MRAS Based LFFC for a Two–Link Rigid Robot Arm

Nguyen Duy Cuong and Tran Xuan Minh
Electronics Faculty, Thai Nguyen University of Technology, Thai Nguyen City, Viet Nam
Email: {nguyenduycuong, tranxuanminh}@tnut.edu.vn

***Abstract*—This paper introduces a systematic robust control structure that consists of a Proportional Derivative (PD) Controller and a Model Reference Adaptive Systems (MRAS) based Learning Feed - Forward Control (LFFC) for nonlinear Multi-Input-Multi-Output (MIMO) systems with variable parameters, and significant coupling in the system dynamics. The purpose of using MRAS-based LFFC is to acquire the (stable part of the) inverse dynamics of the plant. By using Lyapunov theory the adaptive algorithm that is shown in this study is quite simple in its form, robust and converges quickly. Since it captures the system dynamics, the proposed controller has superior capability in efficient learning mechanism and dynamic response. An application design of a two – link rigid robot arm is carried out to demonstrate the effectiveness and robustness of the proposed control method.**

***Index Terms*–-model reference adaptive systems (MRAS), learning feed–forward control (LFFC), multi-input-multi-output (MIMO) systems, two–link robot arm**

I. Introduction

Two-degree-of-freedom robots are major devices in the manufacturing industry due to their several advantages including speed, accuracy, and repeatability [1]. We implicitly expected that we could give arbitrary desired trajectories and that these trajectories would be faithfully performed by the real-world robot. However, control of a two-link rigid robot arm to track accurately a desired trajectory is an extremely challenging due to the dynamics is highly non-linear and significant coupling [2]. In this paper, we look more closely at how to achieve a given joint trajectory on a robot manipulator.

Conventional PD controllers could be successfully applied to the tracking control for a two-link robotic arm [3]. It is often the first choice for a new controller design. The purpose of using PD controller is to stabilize the control system in its nature. However, fixed parameters in a PD controller do not have robust performance for control systems with parametric uncertainties, external disturbances, and coupled dynamics [4]. For accurate motion control, extended control methods are needed.

A typical controller for a high-precision motion system consists of a feed- forward controller and a feedback controller. The inputs to the feed-forward part are the states of the setpoint generator. The feed-forward controller generates a feed-forward signal by summing the profile setpoint signals with properly chosen weights. The feed-forward parameters are adjusted all the time. This implies that they follow changes in the process. As a result, it can be expected that a proper feed-forward controller signal is generated, effective for providing good tracking control performance. Note that, addition of the proper feed-forward component may improve performance, without affecting the stability, and robustness properties [5], [6].

The feed-forward part is considered as a function approximator whose input-output mapping can be adapted during control and is intended to become the (stable part of the) inverse of the plant [6]. It is clear that if an accurate model of the process is available, and if its inverse exists, then process dynamics can be canceled by the inverse model. As a result, the output of the process will be equal to the desired output if no other disturbances are present. In order to approximate the state dependent function, some kind of function approximator is introduced.

Neural Network (NN)-based LFFC has been widely regarded as one of the standard control paradigms for motion systems. The use of NN-based LFFC can improve not only the disturbance rejection, but also the stability robustness of the controlled systems. One of the main drawbacks of the NN-based LFFC is the requirement that the training motions are chosen carefully, such that all possibly relevant input combinations are covered. This requirement may be quite restrictive in practical applications. To overcome such problem, the use of MRAS-based LFFC can be applied [7], [8].

In this paper, in order to obtain high stability and fast convergence for the design of a linear process, the feed-forward part is proposed using adaptive components. The mechanism that adjusts the input-output mapping of the adaptive components is based on the tracking error. The well-known Lyapunov approach is used to find stable adaptive laws for the feed-forward parameters in such way that learning converges.

This paper is organized as follows. MRAS based LFFC is introduced in Section II. In Section III, the dynamics of a two-link rigid robot arm is shown. The design of the proposed controller is introduced in Section IV. At the end of this paper, summary of the paper is given.

Manuscript received April 15, 2014; revised July 20, 2014.

doi: 10.12720/joace.3.3.208-214

II. MRAS BASED LFFC

A state variable filter (SVF) can be used as the reference model to generate a set of profile setpoint signals including position, velocity, and acceleration. For the second-order case, such a filter is described by a second order transfer function. Note that, denominator coefficients should always have positive values to guarantee stabilization. By means of the feed-forward controller, the SVF output signals can be used to generate an inverse model of the process.

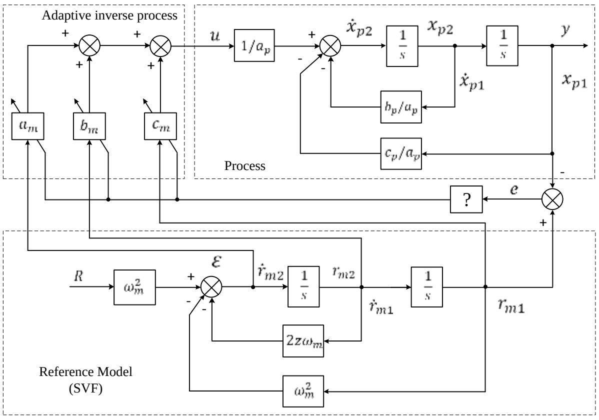

Figure 1. A process, an inverse process, and a reference model

A. Adaptive laws

In a model reference adaptive system the reference model can play the role of a setpoint generator. This leads to the structure of Fig. 1 [9], where the derivative generating structure of the state variable filter is clearly visible.

The reference model is described by

$$\frac{r_{m1}}{R} = \frac{\omega_m^2}{s^2+2z\omega_m s+\omega_m^2} \tag{1}$$

The process is described by

$$\frac{x_{p1}}{u} = \frac{\omega_p^2}{s^2+2z\omega_p s+\omega_p^2} = \frac{1}{a_p s^2+b_p s+c_p} \tag{2}$$

where

$$a_p = \frac{1}{\omega_p^2};\ b_p = \frac{2z}{\omega_p};\ c_p = 1. \tag{3}$$

Describe the process model in state variables

$$\dot{x}_p = \begin{bmatrix}\dot{x}_{p1}\\ \dot{x}_{p2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\dfrac{c_p}{a_p} & -\dfrac{b_p}{a_p}\end{bmatrix}\begin{bmatrix}x_{p1}\\ x_{p2}\end{bmatrix} + \begin{bmatrix}0\\ \dfrac{1}{a_p}\end{bmatrix}u$$

$$= A_p x_p + B_p u \tag{4}$$

where

$$A_p = \begin{bmatrix}0 & 1\\ -\frac{c_p}{a_p} & -\frac{b_p}{a_p}\end{bmatrix};\ B_p = \begin{bmatrix}0\\ \frac{1}{a_p}\end{bmatrix} \tag{5}$$

By means of the feed-forward controller, the SVF output signals can be used to generate an inverse model of the process. We should try to find a learning mechanism that, based on the errors between the output r_m of the setpoint generator and the process output x_p, adjusts the parameters a_m, b_m and c_m such that they converge to the process parameters a_p, b_p and c_p, respectively.

This suggests that we can use the well–known Lyapunov approach to find stable adaptive laws for the feed–forward parameters.The design problem is thus: Find (stable) adjustment laws for the adjustable parameters a_m, b_m and c_m such that the error e between the setpoint generator and the process as well as the error in the feed–forward parameters asymptotically go to zero. The following steps are thus necessary to design an adaptive controller with the method of Lyapunov [3]:

Step 1: Determine the differential equation for e

Describe the reference model in state variables

$$\dot{r}_{m1} = r_{m2};\ \dot{r}_{m2} = \varepsilon \tag{6}$$

$$\dot{r}_m = \begin{bmatrix}\dot{r}_{m1}\\ \dot{r}_{m2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}r_{m1}\\ r_{m2}\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\varepsilon = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}r_m + \begin{bmatrix}0\\ 1\end{bmatrix}\varepsilon \tag{7}$$

Rewrite the process model in state variables

$$\dot{x}_{p1} = x_{p2}$$

$$\dot{x}_{p2} = -\frac{c_p}{a_p}x_{p1} - \frac{b_p}{a_p}x_{p2} + \frac{1}{a_p}(c_m.r_{m1} + b_m.r_{m2}) + \frac{1}{a_p}a_m.\varepsilon \tag{8}$$

$$\dot{x}_p = \begin{bmatrix}\dot{x}_{p1}\\ \dot{x}_{p2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\dfrac{c_p}{a_p} & -\dfrac{b_p}{a_p}\end{bmatrix}\begin{bmatrix}x_{p1}\\ x_{p2}\end{bmatrix} + \begin{bmatrix}0 & 0\\ \dfrac{c_m}{a_p} & \dfrac{b_m}{a_p}\end{bmatrix}\begin{bmatrix}r_{m1}\\ r_{m2}\end{bmatrix} + \begin{bmatrix}0\\ \dfrac{a_m}{a_p}\end{bmatrix}\varepsilon$$

$$= \begin{bmatrix}0 & 1\\ -\frac{c_p}{a_p} & -\frac{b_p}{a_p}\end{bmatrix}x_p + \begin{bmatrix}0 & 0\\ \frac{c_m}{a_p} & \frac{b_m}{a_p}\end{bmatrix}r_m + \begin{bmatrix}0\\ \frac{a_m}{a_p}\end{bmatrix}\varepsilon \tag{9}$$

Here we introduce error e, which is defined in Eq. (10)

$$e = r_m - x_p;\ \dot{e} = \dot{r}_m - \dot{x}_p \tag{10}$$

By subtracting Eq. (8) from Eq. (9), we get

$$\dot{e} = \begin{bmatrix}0 & 1\\ -\frac{c_m}{a_p} & -\frac{b_m}{a_p}\end{bmatrix}r_m - \begin{bmatrix}0 & 1\\ -\frac{c_p}{a_p} & -\frac{b_p}{a_p}\end{bmatrix}x_p + \begin{bmatrix}0\\ 1-\frac{a_m}{a_p}\end{bmatrix}\varepsilon \tag{11}$$

$$= \begin{bmatrix}0 & 1\\ -\dfrac{c_m}{a_p} & -\dfrac{b_m}{a_p}\end{bmatrix}r_m - \begin{bmatrix}0 & 1\\ -\dfrac{c_p}{a_p} & -\dfrac{b_p}{a_p}\end{bmatrix}r_m + \begin{bmatrix}0 & 1\\ -\dfrac{c_p}{a_p} & -\dfrac{b_p}{a_p}\end{bmatrix}r_m - \begin{bmatrix}0 & 1\\ -\dfrac{c_p}{a_p} & -\dfrac{b_p}{a_p}\end{bmatrix}x_p + \begin{bmatrix}0\\ 1-\dfrac{a_m}{a_p}\end{bmatrix}\varepsilon$$

$$= \begin{bmatrix}0 & 0\\ \dfrac{c_p}{a_p}-\dfrac{c_m}{a_p} & \dfrac{b_p}{a_p}-\dfrac{b_m}{a_p}\end{bmatrix}r_m + \begin{bmatrix}0 & 1\\ -\dfrac{c_p}{a_p} & -\dfrac{b_p}{a_p}\end{bmatrix}(r_m - x_p) + \begin{bmatrix}0\\ 1-\dfrac{a_m}{a_p}\end{bmatrix}\varepsilon$$

$$= A_1.r_m + A.e + B.\varepsilon$$

where

$$A_1 = \begin{bmatrix} 0 & 0 \\ \frac{c_p}{a_p} - \frac{c_m}{a_p} & \frac{b_p}{a_p} - \frac{b_m}{a_p} \end{bmatrix}; A = \begin{bmatrix} 0 & 1 \\ -\frac{c_p}{a_p} & -\frac{b_p}{a_p} \end{bmatrix};$$
$$B = \begin{bmatrix} 0 \\ 1 - \frac{a_m}{a_p} \end{bmatrix} \tag{12}$$

Step 2: Choose a Lyapunov function $V(e)$

Simple adaptive laws are found when we use the Lyapunov function

$$V(e) = e^T P e + a^T \alpha a + b^T \beta b \tag{13}$$

where

P is an arbitrary definite positive symmetrical matrix; a and b are vectors which contain the non-zero elements of the A_1 and B matrices in Eq. (11); α and β are diagonal matrices with positive elements which determine the speed of adaptation.

Step 3: Determine the conditions under which $\dot{V}(e)$ is definite negative

$$\begin{aligned} \dot{V} &= (A_1.r_m + A.e + B.\varepsilon)^T.P.e \\ &\quad + e^T.P.(A_1.r_m + A.e + B.\varepsilon) \\ &\quad + 2.\dot{a}.\alpha.a^T + 2.\dot{b}.\beta.b^T \\ &= (A.e)^T.P.e + e^T.P.A.e \\ &\quad +2.e^T.P.A_1.r_m + 2.\dot{a}.\alpha.a^T \\ &\quad +2.e^T.P.B.\varepsilon + 2.\dot{b}.\beta.b^T \end{aligned} \tag{14}$$

Let:

$$A^T P + PA = -Q \tag{15}$$

According to Lyapunov's stability theory, as long as A is stable, there always exist such positive definite matrices P and$Q$. This implies that the first part of Eq. (14):

$$e^T(A^T P + PA)e = -e^T Q e \tag{16}$$

is definite negative. Such that stability of the system can be guaranteed if the two last parts of Eq. (14) get zero

$$e^T.P.A_1.r_m + \dot{a}.\alpha.a^T = 0 \tag{17}$$

$$e^T.P.B.\varepsilon + \dot{b}.\beta.b^T = 0 \tag{18}$$

where

$$a = [a_{21} \quad a_{22}]; e = [e_1 \quad e_2]; \alpha = \begin{bmatrix} \alpha_{11} & 0 \\ 0 & \alpha_{22} \end{bmatrix} \tag{19}$$

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}; A_1 = \begin{bmatrix} 0 & 0 \\ a_{21} & a_{22} \end{bmatrix}; r_m = \begin{bmatrix} r_{m1} \\ r_{m2} \end{bmatrix} \tag{20}$$

After some mathematical manipulations, this yields

$$\dot{a}_{21} = -\frac{1}{\alpha_{11}}.(e_1.p_{21} + e_2.p_{22}).r_{m1} \tag{21}$$

$$\dot{a}_{22} = -\frac{1}{\alpha_{22}}.(e_1.p_{21} + e_2.p_{22}).r_{m2} \tag{22}$$

From Eq. (12) it follows that

$$a_{21} = \frac{c_p}{a_p} - \frac{c_m}{a_p} \rightarrow \dot{a}_{21} = -\frac{1}{a_p}\dot{c}_m \tag{23}$$

It is given by the following expression to complete parameter update

$$c_m = \frac{a_p}{\alpha_{11}}\int[(e_1.p_{21} + e_2.p_{22}).r_{m1}]dt + c_m(0) \tag{24}$$

From Eq. (12) it also follows that

$$a_{22} = \frac{b_p}{a_p} - \frac{b_m}{a_p} \rightarrow \dot{a}_{22} = -\frac{1}{a_p}\dot{b}_m \tag{25}$$

There are given by the following expression to complete parameter update

$$b_m = \frac{a_p}{\alpha_{22}}\int[(e_1.p_{21} + e_2.p_{22}).r_{m2}]dt + b_m(0) \tag{26}$$

$$a_m = \frac{a_p}{\beta_{22}}\int[(e_1.p_{21} + e_2.p_{22}).\varepsilon]dt + a_m(0) \tag{27}$$

where α_{22} and β_{22} are called the adaptive gains, and e_1, e_2, ε, and r_{m2} are defined in Fig. 2; p_{21} and p_{22} are elements of the P matrix. The resulting adaptive system has been given (see Fig. 2).

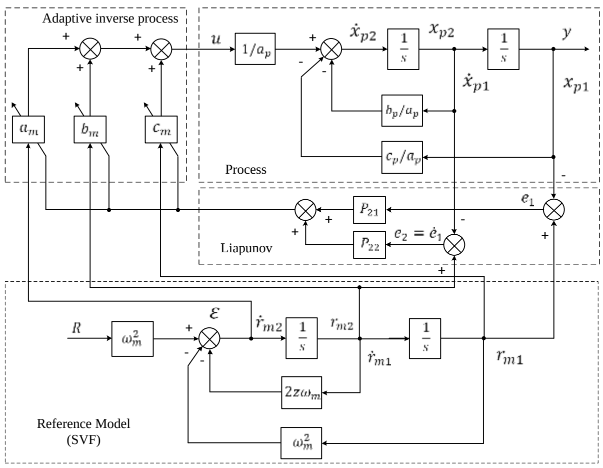

Figure 2. An adaptive inverse process designed with Lyapunov

Like in any MRAS-based system, adaptive disturbance compensation can be added, by realizing that the parameter d_m acts on an extra input signal 1, instead of on one of the state variables:

$$d_m = \frac{1}{\gamma}\int[(p_{21}.e_1 + p_{22}.e_2)1]dt + d_m(0) \tag{28}$$

Step 4: Solve P from $A_m^T P + PA_m = -Q$ (29)

Let

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \tag{30}$$

which yields the following matrix equation:

$$\begin{bmatrix} 0 & -\dfrac{c_p}{a_p} \\ 1 & -\dfrac{b_p}{a_p} \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -\dfrac{c_p}{a_p} & -\dfrac{b_p}{a_p} \end{bmatrix}$$
$$= -\begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \tag{31}$$

This yields

$$p_{21} = p_{12} = \frac{1}{2}.q_{11}.\frac{a_p}{c_p}; \quad p_{22} = \frac{1}{2}.\left(\frac{a_p^2}{c_p^2}.q_{11} + \frac{a_p}{c_p}.q_{22}\right) \tag{32}$$

Based on Eq. (24), Eq. (26), and Eq. (27) the adaptive system designed with Lyapunov in Fig 1 is redrawn as in Fig 2.

Eq. (24), Eq. (26), Eq. (27), and Eq. (28) can be generated to an equation for higher-order systems. For an n^{th} - order system we find for parameter $a_{m,i}$: [8, 9]

$$\frac{da_{m,i}}{dt} = \frac{1}{\alpha_{ii}}\left(\sum_{k=1}^{n} P_{nk} e_k\right) x_{m,i} \tag{33}$$

III. Two-Link Rigid Robot Arm

The robot arm is modeled as a set of n rigid bodies connected in series with one end fixed to the ground and the other end free. According to the Lagrange formulation, dynamic of an n- joint robot manipulator with revolute joints can be formulated as [2, 3]:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) + F(\dot{\theta}) = \tau \tag{34}$$

where

- $M(\theta)$ is the $(n \times n)$ manipulator inertia matrix;
- $C(\theta,\dot{\theta})$ is an $(n \times n)$ matrix describing centrifugal and Coriolis effects;
- $G(\theta)$ is an $(n \times 1)$ vector *modeling* gravity;
- $F(\dot{\theta})$ is an $(n \times 1)$ vector representing friction forces;
- $\theta, \dot{\theta}, \ddot{\theta}$ are the $(n \times 1)$ vector of joint positions, speeds, and accelerations, respectively;
- $\tau$is an $(n \times 1)$ vector of joint motor torques.

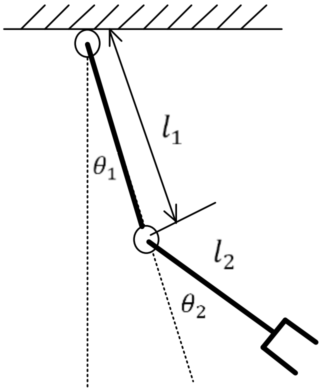

Figure 3. 2-DOF robot manipulator

The system that is considered in the paper is the two-link rigid robot arm with rotational joints shown in Fig. 3. Only the final non-linear differential equations are given below:

$$\begin{aligned}\tau_1 = {} & [l_1^2(m_1+m_2) + m_2 l_2^2 + 2m_2 l_1 l_2 cos\theta_2 + j_1 n_1^2]\ddot{\theta}_1 \\ & +(m_2 l_2^2 + m_2 l_1 l_2 cos\theta_2)\ddot{\theta}_2 - 2m_2 l_1 l_2 sin\theta_2 \dot{\theta}_2 \dot{\theta}_1 \\ & - m_2 l_1 l_2 sin\theta_2 \dot{\theta}_2 \dot{\theta}_2 + (m_1+m_2) g l_1 sin\theta_1 \\ & + m_2 g l_2 \sin(\theta_1+\theta_2) + v_1\dot{\theta}_1 + C_1 sign(\dot{\theta}_1)\end{aligned} \tag{35}$$

$$\begin{aligned}\tau_2 = {} & [m_2 l_2^2 + 2m_2 l_1 l_2 cos\theta_2]\ddot{\theta}_1 + (m_2 l_2^2 + j_2 n_2^2)\ddot{\theta}_2 \\ & + m_2 l_1 l_2 sin\theta_2 \dot{\theta}_1 \dot{\theta}_1 + m_2 g l_2 \sin(\theta_1+\theta_2) \\ & + v_2\dot{\theta}_2 + C_2 sign(\dot{\theta}_2)\end{aligned} \tag{36}$$

with

j_i= moments of inertia for electrical motor i
n_i=factor of reduction gear i
v_i= viscous friction for joint i
C= Coulomb friction for joint i

Equations (35) and (36) can be arranged to be

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -2h\dot{\theta}_2 + v_1 & -h\dot{\theta}_2 \\ h\dot{\theta}_1 & v_2 \end{bmatrix}\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{37}$$

where

$$M_{11} = l_1^2(m_1+m_2) + m_2 l_2^2 + 2m_2 l_1 l_2 cos\theta_2 + j_1 n_1^2 \tag{38}$$

$$M_{12} = M_{21} = m_2 l_2^2 + m_2 l_1 l_2 cos\theta_2 \tag{39}$$

$$M_{22} = m_2 l_2^2 + j_2 n_2^2 \tag{40}$$

$$h = m_2 l_1 l_2 sin\theta_2 \tag{41}$$

$$g_1 = (m_1+m_2) g l_1 sin\theta_1 + m_2 g l_2 \sin(\theta_1+\theta_2) \tag{42}$$

$$g_2 = m_2 g l_2 \sin(\theta_1+\theta_2) \tag{43}$$

In the above equations M_{ii} is the effective inertia; M_{ij} is the coupling inertia.

If the coupling and non-linear terms in Eq. (35) and Eq. (36) are ignored, this yields the following linear equations for Link 1 and Link 2, respectively:

$$\tau_1 = (l_1^2 m_1 + j_1 n_1^2)\ddot{\theta}_1 + v_1\dot{\theta}_1 \tag{44}$$

$$\tau_2 = (l_2^2 m_2 + j_2 n_2^2)\ddot{\theta}_1 + v_2\dot{\theta}_2 \tag{45}$$

From Eq. (44) and Eq. (45) the corresponding linear state equations of Link1 and Link 2 are:

$$\begin{bmatrix} \dot{\theta}_1 \\ \ddot{\theta}_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-v_1}{l_1^2 m_1 + j_1 n_1^2} \end{bmatrix}\begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{l_1^2 m_1 + j_1 n_1^2} \end{bmatrix}\tau_1 \tag{46}$$

$$\begin{bmatrix} \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-v_2}{l_2^2 m_2 + j_2 n_2^2} \end{bmatrix}\begin{bmatrix} \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{l_2^2 m_2 + j_2 n_2^2} \end{bmatrix}\tau_2 \tag{47}$$

IV. Design of Proposed Controller

Step 1 - Design the reference model (SVF 1 and SVF 2 in Fig. 4): Explicit position, velocity, and acceleration profile setpoint signals are created using the reference model, which is described by the transfer function

$$H_{ref}(S) = \frac{\omega_n^2}{s^2 + 2\gamma\omega_n s + \omega_n^2} \tag{48}$$

Step 2 - Design the feedback controller: As discussed in the previous sections, tracking performance is obtained by the MRAS-based LFFC. The feedback controller is designed such that it features robust stability for closed loop when used alone. The compensator that is used is of the PD-type:

$$C(s) = K_p + K_d s \tag{49}$$

Step 3 - Determine the inputs of the feed-forward part: The inputs of the Feed –Forward components depend on the nature of the plant and the reproducible disturbances that the LFFC has to compensate. For random motions, the inputs should consist of the reference position and its derivatives/integrals. The desired feed-forward signal is:

$$M(\theta_d)\ddot{\theta}_d + C(\theta_d, \dot{\theta}_d)\dot{\theta}_d + G(\theta_d) + F(\dot{\theta}_d) = \tau_d \quad (50)$$

A modified version of the feedback PD control law:

$$\underbrace{M(\theta_d)\ddot{\theta}_d + C(\theta_d, \dot{\theta}_d)\dot{\theta}_d + G(\theta_d)}_{u_{feed-forward}} + \underbrace{K_p e + K_d \dot{e}}_{u_{feedback}} = \tau_d \quad (51)$$

where $e^T = [e_1 \quad e_2]$, $\theta^T = [\theta_1 \quad \theta_2]$, $K_p = [K_{p1} \quad 0; \quad 0 \quad K_{p2}]$, $K_d = [K_{d1} \quad 0; \quad 0 \quad K_{d2}]$, $\theta_d^T = [\theta_{1d} \quad \theta_{2d}]$, $e_1 = \theta_{1d} - \theta_1$, $e_2 = \theta_{2d} - \theta_2$, θ_d is the vector of desired joint angels, and θ is the vector of real joint angels. Substituting (37) in (51) yields the following desired feed-forward signals:

$$\begin{bmatrix} M_{11d} & M_{12d} \\ M_{21d} & M_{22d} \end{bmatrix}\begin{bmatrix} \ddot{\theta}_{1d} \\ \ddot{\theta}_{2d} \end{bmatrix} + \begin{bmatrix} C_{11d} & C_{12d} \\ C_{21d} & C_{22d} \end{bmatrix}\begin{bmatrix} \dot{\theta}_{1d} \\ \dot{\theta}_{2d} \end{bmatrix} + \begin{bmatrix} G_{1d} \\ G_{2d} \end{bmatrix} +$$

$$\begin{bmatrix} K_{p1} & K_{d1} \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e_1 \\ \dot{e}_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ K_{p2} & K_{d2} \end{bmatrix}\begin{bmatrix} e_2 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} \tau_{1d} \\ \tau_{2d} \end{bmatrix} \quad (52)$$

From Eq. (35) and Eq. (36) we should select $[\theta_{1d}, \theta_{2d}, \dot{\theta}_{1d}, \dot{\theta}_{2d}, \ddot{\theta}_{1d}, \ddot{\theta}_{1d}, \cos(\theta_{2d}), \sin(\theta_{2d}), \sin(\theta_{1d} + \theta_{2d})]$as inputs of the feed-forward controller (see Table 1 and Fig. 4).

TABLE I: FEED-FORWARD COMPONENTS WITH CORRESPONDING INPUTS AND THE TARGET FUNCTIONS THAT THEY HAVE TO LEARN.

NO	INPUTS	LS	TARGET FUNCTIONS
FF_{11}	$cos\theta_{2d}\ddot{\theta}_{1d}$	u_1	$[(m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 cos\theta_2 + j_1 n_1^2]\ddot{\theta}_1$
FF_{12}	$cos\theta_{2d}\ddot{\theta}_{2d}$	u_1	$(m_2 l_2^2 + m_2 l_1 l_2 cos\theta_2)\ddot{\theta}_2$
FF_{13}	$sin\theta_{2d}\dot{\theta}_{1d}\dot{\theta}_{2d}$	u_1	$m_2 l_1 l_2 sin\theta_2\dot{\theta}_1\dot{\theta}_2$
FF_{14}	$sin\theta_{2d}\dot{\theta}_{2d}\dot{\theta}_{2d}$	u_1	$m_2 l_1 l_2 sin\theta_2\dot{\theta}_2\dot{\theta}_2$
FF_{15}	$sin\theta_{1d}$	u_1	$(m_1+m_2)gl_1 sin\theta_1$
FF_{16}	$\sin(\theta_{1d}+\theta_{2d})$	u_1	$m_2 g l_2 \sin(\theta_1+\theta_2)$
FF_{17}	$sign(\dot{\theta}_{1d})$	u_1	$C_1 sign(\dot{\theta}_1)$
FF_{21}	$cos\theta_{2d}\ddot{\theta}_{1d}$	u_2	$(m_2 l_2^2 + 2m_2 l_1 l_2 cos\theta_2)\ddot{\theta}_1$
FF_{22}	$\ddot{\theta}_{2d}$	u_2	$(m_2 l_2^2 + j_2 n_2^2)\ddot{\theta}_2$
FF_{23}	$sin\theta_{2d}\dot{\theta}_{1d}^2$	u_2	$m_2 l_1 l_2 sin\theta_2\dot{\theta}_1\dot{\theta}_1$
FF_{24}	$\sin(\theta_{1d}+\theta_{2d})$	u_2	$m_2 g l_2 \sin(\theta_1+\theta_2)$
FF_{25}	$sign(\dot{\theta}_{2d})$	u_2	$C_2 sign(\dot{\theta}_2)$

From Eq. (35), Eq. (36), and Eq. (51) the total torques applied to Joint 1 and Joint 2 are indicated in Eq. (53) and Eq. (54), respectively:

$$\tau_1 = \tau_{11} + \tau_{12} + \tau_{13} + \tau_{14} + \tau_{15} + \tau_{16} + \tau_{17} + \tau_{fb1} \quad (53)$$

$$\tau_2 = \tau_{21} + \tau_{22} + \tau_{23} + \tau_{24} + \tau_{25} + \tau_{fb2} \quad (54)$$

From Eq. (33) the adjustment laws of feed-forward components are to be

$$\tau_{11} = \alpha_{11}\int[(P_{21}^* e_1 + P_{22}^*\dot{e}_1)\, cos\theta_{2d}\ddot{\theta}_{1d}]dt + \tau_{11}(0)$$

$$\tau_{12} = \alpha_{12}\int[(P_{21}^* e_1 + P_{22}^*\dot{e}_1) cos\theta_{2d}\ddot{\theta}_{2d}]dt + \tau_{12}(0)$$

$$\tau_{13} = \alpha_{13}\int[(P_{21}^* e_1 + P_{22}^*\dot{e}_1)\, sin\theta_{2d}\dot{\theta}_{1d}\dot{\theta}_{2d}]dt + \tau_{13}(0)$$

$$\tau_{14} = \alpha_{14}\int[(P_{21}^* e_1 + P_{22}^*\dot{e}_1) sin\theta_{2d}\dot{\theta}_{2d}\dot{\theta}_{2d}]dt + \tau_{14}(0)$$

$$\tau_{15} = \alpha_{15}\int[(P_{21}^* e_1 + P_{22}^*\dot{e}_1) sin\theta_{1d}]dt + \tau_{15}(0)$$

$$\tau_{16} = \alpha_{16}\int[(P_{21}^* e_1 + P_{22}^*\dot{e}_1)\sin(\theta_{1d} + \theta_{2d})]dt + \tau_{16}(0)$$

$$\tau_{17} = \alpha_{17}\int[(P_{21}^* e_1 + P_{22}^*\dot{e}_1) sign(\dot{\theta}_{1d})]dt + \tau_{17}(0)$$

$$\tau_{fb1} = K_{p1}e_1 + K_{d1}\dot{e}_1$$

$$\tau_{21} = \alpha_{21}\int[(P_{21}^{**} e_2 + P_{22}^{**}\dot{e}_2)\, cos\theta_{2d}\ddot{\theta}_{1d}]dt + \tau_{21}(0)$$

$$\tau_{22} = \alpha_{22}\int[(P_{21}^{**} e_2 + P_{22}^{**}\dot{e}_2)\, \ddot{\theta}_{2d}]dt + \tau_{22}(0)$$

$$\tau_{23} = \alpha_{23}\int[(P_{21}^{**} e_2 + P_{22}^{**}\dot{e}_2)\, sin\theta_{2d}\dot{\theta}_{1d}^2]dt + \tau_{23}(0)$$

$$\tau_{24} = \alpha_{24}\int[(P_{21}^{**} e_2 + P_{22}^{**}\dot{e}_2)\, \sin(\theta_1 + \theta_2)]dt + \tau_{24}(0)$$

$$\tau_{25} = \alpha_{25}\int[(P_{21}^{**} e_2 + P_{22}^{**}\dot{e}_2)\, sign(\dot{\theta}_2)]dt + \tau_{25}(0)$$

$$\tau_{fb2} = K_{p2}e_2 + K_{d2}\dot{e}_2$$

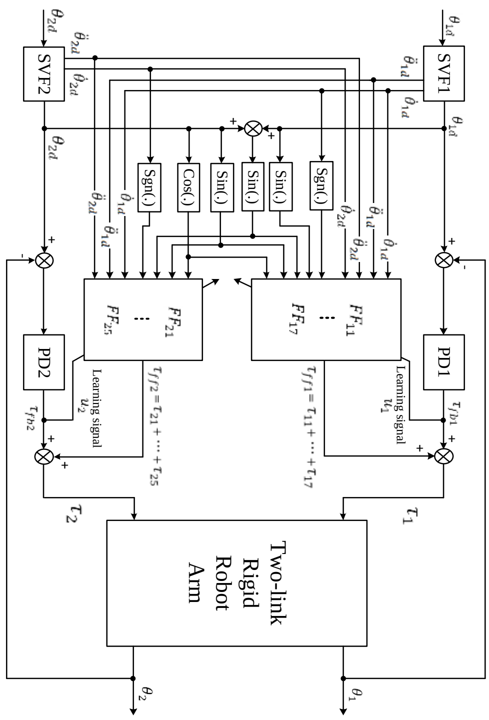

Figure 4. The proposed control system

Step 4 - Determine the structure of the feed-forward part: The proposed feed-forward controller should consist of 12 separate components (see Fig. 4). Since the learning filter (SVF) is not contained in the primary closed loop, it has only limited influence on the robustness.

Step 5 – Solve lyapunov equations: In the form of the adjustment laws P_{21}^{*}, P_{22}^{*}, P_{21}^{**} and P_{22}^{**} are elements of the P_1 and P_2 matrices, obtained from the solution of the Lyapunov equations indicated in Eq. (55) and Eq.(56)

$$A_{m1}^{T}P_1 + P_1A_{m1} = -Q_1 \qquad (55)$$

$$A_{m2}^{T}P_2 + P_2A_{m2} = -Q_2 \qquad (56)$$

where Q_1 and Q_2 are positive definite matrices and A_{m1} and A_{m2} are taken from Eq. (48), this yields

$$A_{m1} = \begin{bmatrix} 0 & 1 \\ -\omega_{n1}^2 & -2\gamma\omega_{n1} \end{bmatrix} \qquad (57)$$

$$A_{m2} = \begin{bmatrix} 0 & 1 \\ -\omega_{n2}^2 & -2\gamma\omega_{n2} \end{bmatrix} \qquad (58)$$

Step 6 - Choose adaptive gains: The adaptive gains which determine the speed of adaptation can in principle be chosen freely. However, un-modeled system dynamics limit these values in practice [2].

Step 7 - Training the LFFC: One component in the feed-forward part should be trained at a time. The reference motions should be chosen such that the desired feed-forward signal of one of untrained components becomes dominant. Only the parameters of the corresponding component are updated during this period, the parameters of the other components are kept constant.

Simulation results:

The parameter values of the considered two-link robotic arm are as follows: $m_1 = 0.5\ [kg]$, $m_2 = 0.5[kg]$, $l_1 = 0.4[m]$, $l_2 = 0.3[m]$, $g = 9.8\ [{}^{m}/_{s^2}]$, $j_1 = 0.5\ [kg * m^2]$, $j_2 = 0.3\ [kg * m^2]$, $v_1 = v_2 = 3\ [{}^{Ns}/_{m}]$, $C_1 = C_2 = 0.5\ [N]$, $n_1 = n_2 = 10$, $K_p = [8000 \quad 0; \quad 0 \quad 800]$, $K_d = [50 \quad 0; \quad 0 \quad 50]$, $Q_1 = [500 \quad 200; \quad 200 \quad 300]$, $Q_2 = [500 \quad 150; \quad 150 \quad 350]$. The desired tracking trajectories are supported to be: $\theta_{1d} = 10\sin(3\pi t)$ and $\theta_{2d} = 15\sin(2\pi t)$ for Link 1 and Link 2, respectively. The parameters of reference models are: $\omega_{n1} = \omega_{n2} = 10$ and $\gamma = 0.7$.

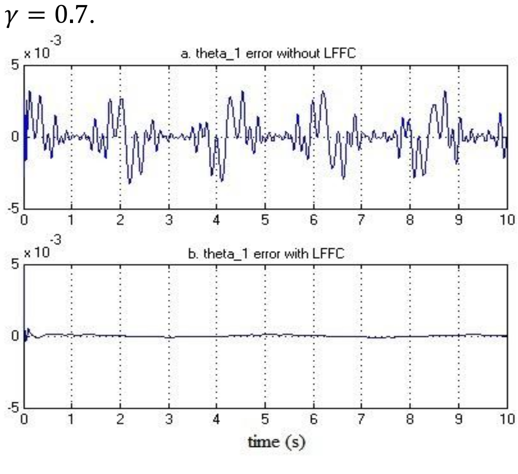

Figure 5. Comparison of the tracking error for theta_1 without (a) and with (b) LFFC.

Fig. 5a and Fig. 6a show the output responses using the PD controllers only. The control system is obviously stable. However, the tracking performance is not satisfactory. Fig. 5b and Fig. 6b show the results when the LFFCs are added. Clearly, the combination between PD controller and MRAS based Learning Feed-Forward controller indeed eliminates the tracking errors. In the beginning, the PD controller dominates. The LFFC controller takes less than 0.2 s to learn. The tracking performance is satisfactory.

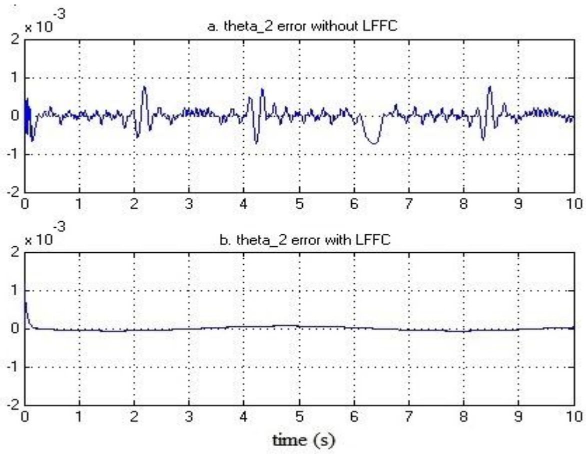

Figure 6. Comparison of the tracking error for theta_2 without (a) and with (b) LFFC.

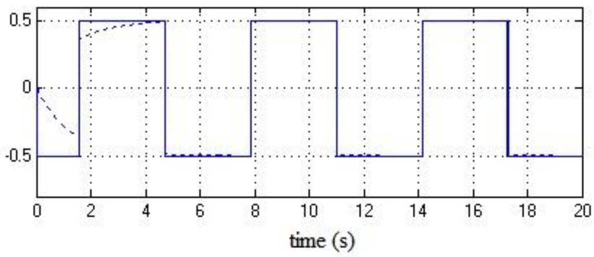

Figure 7. True Coulomb signal $\tau_{17} = C_1 sign(\dot{\theta}_1)$ (solid line) and corresponding estimated signal (dashed line).

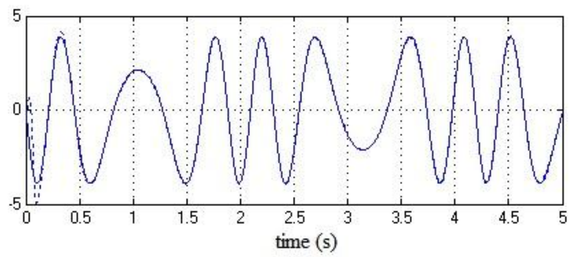

Figure 8. True non-linear signal $\tau_{15} = (m_1 + m_2)gl_1 sin\theta_1$(solid line) and corresponding estimated signal (dashed line).

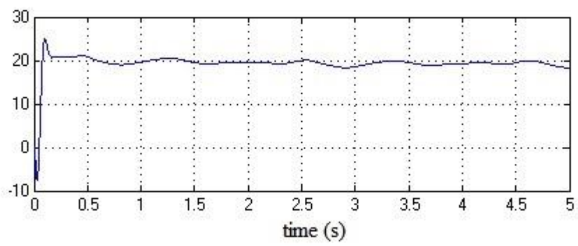

Figure 9. Adaptive gain in the feed-forward component FF_{15}.

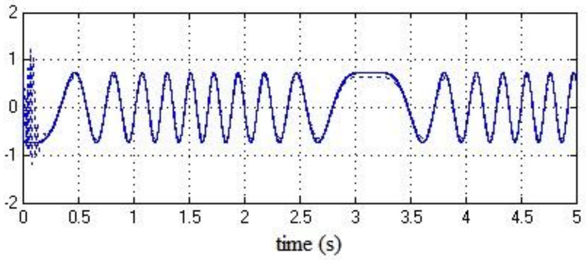

Figure 10. True non-linear coupling signal $\tau_{14} = m_2 l_1 l_2 sin\theta_2 \dot{\theta}_2 \dot{\theta}_2$(solid line) and corresponding estimated signal (dashed line).

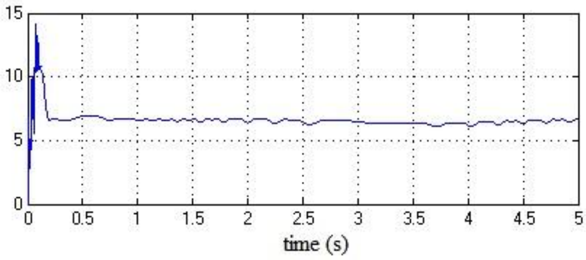

Figure 11. Adaptive gain in the feed – forward componentFF_{14}.

True disturbances in the plant are well compensated by corresponding estimated signals in the feed-forward path (see Fig. 7, Fig. 8, and Fig. 9). The adaptive gains in feed-forward components automatically reach stationary values (see Fig. 9 and Fig. 11).

V. Conclusion

The controller is proposed in this paper taking into account the inherent non-linear disturbances, variations and uncertainties in process behavior. It consists of a PD controller and a separate supplementary MRAS-based Learning Feed-Forward Controller (LFFC). A two-degree-of-freedom control structure is derived, which has properties in common with both the LFFC and the PD control laws. Compared to the NN-based LFFC, the MRAS-based LFFC is simpler to implement. Simulation results show that the effects of inertial loading, coupling reaction forces between joints, and gravity loading are well compensated by learning feed-forward signals.

References

[1] J. J. Craig, *Introduction to Robotics, Mechanics and Control*, 2^ Edition, Addison- Wesley, 1989.

[2] D. Chwa, J. Kang, and J. Y. Choi, "Online trajectory planning of robot arms for interception of fast maneuvering object under torque and velocity constraints," *IEEE Transactions on Systems, man, and cybernetics–part A: Systems and Humans*, vol. 35, no. 6, pp. 831–843, November 2005.

[3] G. R. Yu and L. W. Huang, "Design of LMI-based fuzzy controller for two-link robot arm using genetic algorithms," in *Proc. CACS International Automatic Control Conference National Cheng Kung University*, Tainan, Taiwan, Nov. 21-23, 2008.

[4] R. Burkan, "Modelling of bound estimation laws and robust controllers for robustness to parametric uncertainty for control of robot manipulators," *Journal of Intelligent and Robotic Systems*, vol. 60, pp. 365–394, 2010.

[5] D. Vries, J. A. Theo, L. J. Idema, and Velthuis, Parsimonious Learning Feed-Forward Control, European Symposium on Artificial Neural Networks ESANN, 1998.

[6] Velthuis and J. R. Wubbe, "Learning feed-forward control: Theory, design, and applications," PhD thesis, University of Twente, Enschede, The Netherlands, 2000.

[7] N. D. Cuong, T. J. A. de Vries, and J. van Amerongen, "Application of NN-based and MRAS-based FFC to electromechanical motion systems," in *Proc. EPE'13 ECCE Europe 15th European Conference on Power Electronics and Application*, 2013, pp. 1-10.

[8] J. Van Amerongen, "Intelligent control (Part 1)-MRAS, lecture notes," University of Twente, The Netherlands, March 2004.

[9] N. D. Cuong, N. V. Lanh, and D. V. Huyen, "Design of LQG controller using operational amplifiers for motion control systems," *Journal of Automation and Control Engineering*, pp. 300–306, 2014.

Nguyen Duy Cuong received the M.S. degree in Electrical Engineering from the Thai Nguyen University of Technology, Thai Nguyen city, Viet Nam, in 2001, the Ph.D. degree from the University of Twente, Enschede city, the Netherlands, in 2008.He is currently a lecturer with Electronics Faculty, Thai Nguyen University of Technology, Thai Nguyen City, Vietnam. His current research interests include real-time control, linear, parameter-varying systems, and applications in the industry.

Dr. Nguyen DuyCuong has held visiting positions with the Universityat Buffalo- the State Universityof New York (USA) in 2009 and with the Oklahoma State University (USA) in 2012.

Tran Xuan Minh received the M.S. degree, and the Ph.D degree in Automation and Control Engineering from the Ha Noi University of Science and Technology, Viet Nam, in 1997 and in 2008, respectively.

He is currently a lecturer with Electrical Faculty, Thai Nguyen University of Technology, Thai Nguyen City, Vietnam. His current research interests include electronics Power, Process Control, and applications in the industry.

Dr. Tran Xuan Minh has held visiting positions with the Oklahoma State University (USA) in 2013.

