF				
i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	-90	0	-90
2	0	-90	L_1	$q_1 - 90$
3	0	$+90$	0	$-q_2 + 90$
4	0	-90	$L_3 + L_4$	$+ q_3 - 90$
5	0	$+90$	0	$+q_4 + 90$
6	L_4	0	0	0

Figure 3. The kinematics DH for the 4DOF BERT2 arm.

In case of manipulation of the robot arm, it is usually more convenient to express the model in Cartesian coordinates by considering the forces acting on the end effector at a point $X_m = [x_m, z_m]^T$ in two-dimensional Cartesian space. Where x_m is the global $Cartesian - X$ position and global z_m $Cartesian - Z$ position with the respect to origin which is located in shoulder flexion joint, q_1 (see the kinematics given in Figure 3). As shown in [28], the required transformation of Equation (1) that relies on the Jacobian matrix of the kinematics $X_m = h(q)$ where h is the transformation. It is easy to see that

$$J = \frac{\partial X}{\partial q} \tag{2}$$

By defining a weighted pseudo Jacobian matrix $\overline{J} \in \mathbb{R}^{2\times 2}$ as

$$\overline{J} = M^{-1} J^T \left(J M^{-1} J^T \right)^{-1}, \tag{3}$$

we can express the transformed model by

$$A(q)\ddot{X} + V_{cc}(q,\dot{q}) + f(q) = F, \tag{4}$$

where

$$\begin{cases} A = \left(JM^{-1}J^{T}\right)^{-1}, \\ V_{cc} = \bar{J}^{T}V - A\dot{J}\dot{q}, \\ f = \bar{J}^{T}G, \\ F = \bar{J}^{T}T. \end{cases} \tag{5}$$

$$\mathbf{e}_m = X_d - X_m. \tag{6}$$

It follows that $\mathbf{e}_m(0)$ and $\dot{\mathbf{e}}_m(0)$ are the initial Cartesian position and velocity errors, respectively. According to [29], the integral sliding mode variable can be defined as

$$r = \dot{\mathbf{e}}_m + K_r\mathbf{e}_m \tag{7}$$

where K_r is a full rank 2×2 real matrix with positive entries chosen appropriately.

In order to improve the performance of the controller, we use a combined SMC and proportional derivative (PD) control law described by

$$F = K_p\mathbf{e}_m + K_d\dot{\mathbf{e}}_m + \alpha\frac{r}{\|r+\zeta\|} \tag{8}$$

where K_p and K_d are 2×2 real matrices representing the proportional and derivative gains, respectively, $\alpha > 0$ is a sufficiently large scalar aimed at minimizing the effect of uncertainty and improving the robustness of the SMC controller [29], and ζ is an appropriately selected scalar that minimizes the effect of chattering. The controller described in Equation (8) is well studied in the literature and given a multistep demand $X_d(n)$ results in the response depicted in Figure 4.

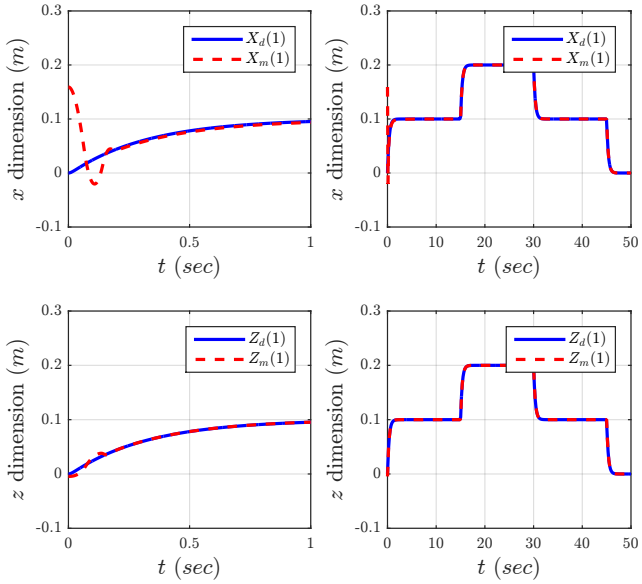


Figure 4. The position X_m of the end-effector in the x_m-z_m plane when the demand X_d is a multistep function.

4. Adaptive Synchronization Control of the Slave

In order to synchronize the motion of the slave arm to follow the same trajectory set out by the master, we use a synchronization adaptive PD controller. We define the instantaneous position synchronization error in discrete time as

$$\mathbf{e}_s = X_m - X_s \tag{9}$$

Differentiating the position error Equation (9) yields the velocity error

$$\dot{\mathbf{e}}_s = \dot{X}_m - \dot{X}_s \tag{10}$$

We define the synchronization control variable as

$$r_s = K_{ps}\mathbf{e}_s + K_{ds}\dot{\mathbf{e}}_s \tag{11}$$

where K_{ps} and K_{ds} are 2×2 real matrices chosen appropriately. Using this adaptive variable, we define the adaptive law

$$\dot{U}_s\,(n) = -\gamma_1 U_s\,(n-1) + \gamma_2 r_s \tag{12}$$

for some experimentally selected real constants $\gamma_1, \gamma_2 > 0$. The control signal U_s is obtained simply by integrating $\dot{U}_s$. Where n represents time step. The adaptive torque term is then, obtained as

$$T_s = J^T U_s \tag{13}$$

Since we are operating in the Cartesian space, the corresponding force demand can be calculated as

$$F_s = \overline{J}^T T_s \tag{14}$$

Given the same multi-step demand adopted earlier in Figure 4, Figure 5 depicts the master and slave positions over a window of 50 s. It is easy to see that the two arms are in fact fully synchronized. Figure 6 shows the synchronization error, which converges towards a reasonably small value.

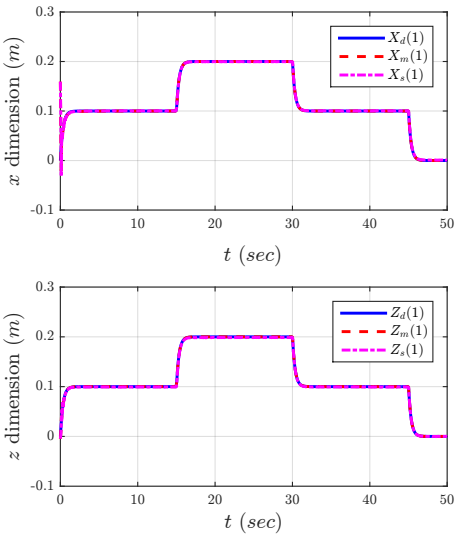


Figure 5. The master and slave positions of the end-effector $X_m\,(t)$ and $X_s\,(t)$, respectively, in the x-z plane when the demand $X_d\,(t)$ is a multistep function.

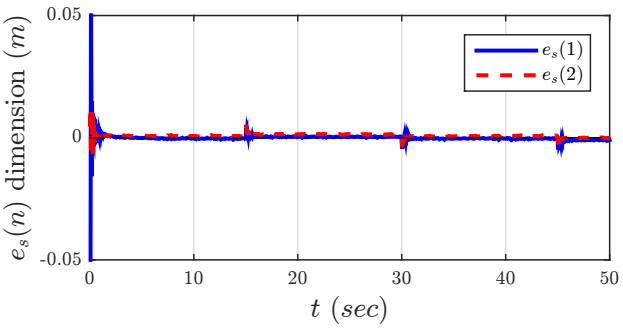


Figure 6. Synchronization error $\mathbf{e}_s$ for the demand depicted in Figure 5.

5. Joint Malfunction and Chaos

Let us consider a fault taking place at one of the slave joints. According to [30], when a fault is observed in one of the actuators, then the control input F_s is disturbed and the modified input is of the form

$$\widetilde{F}_s = \varphi_i(t) F_s + \psi_i(t) \tag{15}$$

where $\varphi_i(t) \in \mathbb{R}^{4\times 4}$ is a diagonal matrix function representing the loss of effectiveness in each actuator and vector $\psi_i(t) \in \mathbb{R}^{4\times 1}$ reflects actuator bias faults. Different values for $\varphi_i(t)$ and $\psi_i(t)$ lead to different types of faults. For instance, setting $\varphi_i = I$ and $\psi_i = 0$ signifies the absence of faults (all healthy actuators). Lowering the diagonal elements of φ_i below 1 corresponds to a loss in the effectiveness of the joints whereas a nonzero vector ψ_i signifies a biased actuator. More details on the types of faults and their identification in robot manipulators can be found in [9]. In what follows we will assume that

$$\varphi_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \psi_i = 0. \tag{16}$$

This is simply synonymous with one of the joints failing completely meaning that either the joint is non-responsive or that a connection problem exists in the configuration.

As described before, a simple SMC scheme is employed for the master arm control and an adaptive synchronization controller for the slave robot arm. Next, we examine the nature of the response and the trajectory followed by the slave arm when a malfunction happens in one of the joints and the control of that joint fails completely, i.e., the control signal no longer reaches the actuator (or the actuator is free swinging). As expected, the arm starts to move in a haphazard and seemingly random way. Figure 7 depicts the controlled x and z coordinates of the end-effector. The Euclidean space demand X_d is simply assumed to be a step with a magnitude of 0.42. The malfunction is introduced after $1s$ of normal operation. It is easy to see that the x-z coordinates start to oscillate in a random fashion. Figure 8 shows the same results in the x-z plane.

Although the phase space plot in Figure 8 shows the boundedness of the position, it does not clearly indicate the dynamic range of the end-effector position and at the same time does not suggest the change in the dynamic range as the input level changes. In order to assess these characteristics, experiment was repeated for step inputs ranging from $0.1m$ to $0.5m$ in steps of $0.01m$. Figure 9 shows the dynamic range of the slave end-effector as a function of the input level. It is easy to see that the dynamic range remains mostly constant throughout the plot.

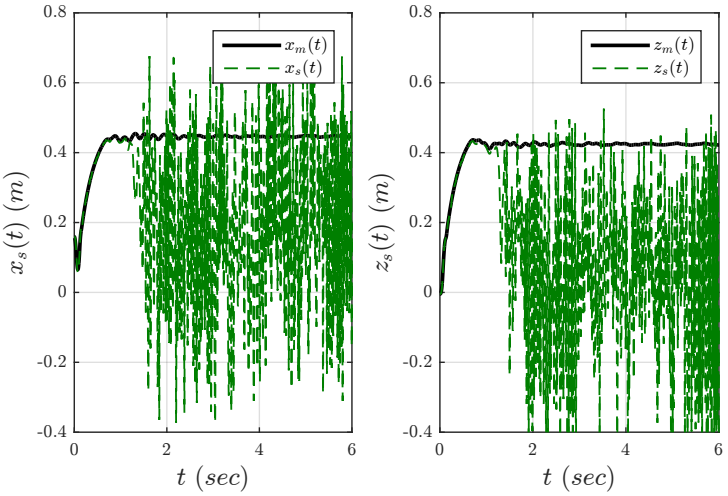


Figure 7. The master and slave positions of the end-effector when the demand $X_d\ (t)$ is a step function and one of the joints completely fails after 1s of operation.

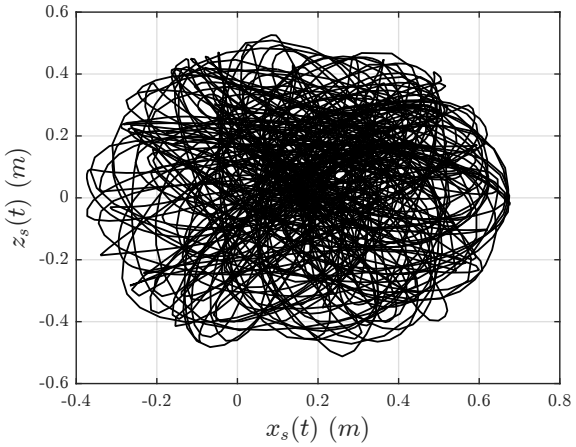


Figure 8. The slave positions after malfunction plotted in the x-z phase plane.

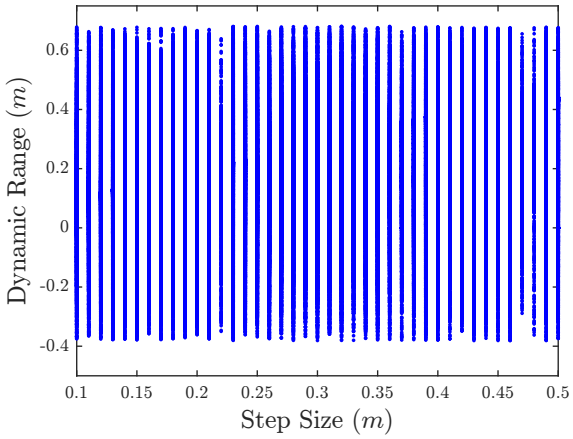


Figure 9. The dynamic range of the end-effector in the x direction after malfunction for a range of step sizes.

6. Chaos Stabilization via Feedback Control

The aim of this section is to introduce a new controller that will operate once a joint fails completely and works to counteract the chaotic tendency of the end-effector. This type of controller is generally referred to as a stabilization scheme. The amount of literature related to the stabilization of chaotic dynamical systems is vast but one set of methods that stands out is based on altering the energy of the system. These methods generally utilize information obtained experimentally from the time series of the system's observable variables to guide the controller. For instance, the Ott-Grebogi-Yorke (OGY) method [31] stabilizes an otherwise unstable periodic orbit (UPO) by means of a small perturbation in the control parameter. However, this can only work if the chaotic trajectory is sufficient close to the desired UPO, which unfortunately is not always the case. The OGY method has been studies extensively in the literature and many variations have proposed including the quasi-continuous OGY [32]. A good comparative study between OGY based methods, multi-parameter methods, and time-delayed feedback methods is given in [33].

Perhaps one of the most interesting perturbation-based control methods for chaotic systems is the tangent scheme proposed in [22]. This controller aims to alter the time averaged compound kinetic and potential oscillatory energy of the system to that of a desired behavior. This type of chaos based control has recently gained popularity in the field of robotics especially for the suppression of unnecessary vibration and the stabilization of an otherwise unstable system. A recent study employs chaos based control to stabilize an electromechanical pendulum [34]. Considering a system of the form

$$\ddot{X} + B\left(X, \dot{X}\right) + C\left(X\right) = F\left(t\right) + g\left(X, \dot{X}\right) \tag{17}$$

where $g\left(X, \dot{X}\right)$ represents the control force, the author showed that in general the control must satisfy the law

$$g\left(X, \dot{X}\right)\dot{X} > 0 \tag{18}$$

for all $\left(X, \dot{X}\right)$. The function g can be reduced to a function of the velocity $\dot{X}$ only and can in general be of the form $g\left(\dot{X}\right) = kh\left(\dot{X}\right)$, where k is a scalar and $h\left(\dot{X}\right)$ is an odd function. One particularly interesting controller is the hyperbolic tangent based one, which is given by

$$g\left(\dot{X}\right) = k\tanh\left(\beta\dot{X}\right) \tag{19}$$

with $0 < \beta \leq \infty$. Although this controller was first proposed in [22] for chaotic oscillators such as the Van der pol and Duffing nonlinear oscillators, it has since been shown that the hyperbolic tangent term can stabilize faulty chaotic systems in general [30]. In this paper, we will show that by adding the feedback term

$$T_f = J^T U_f \tag{20}$$

with

$$U_f = -K_f \tanh\left(\beta\dot{X}_s\right) \tag{21}$$

to control law Equation (13), the chaotic behavior observed when a joint fails can be suppressed.

The overall control scheme is depicted in Figure 10. The complete system has been implemented by means of the Matlab/Simulink environment. The master and slave arms were modeled using the SimMechanics toolbox with a sampling time of 1 ms. As expected, the performance of the stabilization controller Equation (21) depends heavily on the constant parameters K_f and β. The stabilized Cartesian position of the slave end-effector is depicted in Figures 11–14 for different values of K_f and β. Note that in all experiments, the fault is introduced after the first 3 s of normal operation.

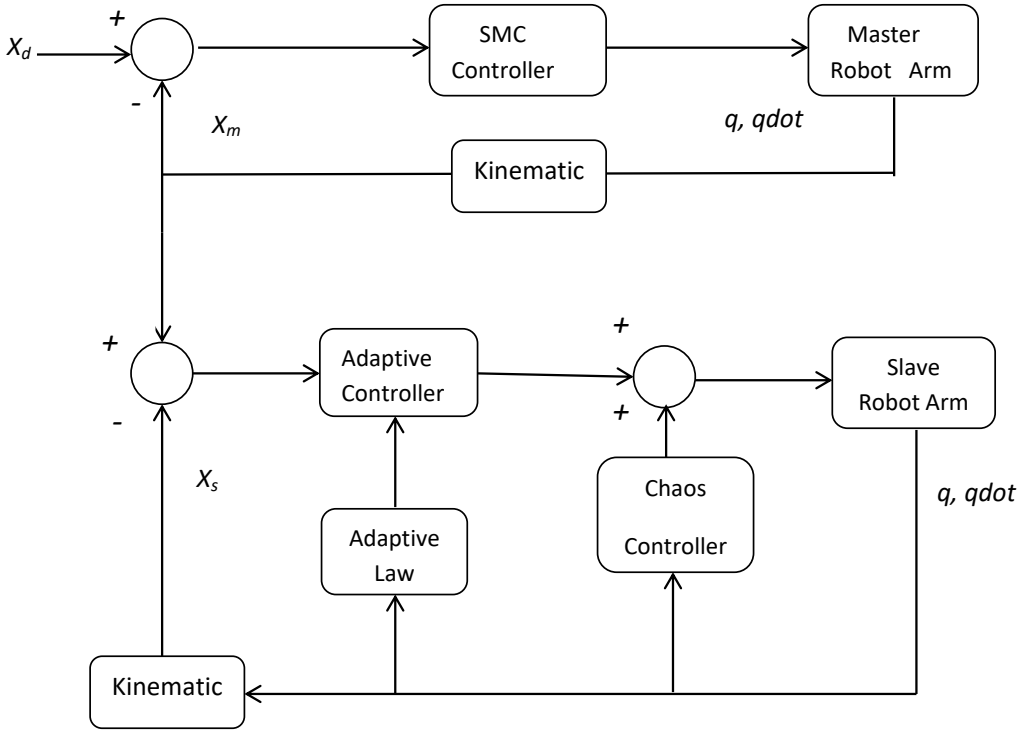


Figure 10. Block diagram of the scheme.

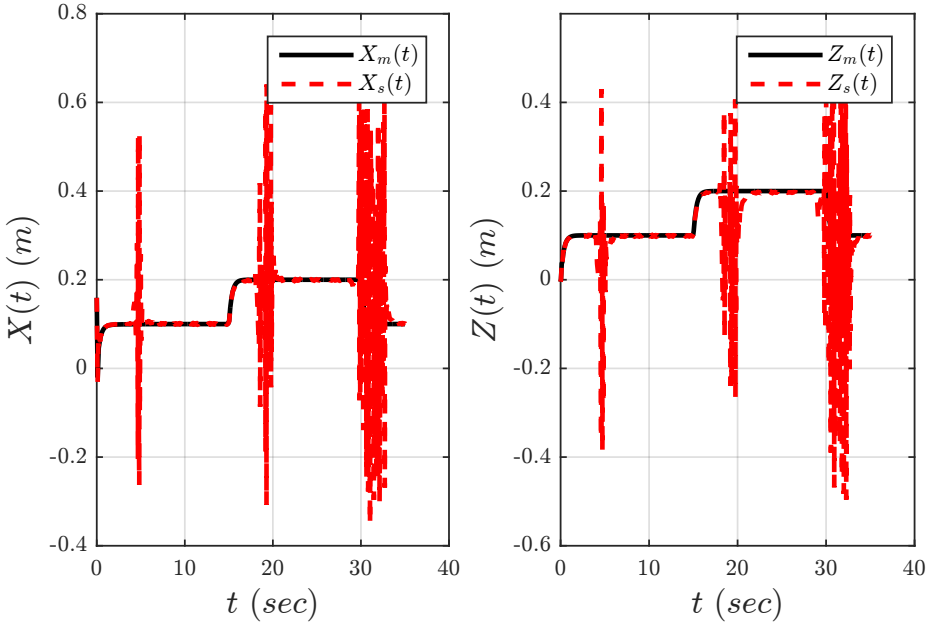


Figure 11. The x and z coordinates of the controlled slave end-effector position for $K_f = 1500$ and $\beta = 0.1$.

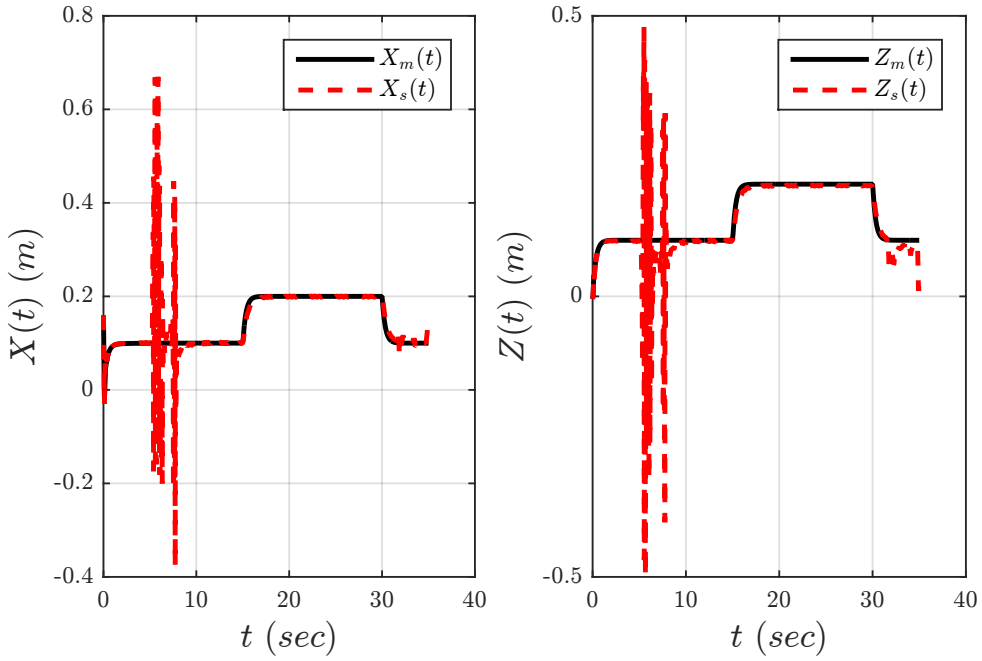


Figure 12. The x and z coordinates of the controlled slave end-effector position for $K_f = 1000$ and $\beta = 1$.

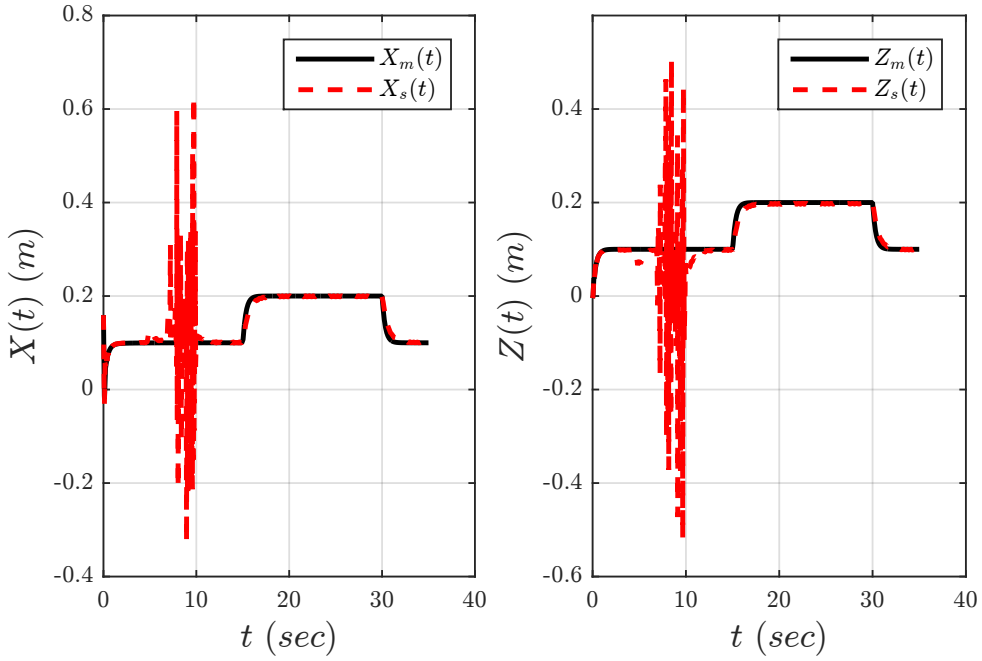


Figure 13. The x and z coordinates of the controlled slave end-effector position for $K_f = 1500$ and $\beta = 1$.

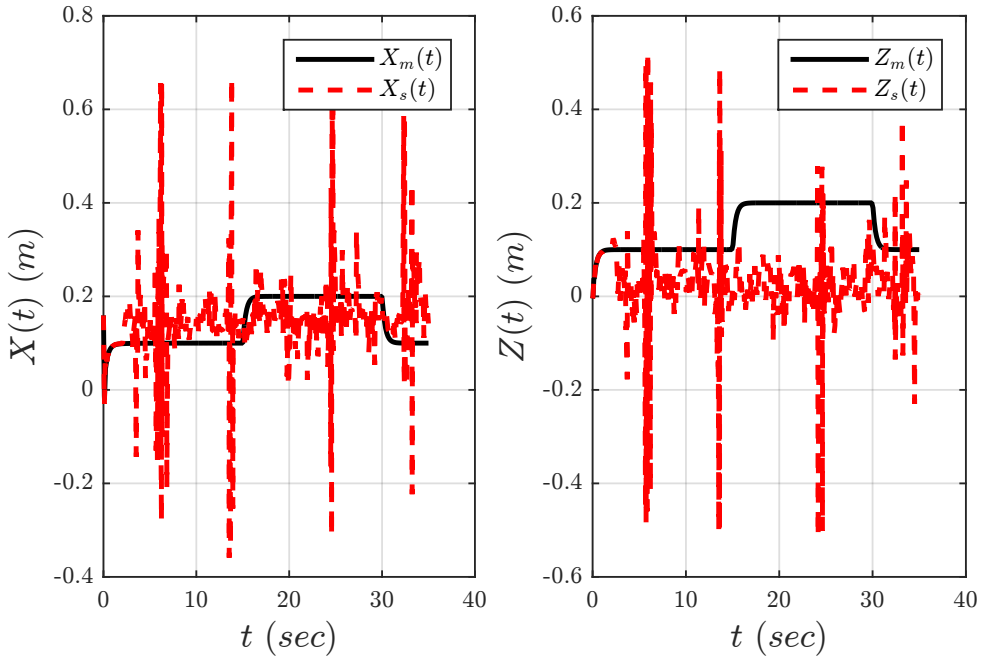


Figure 14. The x and z coordinates of the controlled slave end-effector position for $K_f = 6000$ and $\beta = 1$.

7. Discussion

As shown in Figures 5–7, when an fault occurs in one of the actuators driving the slave arm, the end-effector motion becomes chaotic. Such chaotic motion would lead to a complete collapse of the cooperative task. In addition, the uncontrolled motion of the arm may lead to undesired outcomes in terms of the safety of individuals in the vicinity as well as the completion of the required task. However, when the chaos controller (block diagram shown in Figure 8) is active, it overcomes the oscillations as demonstrated by simulation results shown in Figures 9–12. Currently there is no analytical method for optimizing the values of β and K_f, which means that appropriate values have to be selected through trial and error. We see that out of the parameter sets used in the experiments, the best tracking performance under a faulty joint is achieved with $\beta = 1$ and $K_f = 1500$. The paper is briefly summarized and concluded in the following paragraph.

8. Conclusions

Actuator faults during robot cooperative tasks would lead into a risky scenario. There need arises to minimize the impact of fault on tracking performance and complete the cooperative task successfully. In this paper a Master-Salve robot arms cooperative scenario is considered. During operation, when an actuator of the slave robot fails, it leads to the chaotic motion of the end effector. In this paper a possible remedy is presented in terms of a chaos controller. Which helps recover the tracking performance and help to complete the ongoing cooperative task.

The control scheme have some limitations. For instance, it will work well for a redundant system, i.e., if more DOF are available than required to complete the task (For Instance, in our case, we are controlling Cartesian X and Z position while we are employing four joints of the robots arms). In addition, the controller parameter β and K_f need to be tuned to get better results. This is a trail error process and involve some efforts. It will be good if these parameters are made adaptive or self-tuned. However, this is beyond the scope of the current work and it is potential future work.

Author Contributions: For research Conceptualization, S.G.K. and S.A.; methodology, S.G.K. and S.B.; software, S.G.K. and S.B.; formal analysis, S.G.K., S.B. and S.A.; writing, S.G.K. and S.B.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khan, S.G.; Bendoukha, S.; Mahyuddin, M.N. Dynamic Control for Human-Humanoid Interaction. In *Humanoid Robotics: A Reference*; Goswami, A., Vadakkepat, P., Eds.; Springer: Dordrecht, The Netherlands, 2017.
2. Peng, Y.C.; Carabis, D.S.; Wen, J.T. Collaborative manipulation with multiple dual-arm robots under human guidance. *Int. J. Intell. Robot. Appl.* **2018**, *2*, 252–266. [CrossRef]
3. Khan, S.G.; Herrmann, G.; Al Grafi, M.; Pipe, T.; Melhuish, C. Compliance Control and Human-Robot Interaction: Part 1—Survey. *Int. J. Hum. Robot.* **2014**, *11*, 1430001. [CrossRef]
4. Khan, S.G.; Herrmann, G.; Al Grafi, M.; Pipe, T.; Melhuish, C. Compliance Control and Human-Robot Interaction: Part II—Experimental Examples. *Int. J. Hum. Robot.* **2014**, *11*, 1430002. [CrossRef]
5. Khan, S.G.; Herrmann, G.; Pipe, T.; Melhuish, C. Adaptive multi-dimensional compliance control of a humanoid robotic arm with anti-windup compensation. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010; pp. 2218–2223.
6. Khan, S.G.; Herrmann, G.; Pipe, T.; Melhuish, C.; Spiers, A. Safe Adaptive Compliance Control of a Humanoid Robotic Arm with Anti-Windup Compensation and Posture Control. *Int. J. Soc. Robot.* **2010**, *2*, 305–319. [CrossRef]
7. Liu, G. Control of robot manipulators with consideration of actuator performance degradation and failures. In Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea, 21–26 May 2001; pp. 2566–2571.
8. Visinsky, M.L.; Cavallaro, J.R.; Walker, I.D. Robotic fault detection and fault-tolerance: A survey. *Reliab. Eng. Syst. Saf.* **1994**, *46*, 139–158. [CrossRef]
9. McIntyre, M.L.; Dixon, W.E.; Dawson, D.M.; Walker, I.D. Fault identification for robot manipulators. *IEEE Trans. Robot.* **2005**, *21*, 1028–1034. [CrossRef]
10. Xiao, B.; Yin, S. An Intelligent Actuator Fault Reconstruction Scheme for Robotic Manipulators. *IEEE Trans. Cybern.* **2018**, *48*, *639–647*. [CrossRef] [PubMed]
11. Yazdjerdi, P.; Meskin, N. Design and real-time implementation of actuator fault-tolerant control for differential-drive mobile robots based on multiple-model approach. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2018**, *6*, 652–661. [CrossRef]
12. Cho, C.N.; Hong, J.T.; Kim, H.J. Neural Network Based Adaptive Actuator Fault Detection Algorithm for Robot Manipulators. *J. Intell. Robot. Syst.* **2018**. [CrossRef]
13. Jin, L.; Li, S.; Luo, X.; Li, Y.; Qin, B. Neural Dynamics for Cooperative Control of Redundant Robot Manipulators. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3812–3821. [CrossRef]
14. Li, S.; Zhang, Y. *Neural Networks for Cooperative Control of Multiple Robot Arms*; Springer: Singapore, 2018.
15. Yang, C.; Jiang, Y.; Li, Z.; He, W.; Su, C. Neural Control of Bimanual Robots with Guaranteed Global Stability and Motion Precision. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1162–1171. [CrossRef]
16. Panwara, V.; Kumar, N.; Sukavanamc, N.; HwanBorm, J. Adaptive neural controller for cooperative multiple robot manipulator system manipulating a single rigid object. *J. Appl. Soft Comput.* **2012**, *12*, 216–227. [CrossRef]
17. Verduzco, F.; Alvarez, J. Homoclinic Chaos in 2-DOF Robot Manipulators Driven by PD Controllers. *Nonlinear Dyn.* **2000**, *21*, 157–171. [CrossRef]
18. Boubaker, O.; Jafari, S. *Recent Advances in Chaotic Systems and Synchronization*; Academic Press: New York NY, USA, 2018.
19. Goswami, A. From Being to Becoming: Entropy, Life, and Chaos Theory. In *The Physicists' View of Nature, Part 1*; Springer: New York, NY, USA, 2000.

20. Kocarev, L.; Lian, S. *Chaos-Based Cryptography: Theory, Algorithms and Applications*; Springer: Berlin/Heidelberg, Germany, 2011.
21. Zhu, S.; Zhu, C.; Wang, W. A New Image Encryption Algorithm Based on Chaos and Secure Hash SHA-256. *Entropy* **2018**, *20*, 716. [CrossRef]
22. Tereshko, V. Control and identification of chaotic systems by altering their energy. *Chaos Solitons Fractals* **2009**, *40*, 2430–2446. [CrossRef]
23. Azar, A.T.; Zhu, Q. *Advances and Applications in Sliding Mode Control Systems*; Studies in Computational Intelligence; Springer: Berlin/Heidelberg, Germany, 2015; Volume 576, ISBN 978-3-319-11172-8.
24. Shi, J.; Liu, H.; Bajcinca, N. Robust control of robotic manipulators based on integral sliding mode. *Int. J. Control* **2008**, *81*, 1537–1548. [CrossRef]
25. Jalani, J.; Herrmann, G.; Melhuish, C. Underactuated fingers controlled by robust and adaptive trajectory following methods. *Int. J. Syst. Sci.* **2014**, *45*, 120–132. [CrossRef]
26. Herrmann, G.; Jalani, J.; Mahyuddin, M.N.; Khan, S.G.; Melhuish, C. Robotic hand posture and compliant grasping control using operational space and integral sliding mode control. *Robotica* **2016**, *34*, 2163–2185. [CrossRef]
27. Slotine, J.J.E.; Li, W. *Applied Nonlinear Control*; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 1991.
28. Lewis, F.; Dawson, D.; Abdallah, C. *Robot Manipulator Control: Theory and Practice*; Marcel Dekker Inc.: New York, NY, USA, 2003.
29. Khan, S.G.; Jalani, J. Realisation of model reference compliance control of a humanoid robot arm via integral sliding mode control. *Mech. Sci.* **2016**, *7*, 1–8. [CrossRef]
30. Chen, G.; Song, Y.; Lewis, F.L. Distributed Fault-Tolerant Control of Networked Uncertain Euler-Lagrange Systems Under Actuator Faults. *IEEE Trans. Cybern.* **2017**, *47*, 1706–1718. [CrossRef] [PubMed]
31. Ott, E.; Grebogi, C.; Yorke, J.A. Controlling chaos. *Phys. Rev. Lett.* **1990**, *64*, 1196–1199. [CrossRef] [PubMed]
32. Hübinger, B.; Doerner, R.; Martienssen, W.; Herdering, M.; Pitka, R.; Dressler, U. Controlling chaos experimentally in systems exhibiting large effective Lyapunov exponents. *Phys. Rev. E* **1994**, *50*, 932–948. [CrossRef]
33. de Paula, A.S.; Savi, M.A. Comparative analysis of chaos control methods: A mechanical system case study. *Int. J. Non-Linear Mech.* **2011**, *46*, 1076–1089. [CrossRef]
34. Avanco, R.H.; Tusset, A.M.; Balthazar, J.M.; Nabarrete, A.; Navarro, H.A. On nonlinear dynamics behavior of an electro-mechanical pendulum excited by nonideal motor and chaos control taking into account parametric errors. *J. Braz. Soc. Mech. Sci. Eng.* **2018**, *40*, 23. [CrossRef]

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).