Published in IET Control Theory and Applications
Received on 20th May 2009
Revised on 10th June 2010
doi: 10.1049/iet-cta.2009.0242

ISSN 1751-8644

Optimal state-feedback design for non-linear feedback-linearisable systems

P.M. Esfahani[1] *F. Farokhi*[2] *M. Karimi-Ghartemani*[3]

[1]*Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland*
[2]*Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran*
[3]*Department of Electrical and Computer Engineering, Queens University, Kingston, Ontario, Canada*
E-mail: karimig@queensu.ca

Abstract: This paper addresses the problem of optimal state-feedback design for a class of non-linear systems. The method is applicable to all non-linear systems which can be linearised using the method of state-feedback linearisation. The alternative is to use linear optimisation techniques for the linearised equations, but then there is no guarantee that the original non-linear system behaves optimally. The authors use feedback linearisation technique to linearise the system and then design a state feedback for the feedback-linearised system in such a way that it ensures optimal performance of the original non-linear system. The method cannot ensure global optimality of the solution but the global stability of the non-linear system is ensured. The proposed method can optimise any arbitrary smooth function of states and input, including the conventional quadratic form. The proposed method can also optimise the feedback linearising transformation. The method is successfully applied to control the design of a flexible joint dynamic and the results are discussed. Compared with the conventional linear quadratic regulator (LQR) technique, the minimum value of cost function is significantly reduced by the proposed method.

1 Introduction

Feedback linearisation is an important technique in the study of non-linear control systems. The purpose of feedback linearisation is to transform a given non-linear system into a linear system via state feedback. The exact state-feedback linearisation problem was pioneered and elegantly solved in [1–4]. Sufficient and necessary conditions for exact feedback linearisation of large classes of affine non-linear systems were established and documented in [5, 6].

To enlarge the class of non-linear systems that can be handled using the differential geometric approach, the dynamic feedback linearisation problem was initiated and addressed in [7] by introducing dynamic compensators and searching for the corresponding state and control transformations in the augmented state spaces. Sufficient conditions for dynamic feedback linearisation were given in [8] and necessary conditions were established in [9]. The partial feedback linearisation problem was formulated and studied in [10, 11] by identifying the largest feedback-linearisable subsystems, where conditions were given to transform a portion of the non-linear system into a linear part. When the relative degree of the considered non-linear system is less than system dimension, feedback linearisation-based non-linear control can also render the transformed system consisting of a non-linear zero dynamics plus a linear controllable system [5]. More recently, the non-regular feedback linearisation problem was defined in [12], where the purpose is to transform the non-linear system into the linear controllable form with reduced control input dimensions.

Feedback linearisation technique transforms the original non-linear system into a linear system. A stable controller for the linearised system will then also stabilise the original non-linear system. Performance of the non-linear system, however, is not directly related to that of the linear system and cannot be inferred from that. An optimal design such as linear quadratic regulator (LQR) for the linearised system, for instance, does not necessarily correspond to any optimality in the performance of the non-linear system. This paper addresses optimal state-feedback design for the feedback-linearised system to achieve optimal performance of the non-linear system. A technique is presented, which arrives at the solution for any arbitrary smooth cost function. The method is successfully applied to a flexible-joint system and results are discussed. The proposed controller achieves a performance that is about five times higher than the LQR design (in terms of the minimum point of the cost function). It is worthwhile noting that the stability offered by the feedback linearisation technique is a global stability rather than the local stability ensured by the traditional Jacobian linearisation. Therefore, even though the proposed method cannot guarantee arriving at a global optimum point, it ensures the global stability of the closed-loop system.

Arrangement of the paper is as follows. Section 2 provides a brief background into the concept of feedback linearisation and then states the problem encountered. The proposed technique to optimally design the state feedback is presented in Section 3 and a step-by-step algorithm is provided. Remarks on the convergence of the algorithm and

stability of the closed-loop system (while the algorithm is converging) are presented as well. A method for estimation of the Hessian matrix to increase the algorithm convergence rate is also developed. Section 4 presents some numerical results of the proposed technique in the context of an example and Section 5 concludes the paper.

2 Background and problem statement

Consider the affine non-linear system represented by

$$\dot{x} = f(x) + g(x)u \tag{1}$$

where x is the n-dimensional state vector, f and g are smooth vector fields on $\mathbb{R}^n$ and u is the scalar input signal. The feedback linearisation technique is based on applying

$$z = \phi(x), \quad v = \alpha(x) + \beta(x)u \tag{2}$$

where $z = \phi(x)$ is an admissible state transformation and v is the new control input signal. Upon using the linearising transformation ϕ and associated functions α and β, the representation (1) will change to

$$\dot{z} = Az + Bv \tag{3}$$

where

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & 0 & 1 \\ 0 & \cdots & \cdots & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{4}$$

The necessary and sufficient conditions for existence of $\phi(x)$, $\alpha(x)$ and $\beta(x)$ are given as follows [5, 13].

- The set of vector fields $\{g, ad_f g, \ldots, ad_{f^{n-1}} g\}$ is linearly independent over $\{\mathbb{R}^n\}$.
- The set $\{g, ad_f g, \ldots, ad_{f^{n-2}} g\}$ is involutive in $\{\mathbb{R}^n\}$.

In the above conditions, $ad_f g = [f, g]$, $ad_{f^{k+1}} g = [f, ad_{f^k} g]$ where $[f,g]$ stands for the Lie bracket as defined by $[f, g] = (\partial g/\partial x)f - (\partial f/\partial x)g$ [13]. A set of vector fields, called a distribution, is said to be involutive if the Lie bracket of every pair in that set belongs to the space spanned by all members of the set. If the above conditions hold, then a scalar function $h(x)$ on $\mathbb{R}^n$ exists, which satisfies

$$\frac{\partial h}{\partial x}\begin{bmatrix} g & ad_f g & \cdots & ad_{f^{n-2}} g \end{bmatrix} = 0 \tag{5}$$

In (5), $[\,g \quad ad_f g \quad \cdots \quad ad_{f^{n-1}} g\,]$ means a matrix, each column of which is the shown vector. Equation (5) then shows a set of $n-1$ partial differential equations. If h exists, the distribution is called completely integrable.

The desired transformation $\phi(x)$ and functions $\alpha(x)$ and $\beta(x)$ are then expressed in terms of $h(x)$ as [13, 14]

$$\begin{aligned} \phi(x) &= [\,h(x) \quad L_f h(x) \quad \cdots \quad L_{f^{n-1}} h(x)\,]^{\mathrm{T}} \\ \alpha(x) &= L_{f^n} h(x), \quad \beta(x) = L_g L_{f^{n-1}} h(x) \end{aligned} \tag{6}$$

In (6), the operator L_f shows the Lie derivative with respect to f defined as $L_f h = (\partial h/\partial x) f$ and $L_{f^k} h = L_f(L_{f^{k-1}} h)$.

The main point in proving the necessary and sufficient conditions for feedback linearisability is the equivalence of the involutivity condition and complete integrability of $\{g, ad_f g, \ldots, ad_{f^{n-2}} g\}$. This fact is a result of the Frobenius Theorem [5, 13] in differential geometry.

The control signal is then calculated from

$$u = a(x) + b(x)v = -\frac{\alpha(x)}{\beta(x)} + \frac{1}{\beta(x)} v \tag{7}$$

Lemma 1: The conditions for feedback linearisability imply $\beta(x) \neq 0$.

Proof: According to the involutivity condition, the distribution $\{g, ad_f g, \ldots, ad_{f^{n-2}} g\}$ is completely integrable; namely, there exists a smooth scalar function $h: \mathbb{R}^n \to \mathbb{R}$ such that $\nabla h(ad_{f^i} g) = L_{ad_{f^i} g} h = 0$ for all $i \in \{0, 1, \ldots, n-2\}$. According to the property of the Lie brackets which is

$$L_{ad_{f^k} g} = \sum_{i=0}^{k} (-1)^i \binom{k}{i} L_{f^{k-i}} L_g L_{f^i}$$

one can deduce that $L_{ad_{f^i} g} h = 0$, $i \in \{0, 1, \ldots, n-2\}$ is equivalent to $L_g L_{f^i} h = 0$, $i \in \{0, 1, \ldots, n-2\}$. Now, by contradiction, we assume there exists $x_0 \in \mathbb{R}^n$ that satisfies the feedback linearisation conditions but $\beta(x_0) = L_g L_{f^{n-1}} h(x_0) = 0$. This results in that $L_g L_{f^i} h|_{x=x_0} = 0$ for all $i \in \{0, 1, \ldots, n-1\}$. Again from the aforementioned property of the Lie brackets, one can deduce that $\nabla h(ad_{f^i} g)(x_0) = 0$ for all $i \in \{0, 1, \ldots, n-1\}$. This means that there exists a non-zero vector $\nabla h(x_0)$ (since $h(x)$ is smooth) such that $\nabla h(x_0)[g, ad_f g, \ldots, ad_{f^{n-1}} g] = 0$. This latter result contradicts the linear independence of $\{g, ad_f g, \ldots, ad_{f^{n-1}} g\}$ and thus, $\beta(x_0)$ cannot be zero.

Having transformed (1) into (3), the stabilisation problem can now simply be addressed by choosing

$$v = -K^{\mathrm{T}} z \tag{8}$$

where K is an $n \times 1$ constant vector such that all eigenvalues of $(A - BK^{\mathrm{T}})$ lie on the negative left-half of the complex plane. This selection of control input ensures stability of the original system (1). However, desired performance of the system (1) cannot be inferred from desired performance of the system (3). An LQR can for example be designed for the system (3) by properly selecting the vector K, but it may or may not result in an optimal (nor even a suboptimal) performance for the original system (1). The LQR optimal controller for the linear system (3) is the one that minimises the following cost function

$$J_z = \int_0^\infty (v^2 + z^{\mathrm{T}} \bar{Q} z) \mathrm{d}t \tag{9}$$

where Q is a positive-definite $n \times n$ matrix and T stands for matrix transposition. A solution to this problem exists since (A,B) in (4) is controllable. The solution can easily be obtained using the lqr command in Matlab. We used the subscript z to emphasise that this cost function is defined on z-space not on the original x-space. However, we are

doi: 10.1049/iet-cta.2009.0242

interested in a solution to minimise

$$J_x = \int_0^\infty (u^2 + x^{\mathrm{T}} Q x)\mathrm{d}t \tag{10}$$

which is the associated cost function in x-space. Formulating the general solutions to (10) is challenging because of the non-linearities involved. This paper addresses this problem and formulates a solution to (10) for the special case where the control input is of the form (8). In other words, we use the feedback linearisation technique to linearise the system but then we design the controller coefficients K to ensure the optimality of the non-linear system (quantified by J_x) rather than the linearised system (quantified by J_z). (Note that the controller u that (globally) minimises J_x need not necessarily correspond to the form given by (8). However, here only a solution is sought within all controllers u that correspond to the form of (8). Thus, the controller presented here will only be a suboptimal solution. Presenting the optimal solution (which is not confined to the form (8)) is in general challenging due to the non-linearities involved. One possible extension is to consider the form $v = -K_0^{\mathrm{T}} z + z^{\mathrm{T}} K_1 z$ as a more generalised version. The proposed technique of this paper is applicable to this form as well. However, we confine our study to the form of (8) for simplicity.

Problem Statement: For the non-linear affine system (1) with the original state vector x and the transformed system of (3) with the state vector of z, determine K in (8) which minimises J_x of (10).

3 Proposed method

3.1 Derivation of the algorithm

Using $v = -K^{\mathrm{T}} z = -K^{\mathrm{T}} \phi(x)$, the original system (1) can be represented as $\dot{z} = (A - BK^{\mathrm{T}})z$ in z-space. In x-space, it will be

$$\begin{aligned}\dot{x} &= f(x) + g(x)u = f(x) + g(x)[a(x) - b(x)K^{\mathrm{T}}\phi(x)] \\ &= f(x) + a(x)g(x) - b(x)g(x)K^{\mathrm{T}}\phi(x) \\ &= f_1(x) - g_1(x)K^{\mathrm{T}}\phi(x) \\ &= F(x, K)\end{aligned} \tag{11}$$

where $K = [k_1, k_2, \ldots, k_n]^{\mathrm{T}}$, $u = a(x) + b(x)v$, $a(x) = -\alpha(x)/\beta(x)$, $b(x) = 1/\beta(x)$ (for all x where $\beta(x) \neq 0$), $f_1(x) = f(x) + a(x)g(x)$ and $g_1(x) = b(x)g(x)$. Let us assume a general form for the cost function J as

$$J = \int_0^{T_f} \Gamma(x, K)\mathrm{d}t \tag{12}$$

where Γ is a function from $\mathbb{R}^{2n}$ to $\mathbb{R}$. For the quadratic case, we will have

$$\Gamma(x, K) = x^{\mathrm{T}} Q x + [a(x) - b(x)K^{\mathrm{T}}\phi(x)]^2 \tag{13}$$

Initial condition $x_0 = x(0)$ and final time T_f are assumed to be known. The objective is to reach a desired K that minimises J.

Define a new variable

$$x_{n+1}(t) = \int_0^{\mathrm{T}} \Gamma(x(\tau), K)\mathrm{d}\tau$$

It is clear that $x_{n+1}(0) = 0$ and $x_{n+1}(T_{\mathrm{f}})$ is equal to J in (12), which is to be minimised. Moreover, for all $0 < t < \mathrm{T_f}$

$$\dot{x}_{n+1}(t) = \Gamma(x(t), K) \tag{14}$$

Augmenting (11) and (14) yields

$$\dot{X} = H(X(t), K) \tag{15}$$

where X is the augmented $(n+1)$-dimensional state vector

$$X(t) = \begin{bmatrix} x(t) \\ x_{n+1}(t) \end{bmatrix} \tag{16}$$

and $H(X, K)$ is a function from $\mathbb{R}^{(2n+1)}$ to $\mathbb{R}^{n+1}$ given by

$$H(X, K) = \begin{pmatrix} \bar{F}(X(t), K) \\ \bar{\Gamma}(X(t), K) \end{pmatrix} = \begin{pmatrix} F(x(t), K) \\ \Gamma(x(t), K) \end{pmatrix} \tag{17}$$

The initial condition for (15) is $X_0 = [x_0, 0]^{\mathrm{T}}$ and the final time is T_f; both are known. Thus, the objective will now be to find K that minimises $x_{n+1}(T_f) = J$. The following theorem adopted from [13] provides sufficient conditions for existence of solutions.

Theorem 2 (Dependence on parameters): Let E be an open subset of $\mathbb{R}^{n+1} \times \mathbb{R}^n$ containing the point (X_0, K_0) where $X_0 \in \mathbb{R}^{n+1}$ and $K_0 \in \mathbb{R}^n$ and assume that $H \in C^1(E)$. It then follows that there exists an $a > 0$ and a $\delta > 0$ such that for all $\tilde{X}_0 \in N_\delta(X_0)$ and $K \in N_\delta(K_0)$, the initial value problem

$$\begin{aligned}\dot{X}(t) &= H(X(t), K) \\ X(0) &= \tilde{X}_0\end{aligned} \tag{18}$$

has a unique solution $X(t, \tilde{X}_0, K) \in C^1(G)$ where $G = [-a, a] \times N_\delta(X_0) \times N_\delta(K_0)$; furthermore, for each $K \in N_\delta(K_0)$, $X(t, X_0, K)$ is a continuously differentiable function of K.

Define

$$W = \frac{\partial X}{\partial K} \tag{19}$$

which implies that $W \in \mathbb{R}^{(n+1)\times n}$. Taking the time derivative of W in (19) and using the chain rule results in

$$\dot{W} = \frac{\partial H}{\partial X} \times \frac{\partial X}{\partial K} + \frac{\partial H}{\partial K} = \frac{\partial H}{\partial X} W + \frac{\partial H}{\partial K} \tag{20}$$

The matrices $\partial H/\partial X$ are equal to

$$\begin{pmatrix} \dfrac{\partial f_1}{\partial x} + \dfrac{\partial g_1}{\partial x}K^{\mathrm{T}}\phi + g_1 K^{\mathrm{T}}\dfrac{\partial \phi}{\partial x} & 0 \\ 2x^{\mathrm{T}}Q + 2\left(a + bK^{\mathrm{T}}\phi\right)\left(\dfrac{\partial a}{\partial x} + \dfrac{\partial b}{\partial x}K^{\mathrm{T}}\phi + bK^{\mathrm{T}}\dfrac{\partial \phi}{\partial x}\right) & 0 \end{pmatrix}$$

and

$$\frac{\partial H}{\partial K} = \begin{pmatrix} g_1(x)\phi^{\mathrm{T}}(x) \\ 2[a(x) + b(x)K^{\mathrm{T}}\phi(x)]b(x)\phi^{\mathrm{T}}(x) \end{pmatrix}$$

These two latter relationships are only valid for the quadratic cost function.

Notice that $W(0) = 0$ because X at $t = 0$ is independent from the choice of K. The last row of W, evaluated at $t = T_f$, is the gradient of J with respect to K, which shows variational behaviour of J with respect to changes in K.

Based on the above observations, it is now possible to propose an iterative algorithm to obtain the optimal K as follows.

3.1.1 Proposed algorithm:

- *Step 1.* Choose an initial value for K that ensures stability.
- *Step 2.* Jointly solve (15) and (20) with initial conditions $X(0) = [x^{\mathrm{T}}, 0]^{\mathrm{T}}$ and $W(0) = 0$.
- *Step 3.* Update K using the information at time T_f.

A proper initial value can, for example, be obtained by solving the LQR problem in z-space, which ensures the stability and, moreover, might have the chance of being close to the desired K. However, the only requirement when choosing the initial value is the closed-loop stability. Any K that places eigenvalues of $A - BK^{\mathrm{T}}$ in the stable region is acceptable. Step 2 involves a set of $(n + 1) + n(n + 1) = (n + 1)^2$ ordinary differential equations. The updating step can simply be done using the gradient descent rule as below

$$K_{i+1} = K_i - \mu_i(W_i^{n+1})^{\mathrm{T}} \tag{21}$$

where W_i^{n+1} is the last row of matrix W at stage i and μ_i is a positive definite matrix that controls the convergence rate of the algorithm. The algorithm stops when the gradient vector W_i^{n+1} becomes small enough.

3.2 Convergence of the algorithm

Assume that the cost function J, defined in (12), is a smooth function of K. Then its Taylor series expansion around the point K_i is

$$J(u) = J(K_i) + \frac{\partial J}{\partial K}(K_i)(K - K_i) + \mathcal{O}(K - K_i) \tag{22}$$

Evaluating J at K_{i+1} and substituting from (18) yields

$$J(K_{i+1}) = J(K_i) - \varepsilon_i W_i^{n+1}\bar{\mu}_i(W_i^{n+1})^{\mathrm{T}} + \mathcal{O}(\varepsilon_i) \tag{23}$$

where the real positive ε_i is a measure of the distance between K_i and K_{i+1}, $\bar{\mu}_i = \mu_i/\varepsilon_i$ and $\mathcal{O}(\varepsilon)/\varepsilon \to 0$ as $\varepsilon \to 0$. Since $\bar{\mu}_i > 0$, an $\varepsilon_0 > 0$ exists that for all $0 < \varepsilon_i < \varepsilon_0$

$$J(K_{i+1}) - J(K_i) = -\varepsilon_i\left\{W_i^{n+1}\bar{\mu}_i(W_i^{n+1})^{\mathrm{T}} + \frac{\mathcal{O}(\varepsilon_i)}{\varepsilon_i}\right\} < 0 \tag{24}$$

Thus, the sequence $\{J_i\}$ associated with the sequence $\{K_i\}$ is decreasing for sufficiently small ε_i. On the other hand, note that J is positive and thus J_i will have a lower bound. Thus, proper selection of μ_i at each iteration ensures that J_i converges at least to a local minimum. In general, a proper selection is that which corresponds to small steps taken from K_i to K_{i+1} and hence to small μ_i. More explanation on how to select μ_i based on the concept of Hessian matrix is given in Section 3.6.

3.3 Stability of the system

In this section, it is shown that proper selection of μ_i will guarantee stability of the closed-loop system. This is done by induction: assume that K_i maintains the closed loop stability, that is

$$\Re\{\lambda(A - BK_i^{\mathrm{T}})\} < 0 \tag{25}$$

where $\Re\{\cdot\}$ and $\lambda(\cdot)$ denote the real part and the set of eigenvalues, respectively. The matrixes A and B are defined in (4). The objective is to show that there exists an $\varepsilon_0 > 0$ that ensures the stability of the closed-loop system for iteration $i + 1$, that is $\Re\{\lambda(A - BK_{i+1}^{\mathrm{T}})\} < 0$ where

$$K_{i+1} = K_i - \varepsilon_i\bar{\mu}_i(W_i^{n+1})^{\mathrm{T}} \tag{26}$$

for all ε_i satisfying $0 < \varepsilon_i < \varepsilon_0$.

Multiplying equation (26) by B and adding A results in

$$(A - BK_{i+1}^{\mathrm{T}}) = (A - BK_i^{\mathrm{T}}) + \varepsilon_i BW_i^{n+1}\bar{\mu}_i \tag{27}$$

Thus, the eigenvalues of the system at iteration $(i + 1)$ are related to the eigenvalues at iteration i according to

$$\lambda(A - BK_{i+1}^{\mathrm{T}}) = \lambda((A - BK_i^{\mathrm{T}}) + \varepsilon_i BW_i^{n+1}\bar{\mu}_i) \tag{28}$$

On the other hand, it is a fact that the eigenvalues of a matrix are continuous functions of its entries. This means that the eigenvalues of the system at iteration $(i + 1)$ are continuous functions of ε_i which coincide with the eigenvalues at iteration i for $\varepsilon_i = 0$. Then if $\varepsilon_i \to 0$, the eigenvalues of $(A - BK_{i+1}^{\mathrm{T}})$ move continuously towards the eigenvalues of $(A - BK_i^{\mathrm{T}})$. Since K_i corresponds to a stable closed-loop system, there exists an ε_0 for which the closed-loop stability is guaranteed for all ε_i satisfying $0 < \varepsilon_i < \varepsilon_0$.

3.4 On selection of step-size

The above discussion shows that convergence and stability are ensured for a sufficiently small selection of step-size μ_i. Further improvements can be made using the so-called strong Wolfe conditions. Rewrite (21) in the form

$$K_{i+1} = K_i + \mu_i p_i \quad \text{where } p_i = -(W_i^{n+1})^{\mathrm{T}} \tag{29}$$

Lemma 3 [15]: Suppose that $J: \mathbb{R}^{2n} \to \mathbb{R}$ is continuously differentiable. Then p_i defined in equation (29) is a descent direction at K_i. This means that

$$J(K_{i+1}) \leq J(K_i) \tag{30}$$

for sufficiently small μ_i.

doi: 10.1049/iet-cta.2009.0242

Definition 4 [15]: The strong Wolfe conditions require μ_i to satisfy

$$J(K_i + \mu_i p_i) \leq J(K_i) + c_1 \mu_i p_i^{\mathrm{T}} \nabla J(K_i) \tag{31}$$

$$|p_i^{\mathrm{T}} \nabla J(K_i + \mu_i p_i)| \leq c_2 |p_i^{\mathrm{T}} \nabla J(K_i)| \tag{32}$$

with $0 < c_1 < c_2 < 1$. The first inequality is called the sufficient decrease condition and the second one is called the curvature condition.

Lemma 5 [15]: Suppose that $J: \mathbb{R}^{2n} \to \mathbb{R}$ is continuously differentiable. Let p_i be a descent direction at K_i, and assume that J is bounded below along the ray $\{K_i + \mu p_i | \mu > 0\}$. Then if $0 < c_1 < c_2 < 1$, there exist intervals of step lengths satisfying the strong Wolfe conditions.

3.5 Rate of convergence

It is generally difficult to predict the rate of convergence of the algorithm. The following theorem provides hints on this issue.

Theorem 6 [15]: Suppose that $J: \mathbb{R}^{2n} \to \mathbb{R}$ is twice continuously differentiable, and that the iterates generated by the steepest descent method with exact line search ($\mu_i = \mathrm{argmin}\{J(K_i + \mu p_i)\}$) converge to a fixed point K^* where the Hessian matrix $\nabla^2 J(K^*)$ is positive definite. Then

$$|J(K_{n+1}) - J(K^*)| \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 |J(K_n) - J(K^*)| \tag{33}$$

where $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the eigenvalues of the Hessian matrix $\nabla^2 J(K^*)$.

In general we cannot expect the rate of convergence to improve if an inexact line search based on strong Wolfe conditions is used.

3.6 Estimation of Hessian matrix

The previous discussion in Sections 3.2 and 3.3 shows that convergence of the proposed algorithm and stability of the closed-loop system are guaranteed as long as the matrix μ_i in (21) is positive definite and small for all iterations. The most straightforward selection for the positive-definite matrix μ_i is $\mu_i = \epsilon I$, where I is the identity matrix and ϵ is a small positive number. At every iteration i, the value of ϵ is decreased if the norm of the gradient vector does not decrease or if the stability condition is violated. Both these conditions are easy to verify. However, the algorithm with this process may become too slow and the convergence rate will become low. This section provides an alternative for choosing the positive definite matrix μ_i based on the concept of the Hessian matrix, which results in a fast convergence rate.

It is well known that, for a quadratic cost function, the best option for the matrix μ_i in the gradient method is the inverse of the Hessian matrix $H_i = \nabla^2 J(K_i) = (\partial^2 J / \partial K^2)(K_i)$. The difficulty is, however, in calculating this matrix. Here a method is presented that estimates the Hessian matrix using m previous iterations of the algorithm. Denote the gradient function of J as $\nabla J(K) = (\partial J / \partial K)(K)$. Then a linear approximation yields

$$\nabla J(K) \simeq (K - K_i)^{\mathrm{T}} H_i + \nabla J(K_i) \tag{34}$$

We pick the best H_i that satisfies this approximation at m iterations prior to K_i, that is $\{K_{i-1}, \ldots, K_{i-m}\}$. Such an H_i is the one that minimises

$$\min_{H_i \in \Re^{n \times n}} \|\Theta_i - \Delta_i H_i\| \tag{35}$$

where Θ_i and Δ_i are $m \times n$ matrixes and defined as

$$\Theta_i = \begin{pmatrix} \nabla J(K_i) - \nabla J(K_{i-1}) \\ \nabla J(K_i) - \nabla J(K_{i-2}) \\ \cdots \\ \nabla J(K_i) - \nabla J(K_{i-m}) \end{pmatrix}, \quad \Delta_i = \begin{pmatrix} (K_i - K_{i-1})^{\mathrm{T}} \\ (K_i - K_{i-2})^{\mathrm{T}} \\ \cdots \\ (K_i - K_{i-m})^{\mathrm{T}} \end{pmatrix} \tag{36}$$

The solution to (35) is the pseudoinverse of Δ_i multiplied by Θ_i. Hence, the Hessian matrix is approximated as

$$H_i = (\Delta_i^{\mathrm{T}} \Delta_i)^{-1} \Delta_i^{\mathrm{T}} \Theta_i \tag{37}$$

There is, however, no guarantee that H_i evaluated from (37) remains symmetric and positive definite. To overcome these two problems, the Hessian matrix can be obtained from the modified equation

$$\bar{H}_i = \alpha I + \frac{H_i + H_i^{\mathrm{T}}}{2} \tag{38}$$

where α is positive gain and greater than $-\lambda_{\min}((H_i + H_i^{\mathrm{T}})/2)$ if $(H_i + H_i^{\mathrm{T}})/2$ is not positive definite. When $(H_i + H_i^{\mathrm{T}})/2$ is positive definite, then $\alpha = 0$. The matrix $\bar{H}_i$ in (38) may or may not be an exact estimation of the Hessian matrix; nevertheless, it can carry the information of the Hessian matrix and hence can result in a much quicker convergence of the algorithm.

3.7 Further discussion

The gradient method is very simple to implement and its memory usage is very low, making it attractive for large systems. The linear rate of convergence and the possibility of local minima are, however, its main drawbacks. Higher order methods such as Newton's method and quasi-Newton methods can be used to achieve faster convergence at the cost of more complexity. The idea of estimating the Hessian matrix in Section 3.6 is the same as the quasi-Newton method and it increases the convergence speed-specially around the optimal point. When the controller parameters are far away from the optimal parameters, the estimated Hessian may not be positive definite, and then according to the proposed modifications, the search direction may become close to the simple scalar step-size because the αI term is dominant. But as the parameters become closer to the optimal point, the estimated Hessian matrix becomes similar to the real Hessian matrix and the convergence rate becomes super-linear.

One may use any other numerical optimisation technique such as the conjugate gradient approach to approximate the Hessian matrix to improve the convergence rate. Furthermore, the Hessian matrix can be precisely computed using the sensitivity analysis approach too. Namely, similar to equation (20), one can derive the corresponding ordinary differential equations for the Hessian matrix and compute it at the terminal time, which in turn require solving

$(n+1)(n^2+n+1)$ ordinary differential equations (ODEs) that is $(n+1)n^2$ more ODEs than the proposed algorithm of the paper.

4 Numerical results

This section studies the feasibility of the proposed algorithm in the context of a flexible-joint single-link arm. In this study, the ability of the proposed algorithm to handle non-quadratic index functions is also illustrated. Moreover, an extension of the proposed algorithm to optimising the feedback linearising transformation $z = \phi(x)$ is also carried out.

4.1 Case study

Consider the mechanical system of Fig. 1 which represents a link driven by a motor through a torsional spring conveniently called a single-link flexible-joint manipulator in the vertical plane [14]. The motion equations of the manipulator are

$$\begin{cases} I\ddot{q}_1 + MgL\sin(q_1) + k(q_1 - q_2) = 0 \\ J\ddot{(q_2)} - k(q_1 - q_2) = u \end{cases} \tag{39}$$

Defining the state vector $x = [q_1\ \dot{q}_1\ q_2\ \dot{q}_2]^{\mathrm{T}}$, the equations can be easily derived by a fourth-order model of the form $\dot{x} = f(x) + g(x)u$ in which f and g are given as $f = [x_2,\ -a\sin(x_1) - b(x_1 - x_3),\ x_4,\ c(x_1 - x_3)]^{\mathrm{T}}$ and $g = [0,\ 0,\ 0,\ d]^{\mathrm{T}}$ where a,b,c,d are some positive constants determined by the link physical quantities as described by $a = MgL/I$, $b = k/I$, $c = k/J$, $d = 1/J$. The state vector is defined as $x = [q_1\ \dot{q}_1\ q_2\ \dot{q}_2]^{\mathrm{T}}$ and the input signal is u. The parameters M, I, L, k, J, variables q_1, q_2 and the input signal u are shown in Fig. 1. In view of the fact that the unforced system has an equilibrium point at $x = 0$, we expect to find a finite energy input to regulate the arm from any arbitrary 'small' initial value to zero. Simple derivations show that

$$\begin{aligned} ad_f g &= [f, g] = -\frac{\partial f}{\partial x} g = [0\ 0\ -d\ 0]^{\mathrm{T}} \\ ad_{f^2} g &= [f, ad_f g] = -\frac{\partial f}{\partial x} ad_f g = [0\ bd\ 0\ -cd]^{\mathrm{T}} \\ ad_{f^3} g &= [f, ad_{f^2} g] = -\frac{\partial f}{\partial x} ad_{f^2} g = [-bd\ 0\ cd\ 0]^{\mathrm{T}} \end{aligned} \tag{40}$$

It is easy to verify that the matrix $(g\ ad_f g\ ad_{f^2} g\ ad_{f^3} g)$ has full rank for all $x \in \mathbb{R}^4$ and the set $\{g, ad_f g, ad_{f^2} g\}$ is involutive for all $x \in \mathbb{R}^4$. Thus, both conditions for the existence of a feedback linearising transformation are satisfied for all $x \in \mathbb{R}^4$ that guarantees existence of a function $h(x)$ satisfying $(\partial h/\partial x)(g, ad_f g, ad_{f^2} g) = 0$. Expansion of these partial differential equations will restrict $h(x) = h(x_1)$, that is only a function of x_1. The control signal is obtained from $h(x)$ using (2) as below

$$u = \frac{v - L_{f^4} h(x)}{L_g L_{f^3} h(x)} = \frac{-\sum_{i=1}^{4} k_i z_i - L_{f^4} h(x)}{L_g L_{f^3} h(x)} \tag{41}$$

Note that the new input signal $v = -K^{\mathrm{T}} z = -\sum_{i=1}^{4} k_i z_i$ is substituted in (41) where the relationship between the new state vector z and $h(x)$ is given by (2) and (6). One can deduce from (41) that scaling the function $h(x)$ does not change the control signal; thus, without loses of generality it can be assumed that $h(0) = 0$ and $(\partial h/\partial x_1)(0) = 1$. A selection of $h(x) = x_1$ holds for all conditions and is considered here. In Section 4.4, we will discuss how to optimally choose $h(x)$ as well. Thus

$$\begin{aligned} z_1 &= h(x) = \phi_1(x) = x_1 \\ z_2 &= \phi_2(x) = L_f z_1 = x_2 \\ z_3 &= \phi_3(x) = L_f z_2 = -a\sin x_1 - bx_1 + bx_3 \\ z_4 &= \phi_4(x) = L_f z_3 = -ax_2\cos x_1 - bx_2 + bx_4 \end{aligned} \tag{42}$$

The control signal u is now a function of x and K as $u = u(x, K)$ and the objective is to locate the optimal K that minimises the cost function

$$J(K) = \int_0^\infty [u^2(x, K) + x^{\mathrm{T}} Q x]\mathrm{d}t \tag{43}$$

for a given positive-definite matrix Q. The stability requirement on the eigenvalues of $A - BK^{\mathrm{T}}$ poses a constraint on the elements of K as

$$k_4 k_3 k_2 - (k_2^2 + k_4^2 k_1) > 0, \quad k_i > 0, \quad i = 1, 2, 3, 4 \tag{44}$$

The constraint (44) is derived using the Routh–Hurwitz criterion and ensures stability of the closed-loop system. Using the same notations introduced in the previous section, we have

$$\begin{aligned} F(x, K) &= f(x) + g(x)u(x, K) \\ \Gamma(x, K) &= u(x, K)^2 + \beta\|x\|_2^2 \end{aligned} \tag{45}$$

where a selection of $Q = \beta I$ ($\beta > 0$) is made for simplicity.

4.2 General simulations

The results of computer simulations of the proposed algorithm on the above case study are presented in this section. Numerical values of simulation are $\beta = 3$ (for the cost function), $x_0 = [1\ 0.7\ 0.1\ 0.2]$ (initial state) and $T_f = 40$s (final time). The physical parameters a, b, c, d are estimated as 5, 0.5, 0.1, 1, respectively, from the real physical quantities. The initial value for controller coefficients K is randomly selected as long as the stability requirement (44) is satisfied. Two cases are considered: Case I. $\mu_i = \varepsilon_i I$. Case II. $\mu_i = H_i^{-1}$. For the first case, ε_i is primarily set at 0.0001 and it remains constant as long as the sequences of $J_i = J(K_i)$ are decreasing. When this sequence does not decrease, ε_i is scaled by γ where $\gamma < 1$. We chose $\gamma = 0.9$ in this simulation. The scaling with γ keeps going until the sequence again starts decreasing.

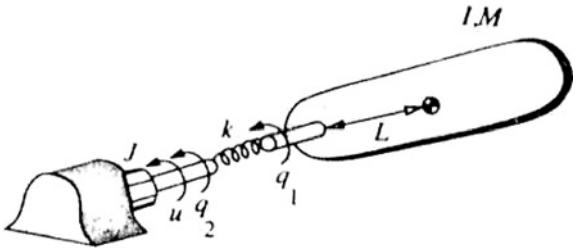

Fig. 1 *A flexible-joint mechanism [14]*

doi: 10.1049/iet-cta.2009.0242

In the second case, the parameter α in (38) is set at 0.1 when $(H_i + H_i^T)/2$ is not positive definite.

In Fig. 2, the solid curve and dashed curve show the evolution of the cost function for Cases I and II, respectively. Also, Fig. 3 shows the variations of the norm of the gradient vectors as iterations go on for both cases. In Case II, first the algorithm is run with fixed step size (similar to Case I) for the first m iterations in order to have m previous iterations information to estimate the Hessian matrix. That is why both simulations show identical results till the fifth iteration (m =5). Case II, which is based on estimation of the Hessian matrix, converges significantly faster than the first case. Within about 15 iterations, the gradient vector becomes sufficiently small and the algorithm can be stopped. The obtained controller is $K_{opt} = [1.92\ 5.05\ 5.89\ 1.23]$ and the minimum value of the cost function is about $J_{min} = 422$.

To compare the results with those of an optimally designed LQR controller, the Matlab command lqr is used to obtain the optimal controller $K_{LQR} = [1.732, 4.943, 6.1897, 3.9216]$ with the same β. This controller results in a value of $J_{LQR} = 1630$ for the same cost function $J(K)$ given in (43). The index function is improved about four times by the proposed controller.

The time responses of the closed-loop control systems, using both the designed controller, using the proposed algorithm and the LQR controller, are obtained and shown in Figs. 4 and 5. In Fig. 4 parts (a) and (b) depict the state variables, and in Fig. 5 parts (a) and (b) show the control signals for both methods. Comparing with the LQR responses, variations of the state variables as well as the control signal are within a much smaller range in the proposed controlled system.

We also repeated the simulation for different initial conditions. The results confirm robustness of the controller gains to a wide range of initial conditions. Some cases are reported below as examples.

$$x_0 = [0.5, 1, -0.1, 0.1]^T, \quad K_{opt} = [1.96, 5.56, 6.26, 1.46]^T$$

$$J(K_{LQR}) = 1216, \quad J(K_{opt}) = 146$$

$$x_0 = [0.7, -1, 0.2, 0.4]^T, \quad K_{opt} = [1.98, 5.58, 6.23, 1.41]^T$$

$$J(K_{LQR}) = 2029, \quad J(K_{opt}) = 374$$

$$x_0 = [0.4, 0.4, 0.1, 0.1]^T, \quad K_{opt} = [1.96, 5.57, 6.27, 1.43]^T$$

$$J(K_{LQR}) = 797, \quad J(K_{opt}) = 56$$

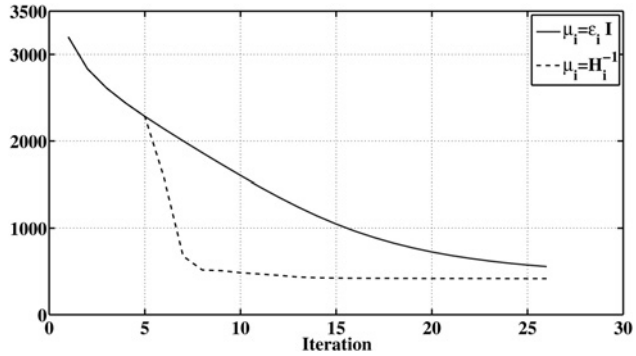

Fig. 2 *Evolution of the cost function J_x for $\mu_i = \varepsilon_i I$ (solid curve) and $\mu_i = H_i^{-1}$ (dashed curve)*

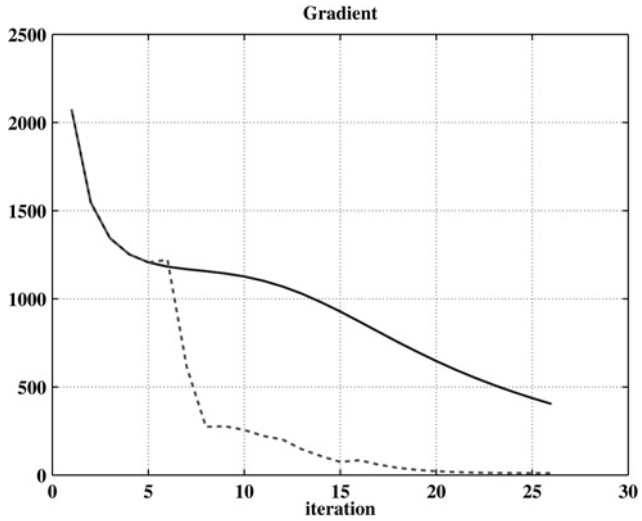

Fig. 3 *Evolution of the gradient norm $\|\nabla J_x(K_i)\|$ for $\mu_i = \varepsilon_i I$ (solid curve) and $\mu_i = H_i^{-1}$ (dashed curve)*

4.3 Non-quadratic cost functions

The formulation proposed in this paper is not restricted to quadratic cost functions. The method can be applied to any cost function as long as the cost function can be formulated as a smooth function of unknown parameters with the possibility of obtaining its partial derivatives with respect to those parameters. This is a strong feature, which is studied in this section by way of an example.

The previous section showed how the proposed controller can improve LQR performance. It is observed from the last two figures that improvement is obtained at the expense of some higher frequency dynamics. To control the variations of input signals, the cost function can be modified as

$$J(K) = \int_0^\infty [u^2(x, K) + \delta \dot{u}^2(x, K) + x^T Q x] dt \qquad (46)$$

where $\dot{u}(x, K)$ is the time derivative of the control signal. This signal can be obtained from

$$\dot{u}(x, K) = \frac{\partial u(x, K)}{\partial x}\dot{x} = \frac{\partial u(x, K)}{\partial x}F(x, K) \qquad (47)$$

where $F(x,K)$ is introduced in (45). The partial derivative of this term with respect to K can thus be computed.

State variables and the control input of the closed-loop control systems using the designed controller subject to a new cost function (with $\delta = 0.3$) are shown in (Fig. 6). The results show that the control signal is smoother and generates smoother states as compared with the previous case.

4.4 Optimising the linearising transformation

The feedback linearising transformation $z = \phi(x)$ is not unique. For the example discussed in this section, $h(x)$ can be any smooth function of x_1. Our previous study was performed based on the simplest selection which is $h(x) = x_1$. The next straightforward candidate is

$$h(x) = x_1 + \sum_{i=1}^{N} \theta_i x_1^{i+1}$$

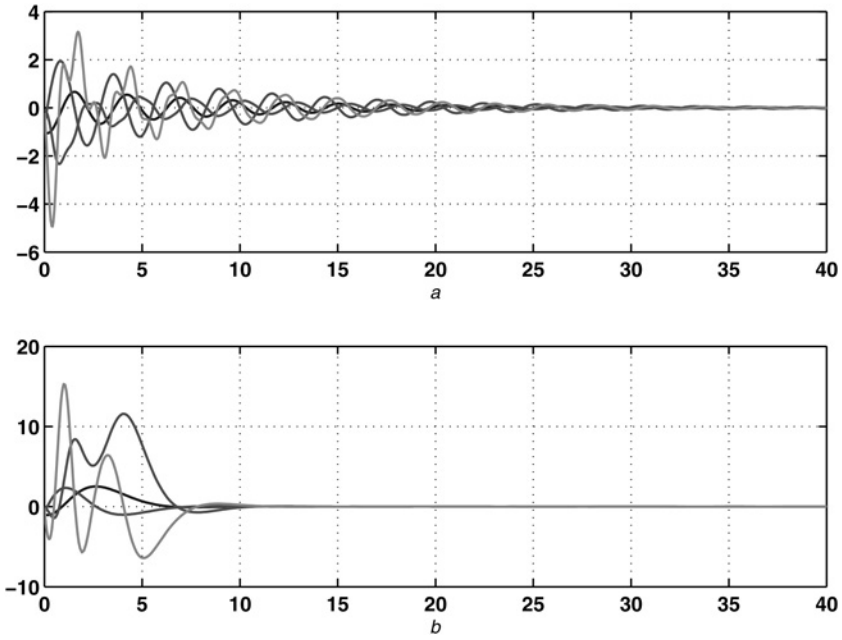

Fig. 4 *State variables*
a Proposed controller
b LQR method

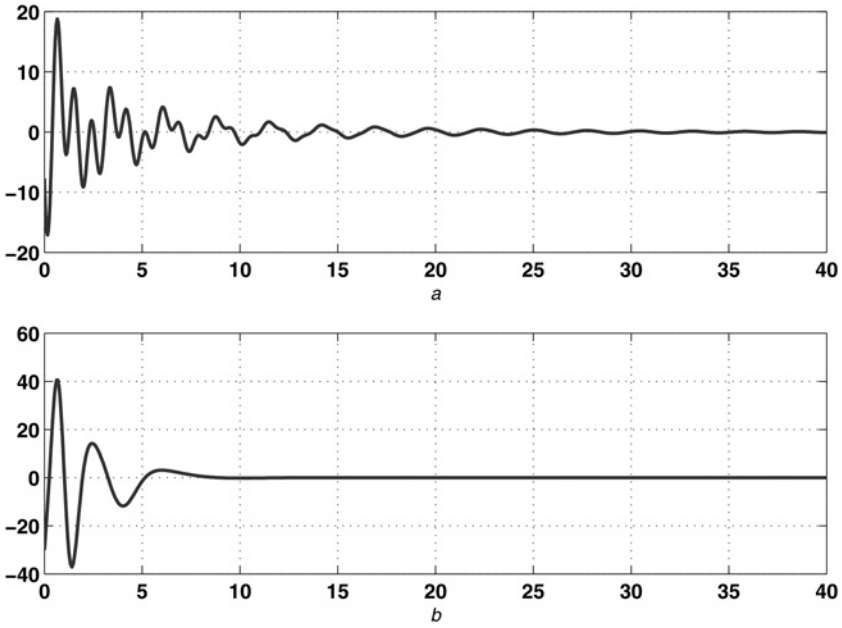

Fig. 5 *Control signal*
a Proposed controller
b LQR method

which remains a smooth and well-defined function (global diffeomorphism) under some conditions. It is possible to extend the method of this paper to obtain optimum values of θ_i's.

Define $\Theta = [\,\theta_1 \quad \theta_2 \quad \ldots \quad \theta_N\,]^{\mathrm{T}}$. Then the closed-loop equation is

$$\dot{x} = F(x, K, \Theta) \tag{48}$$

The task of the optimal controller is to minimise the cost function

$$J(K, \Theta) = \int_0^{T_f} \Gamma(x, K, \Theta)\mathrm{d}t \tag{49}$$

A similar formulation to what was presented before can be used to obtain the set of parameters K and θ. We avoid repeating the formulations due to similarity with the discussed case.

doi: 10.1049/iet-cta.2009.0242

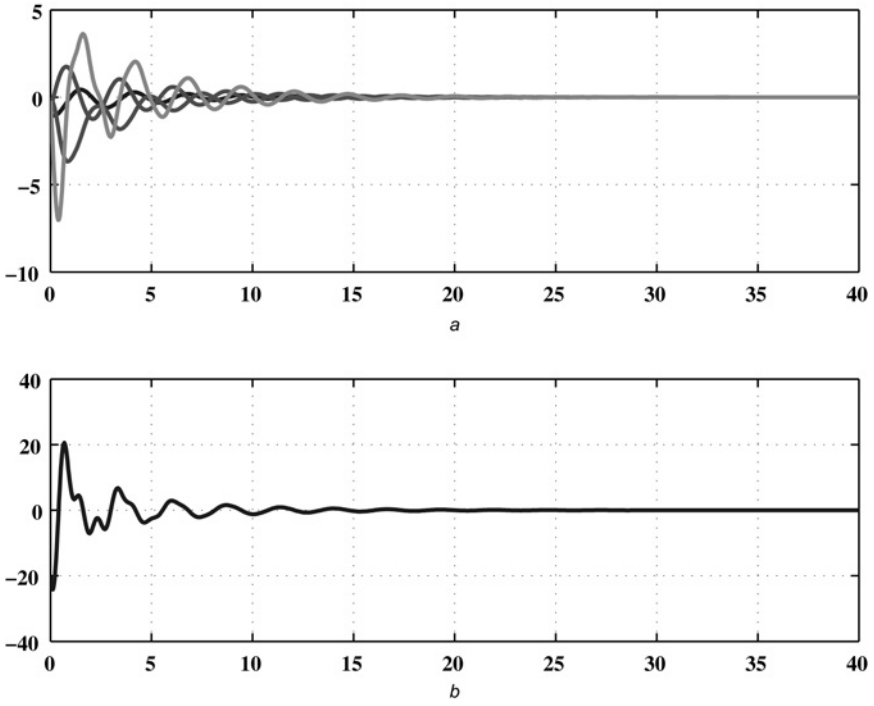

Fig. 6 *Controller design using the modified cost function introduced in equation*

a State variables
b Control signal

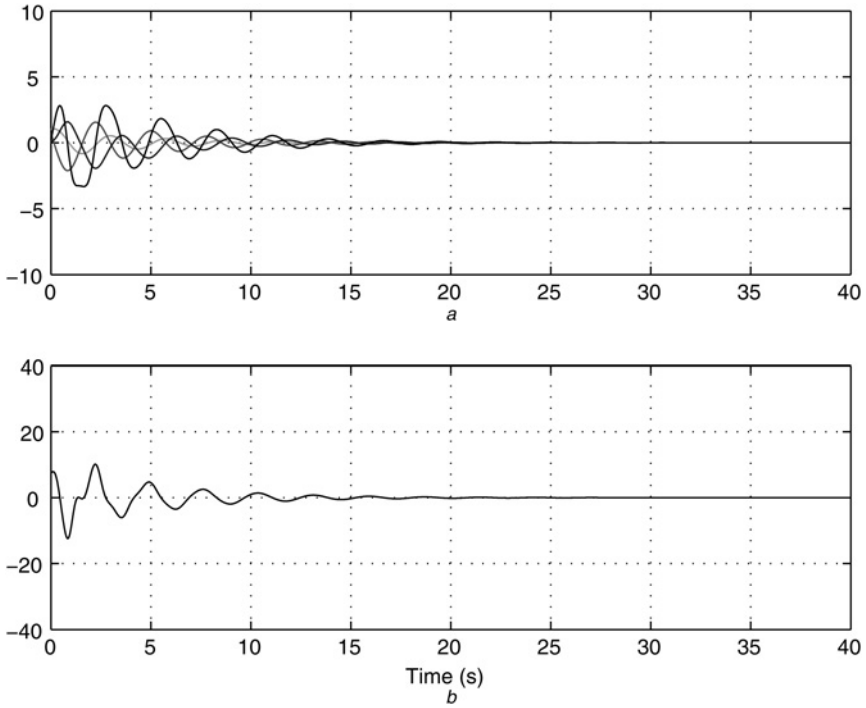

Fig. 7 *Controller design using the optimal feedback-linearisation transformation of the form* $h(x) = x_1 + \theta x_1^3$

a State variables
b Control signal

The method is applied to the case-study example with a selection of $h(x) = x_1 + \theta x_1^3$. The same quadratic cost function with the given system parameters are used for simulation, Fig. 7. The value of $\theta_{opt} = 0.018$ is obtained and the optimum controller gains are $K_{opt} = [2.05 \quad 5.53 \quad 6.01 \quad 1.49]^T$. The cost function is reduced to $J_{min} = 294$, which shows about 30% improvement as compared with the previous minimum value of 422.

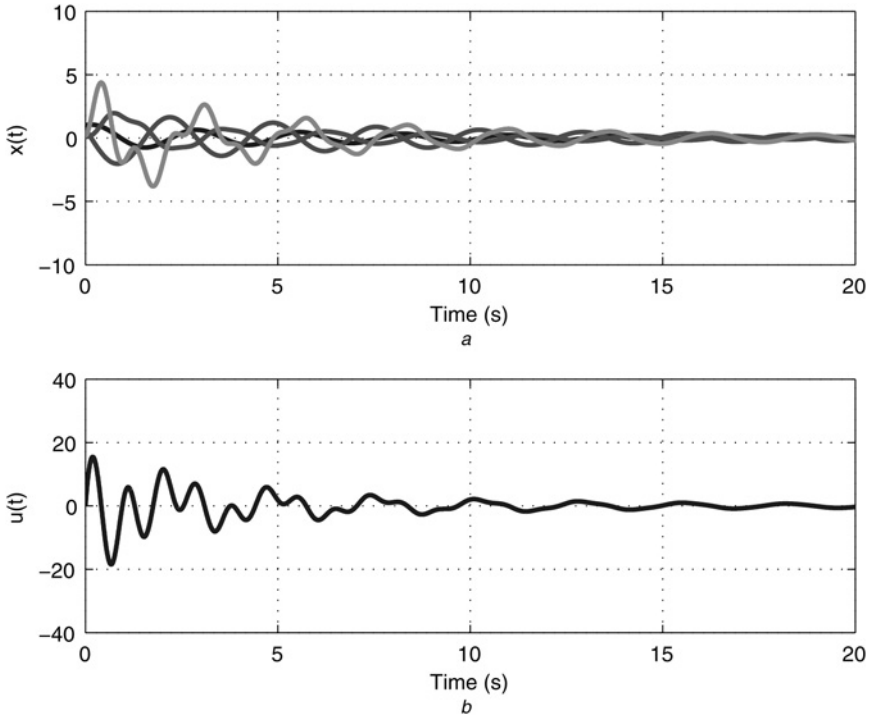

Fig. 8 *Controller design using the particle swarm optimisation method*
a State variables
b Control signal

Some results for different initial conditions are given below.

$$x_0 = [0.5, 1, -0.1, 0.1]^{\mathrm{T}}, \quad \theta_{\mathrm{opt}} = 0.0182, J(K_{\mathrm{LQR}}) = 1216$$
$$J(K_{\mathrm{opt}}) = 146, \quad J(K_{\mathrm{opt}}, \theta_{\mathrm{opt}}) = 124$$
$$x_0 = [0.7, -1, 0.2, 0.4]^{\mathrm{T}}, \quad \theta_{\mathrm{opt}} = 0.0209, \quad J_{\mathrm{LQR}} = 2029$$
$$J(K_{\mathrm{opt}}) = 374, \quad J(K_{\mathrm{opt}}, \theta_{\mathrm{opt}}) = 196$$
$$x_0 = [0.4, 0.4, 0.1, 0.1]^{\mathrm{T}}, \quad \theta_{\mathrm{opt}} = 0.0124, \quad J_{\mathrm{LQR}} = 797$$
$$J(K_{\mathrm{opt}}) = 56, \quad J(K_{\mathrm{opt}}, \theta_{\mathrm{opt}}) = 51$$

4.5 Comparison with the PSO algorithm

The particle swarm optimisation (PSO) algorithm is a population-based evolutionary algorithm that was developed from research on swarm such as fish schooling and bird flocking [16, 17]. It has become one of the most powerful methods for solving optimisation problems. The method is proved to be robust in solving problems featuring non-linearity and non-differentiability, multiple optima and high dimensionality. The advantages of PSO are its relative simplicity and stable convergence characteristics with good computational efficiency [18].

The proposed method of the paper in the context of the studied numerical example is compared with the PSO technique. The PSO obtained the optimal coefficient $K = [32.95 \quad 26.19 \quad 12.62 \quad 4.73]$ after 15 000 iterations, whereas the proposed algorithm reaches the same optimal value of cost function in less than 20 iterations. The evolvement of state variables and control signal for the PSO-based designed controller are shown in Fig. 8. Although the controller gains are very different from those obtained by the proposed method, the time responses of the two methods are very close.

Each iteration of the PSO requires an ODE of order n to be solved numerically while in each iteration of the proposed algorithm an ODE of order $(n+1)^2$ must be solved. This, however, does not demerit the proposed algorithm because solving an ODE in general is relatively a cheap process and solving an ODE with a large number of states can be done in parallel (thanks to the recent technology of multi-core process units). Moreover, the PSO algorithm, similar to any other evolutionary algorithm such as the genetic algorithm, requires huge storage and it also suffers from the same problem of trapping in local minima within a finite number of iterations.

5 Conclusion

The problem of designing a suboptimal state-feedback controller is addressed for a class of non-linear systems characterised by those that can be linearised using the input-state linearisation technique. The controller is in the form of a linear feedback on the state variables of the linearised system. The best coefficients that minimise a smooth cost function (of the non-linear system states and input) are obtained using a recursive algorithm that is based on gradient descent. A method for estimating the Hessian matrix to improve the algorithm convergence rate is also presented. It was shown that the proposed method can be used to optimise the linearising transformation as well. The simulations performed on a flexible-joint robot arm showed that the minimum value of the cost function can be reduced from 1600 (offered by the LQR technique) to about 300. The proposed technique can also handle non-quadratic cost functions, a feature that can be helpful in reaching different control goals, as confirmed in the paper. As compared with the evolutionary type algorithms such as genetic algorithm

doi: 10.1049/iet-cta.2009.0242

and particle swarm optimisation, the proposed algorithm is much faster and reaches the solution in a fewer number of iterations but it engages a solution of differential equations of higher order. As compared with the traditional method of Jacobian linearisation and designing an optimal controller for the linear system, the proposed method has the advantage of guaranteeing global stability for the whole region where the feedback linearising transformation is valid, and the method also enjoys independence from any particular operating point. As future work, we can suggest the design of a state observer for cases where the state variables are not available. Using a non-linear observer with tunable constant gains and augmenting its states to the actual dynamic system, one can extend the proposed method to the design of a state observer.

6 References

1 Krener, A.J.: 'On the equivalence of control systems and the linearisation of non-linear systems', *SIAM J. Control Optim.*, 1973, **11**, pp. 670–676
2 Jakubczyk, B., Respondek, W.: 'On linearisation of control systems', *Bull. Acad. Polonaise, Sci. Ser. Sci. Math.*, 1980, **28**, pp. 517–522
3 Isidori, A., Krener, A., Gori, A.J., Monaco, S.: 'Nonlinear decoupling via feedback: a differential geometric approach', *IEEE Trans. Automat. Control*, 1981, **26**, pp. 331–345
4 Hunt, L.R., Su, R., Meyer, G.: 'Global transformations of non-linear systems', *IEEE Trans. Automat. Control*, 1983, **28**, pp. 24–31
5 Isidori, A.: 'Nonlinear Control Systems' (Springer, Berlin, 1995)
6 Nijmeijer, H., Van der Schaft, A.J.: 'Nonlinear Dynamical Control Systems' (Springer, Berlin, 1990)
7 Charlet, B., Levine, J., Marino, R.: 'On dynamic feedback linearisation', *Syst. Control Lett.*, 1989, **13**, pp. 143–151
8 Charlet, B., Levine, J., Marino, R.: 'Sufficient conditions for dynamic state feedback linearisation', *SIAM J. Control Optim.*, 1991, **29**, pp. 38–57
9 Sluis, W.M.: 'A necessary condition for dynamic feedback linearisation', *Syst. Control Letters*, 1993, **21**, pp. 277–283
10 Marino, R.: 'On the largest feedback linearisable subsystem', *Syst. Control Lett.*, 1986, **6**, pp. 345–351
11 Respondek, W.: 'Partial linearisation, decompositions and fiber linear systems', in Byrnes, C.I., Lindquist, A. (Eds.): 'Theory and Applications of Nonlinear Control' (North-Holland, Amsterdam, The Netherlands, 1986), pp. 137–154
12 Sun, Z., Ge, S.: 'Nonregular feedback linearisation: a nonsmooth approach', *IEEE Trans. Automat. Control*, 2003, **48**, pp. 1772–1776
13 Khalil, H.: 'Nonlinear Systems' (Prentice-Hall, Upper Saddle River, NJ, 2002, 3rd edn.)
14 Slotine, J.J.E., Li, W.: 'Applied Nonlinear Control' (Prentice Hall, 1991)
15 Nocedal, J., Wright, S.J.: 'Numerical Optimization' (Springer-Verlag, 1999, 1st edn.)
16 Kennedy, J., Eberhart, R.: 'Particle swarm optimisation'. Proc. IEEE International Conf. on neural networks, 1995, vol. 4, pp. 1942–1947
17 Eberhart, R., Shi, Y.: 'Particle swarm optimisation: developments, applications and resources'. Proc. IEEE Int. Conf. on evolutionary computation, 2001, pp. 1–86
18 Zamani, M., Karimi-Ghartemani, M., Sadati, N., Parniani, M.: 'Design of a fractional order PID controller for an AVR using particle swarm optimisation', *J. Control Eng. Pract.*, 2009, **17**, (12), pp. 1380–1387