Robotica (2010) volume 28, pp. 959–973. © Cambridge University Press 2010
doi:10.1017/S0263574709990798

Determining maximum load carrying capacity of planar flexible-link robot: closed-loop approach

M. H. Korayem*, R. Haghighi, A. H. Korayem, A. Nikoobin and A. Alamdari

Robotic Research Laboratory, College of Mechanical Engineering, Iran University of Science and Technology, Tehran, Iran

(Received in Final Form: December 3, 2009. First published online: January 7, 2010)

SUMMARY

Maximum load carrying capacity (MLCC) of flexible robot manipulators is computed based on closed-loop approach. In open-loop approach, controller is not considered, so the end effector deviation from the predefined path is significant and the accuracy constraint restrains the maximum payload before actuators go into saturation mode. In order to improve the MLCC, a method based on closed-loop strategy is presented. Since in this case the accuracy is improved the actuators constraint is not a major concern and full power of actuators can be used. Since controller can play an important role in improving the maximum payload, a sliding mode based partial feedback linearization controller is designed. Furthermore, a fuzzy variable layer is used in sliding mode design to boost the performance of the controller. However, the control strategy required measurements of elastic variables velocity that are not conveniently measurable. So a nonlinear observer is designed to estimate these variables. Stability analysis of the proposed controller and state observer are performed on the basis of Lyapunov's direct method. In order to verify the effectiveness of the presented method, simulation is done for a two-link flexible manipulator. The obtained maximum payload in open-loop and closed-loop cases is compared and the superiority of the method is illustrated and the results are discussed.

KEYWORDS: Flexible manipulator; Sliding mode; Observer; Feedback linearization.

1. Introduction

In traditional rigid manipulators, the dynamic load carrying capacity (DLCC) is usually defined as the maximum load which a manipulator can repeatedly lift and carry on the fully extended configuration while the dynamics of both the load and manipulator must be taken into account.[1] Another definition of maximum payload is the maximum value of load which a robot manipulator is able to carry on a desired trajectory which is based on the consideration of inertia effects on this desired path.[2] For rigid manipulators, maximum load on a given trajectory is primarily constrained by the joint actuator torque and its velocity characteristic. However, for flexible manipulators another constraint, maximum allowable deflection, must be considered. Korayem and Basu,[3] presented an algorithm for computing the DLCC of elastic manipulators by imposing the accuracy constraint of the end effector in addition to torque constraint of actuators.

On the other hand, the maximum load carrying capacity (MLCC) along the given path can be determined in both open-loop and closed-loop cases. In open-loop case, several algorithms are proposed for finding the MLCC of multiple cooperating robotic manipulators,[4] rigid mobile manipulators,[5] flexible joint mobile manipulators,[6] flexible-link manipulators,[7] cable-suspended parallel manipulators,[8] and redundant manipulators.[9] In the closed-loop case, controller type and its parameters have a significant effect on increasing the maximum payload. A closed-loop approach has been employed to determine the DLCC of a flexible joint manipulator by considering the feedback linearization controller to track the predefined path.[10] Another work based on sliding mode technique has been done for flexible joint manipulators.[11]

In flexible-link manipulators, complexities such as nonminimum-phase property of the tip transfer function, the existence of unstructured uncertainties, and system nonlinearities makes it difficult to accurately position the tip of the flexible manipulators and it caused designing controllers for flexible manipulators to be a very challenging task. One method which is excessively used in control of nonlinear systems is feedback linearization.[12] However for flexible-link robots due to highly nonlinear coupled dynamics and existence of passive degrees-of-freedom only partial feedback linearization (PFL) is suitable. For flexible-link robots, the portion of the dynamics corresponding to the active degrees of freedom can be linearized by the nonlinear feedback. The remaining portion of the dynamics after such PFL is nonlinear and represents internal dynamics.[13] A major drawback in this method which can not be used solely as a controller is lack of robustness with respect to uncertainties.

In the design of SMC, it is assumed that the control can be switched from one structure to another infinitely fast. However, because of the switching delay computation and the limitation of physical actuators, which cannot handle the switching of control signal at an infinite rate, it is practically impossible to achieve high-speed switching control. As a result of this imperfect control switching between structures, the system trajectory appears to chatter instead of sliding along the switching surface. Chatter involves high frequency control switching and may lead to

* Corresponding author. E-mail: hkorayem@iust.ac.ir

excitation of previously neglected high frequency system dynamics. Smoothing techniques such as boundary layer normalization, replacement of discontinuous control term by a fuzzy system[14,15] have been employed. The smoothing of control discontinuity inside the boundary layer essentially assigns a low pass filter structure to the local dynamics of the variable s, thus eliminates chattering. Through this approach the transient performance of the closed-loop system is maintained, however controller can only guarantee final tracking accuracy to within the ε-vicinity of the demand, where ε is the radius of the boundary layer. A compromise must therefore be sought between desired tracking accuracy and controller bandwidth. To reach a better compromise between small chattering and good tracking precision in the presence of parameter uncertainties, various compensation strategies have been proposed. For example, integral sliding mode control, sliding mode control with time-varying boundary layers, adaptively selecting the switching gain of the sliding mode controller.[16–18] The sliding mode control with the variable boundary layer needs a strategy to adjust the thickness of the boundary automatically and research on the strategies has been conducted widely. For tuning the boundary layer a fuzzy system is used which adopts an algebraic distance s to the sliding surface as input and the thickness of the boundary layer as output such that better tracking achieved.

An accurate knowledge of the arm state variables is required by many advanced control techniques for flexible multilink robots. It can be conveniently achieved using a state observer. Some works have been done using linear observers derived for a linearized model of the arm.[19] Other papers propose the use of nonlinear state observers to obtain the values of immeasurable state variables.[20,21] In flexible manipulators it is possible to measure joint positions, velocities, and flexible modes of manipulators using shaft encoders, tachometers, and strain gauges, respectively. However, measuring the flexural generalized velocities cannot be easily or accurately accomplished. Thus, a state observer is desirable in these circumstances. In order to decrease in computational effort a reduced order observer for estimating only flexible variables can be very helpful. The proposed observer requires positions and velocities of joints as well as flexible modes, and it estimates the rates of change of flexible modes. In order to show the effectiveness of the proposed closed-loop algorithm the simulation is done for both open-loop and closed-loop cases.

In this paper, the general dynamic equations of flexible-link manipulator are derived. Then, by using PFL, a controller is designed for partially linearized model of flexible manipulator based on fuzzy variable boundary layer sliding mode approach. An algorithm is proposed to compute maximum allowable load by considering the limiting factors. Finally, simulation study is conducted for two-link flexible planar manipulator.

2. Dynamics of Flexible-Link Manipulator

An n-link flexible manipulator moving in a vertical plane, with rotary joints subject to only bending deformation in the plane of motion is shown in Fig. 1. In order to derive a finite-dimensional ordinary differential equation an approximation approach using assumed mode methods is taken into account.

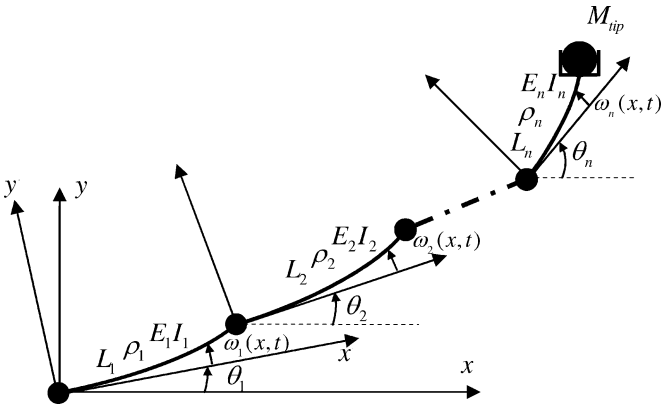

Fig. 1. Flexible manipulator (L_i is length of the ith link, ρ_i is mass per length of the ith link, $E_i I_i$ is flexural rigidity of the ith link, ω_i is deformation of the ith link, and θ_i is joint angular position of the ith link).

The dynamics of any multilink flexible-link robot can be represented by:

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

where $q(t) = [q_r^T, q_f^T]^T$, in which q_r is the vector of generalized joint coordinates and q_f is the vector of flexible modes. $M(q)$ represents the inertia matrix, $N(q, \dot{q})$ is a $n \times 1$ vector of centripetal and Coriolis velocity terms and τ denotes the vector of generalized control forces (torque) applied at each joint.

The flexible manipulator dynamics are partitioned into rigid and flexible degrees-of-freedom as

$$\begin{cases} M_{rr}\ddot{q}_r + M_{rf}\ddot{q}_f + N_r = u & \text{I} \\ M_{fr}\ddot{q}_r + M_{ff}\ddot{q}_f + N_f = 0 & \text{II} \end{cases}, \tag{2}$$

where M_{rr}, M_{rf}, M_{fr}, and M_{ff} are the blocks of inertia matrix, M, N_r, and N_f are the blocks of Coriolis and centrifugal matrix N and u includes the control torques applied to each joint.

The following properties are known to be verified, by the Lagrangian structure definite matrices.

3. Controller Design of Flexible-Link Manipulator

The controller design of flexible-link manipulator is divided into two steps. First by applying PFL, the dynamic of flexible link is divided into two parts: partially linearized model and internal model. Second, the state trajectory of a system to the origin in the error phase hyperplane by using the sliding mode approach force during two distinct phases: reaching phase and sliding phase.

3.1. Partial feedback linearization

Using Eq. (2-II), $\ddot{q}_f$ can be expressed as below

$$\ddot{q}_f = -M_{ff}^{-1}[M_{fr}\ddot{q}_r + N_f]. \tag{3}$$

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

Substituting for $\ddot{q}_f$ from Eq. (3) in Eq. (2-I) gives

$$\left[M_{rr} - M_{rf}M_{ff}^{-1}M_{fr}\right]\ddot{q}_r + N_r - M_{rf}M_{ff}^{-1}N_f = u. \quad (4)$$

Equation (4) is similar in form to rigid manipulator modeling with the equivalent symmetric and positive definite mass matrix $M_{rr} - M_{rf}M_{ff}^{-1}M_{fr}$ based on property II.[22] The zero dynamic is defined for Eq. (2) by putting q_r and its derivatives equal to zero.

$$M_{ff}\ddot{q}_f + N_f = 0, \quad (5)$$

where N_f simplified to Kq_f, and K is the stiffness matrix and can be written as

$$K = diag\left\{\omega_{11}^2, \omega_{12}^2, \ldots, \omega_{ij}^2, \ldots, \omega_{nm}^2\right\}, \quad (6)$$

where ω_{ij} is the jth natural frequency of the ith link. Equation (5) can be written as

$$M_{ff}\ddot{q}_f + Kq_f = 0. \quad (7)$$

Since M_{ff} and K are the positive definite symmetric matrices, the equilibrium point $[q_f, \dot{q}_f]$ of Eq. (7) is stable in the sense of Lyapunov but not asymptotically stable.

3.2. Sliding mode design

The Lyapunov sliding condition forces system states to reach a hyperplane and keeps them sliding on this hyperplane. Essentially, a SMC design is composed of two phases: hyperplane design and controller design. There are various methods for designing hyperplane, however a method proposed by Slotine is used.[23] In this method the sliding surface is defined as

$$s = (\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f) + \lambda(\tilde{q}_r + \lambda'\tilde{q}_f), \quad (8)$$

where $\tilde{q}_r = q_r - q_r^{ref}$ and $\tilde{q}_f = q_f - q_f^{ref}$. q_r^{ref} is the desired trajectory of joints and $q_r^{ref} = 0$, because the desired value for flexible variables is zero. Also λ and λ' are positive constants.

To determine the control law, the derivative of the sliding surface must be determined as follow.

$$\dot{s} = (\ddot{\tilde{q}}_r + \lambda'\ddot{\tilde{q}}_f) + \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f). \quad (9)$$

Treating the term $\lambda'\ddot{q}_f^{ref}$ as disturbance, Eq. (9) is rearranged as below

$$\dot{s} = \ddot{q}_r - \ddot{q}_r^{ref} + \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f). \quad (10)$$

Since sliding condition is defined by

$$\dot{s} \leq -\Lambda\ sign(s), \quad (11)$$

where Λ is positive definite matrix. Equation (10) in order to satisfy the sliding condition must be written as

$$\ddot{q}_r - \ddot{q}_r^{ref} + \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f) = -\Lambda\ sign(s). \quad (12)$$

By substituting $\ddot{q}_r$ from Eq. (4), Eq. (12) becomes

$$\left[M_{rr} - M_{rf}M_{ff}^{-1}M_{fr}\right]^{-1}\left[u - N_r + M_{rf}M_{ff}^{-1}N_f\right] - \ddot{q}_r^{ref} + \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f) = -\Lambda\ sign(s). \quad (13)$$

By extracting u from Eq. (13), control law is defined as

$$u = M\left[\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f) - \Lambda\ sign(s)\right] + N, \quad (14)$$

where $M = [M_{rr} - M_{rf}M_{ff}^{-1}M_{fr}]$ and $N = N_r - M_{rf}M_{ff}^{-1}N_f$. By defining the $\hat{M}$ and $\hat{N}$ which are the nominal value of M and N respectively, the control law can be rewritten as follow

$$u = \hat{M}\left[\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f) - \Lambda\ sign(s)\right] + \hat{N}. \quad (15)$$

3.3. Stability analysis of SMC

A simple stability analysis based on Lyapunov direct method is carried out. Define the Lyapunov function candidate

$$V = \frac{1}{2}s^2. \quad (16)$$

Differentiating Eq. (16) and using Eqs. (4), (11), and (15), one can write

$$\begin{aligned}\dot{V} &= s\dot{s} \\ &= s\left\{\mathrm{M}^{-1}\left[\hat{\mathrm{M}}\left[\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f) - \Lambda\ sign(s)\right]\right.\right. \\ &\quad \left.\left. + \hat{\mathrm{N}} - \mathrm{N}\right] - \ddot{q}_r^{ref} + \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f)\right\}.\end{aligned} \quad (17)$$

By using simplification Eq. (17) becomes

$$\begin{aligned}\dot{V} = s\left\{\mathrm{M}^{-1}\hat{\mathrm{M}}\left[\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f) - \Lambda\ sign(s)\right]\right. \\ \left. + \mathrm{M}^{-1}\Delta\mathrm{N} - \ddot{q}_r^{ref} + \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f)\right\}.\end{aligned} \quad (18)$$

For stability, $\dot{V}$ must be negative. Since $\dot{V} = s\dot{s}$ another condition which assures the stability of system can be defined as $s\dot{s} \leq -\eta\,|s|$ or $\dot{s} \leq -\eta\ sign(s)$ which is called sliding condition. By applying sliding condition we have

$$\begin{aligned}(\mathrm{M}^{-1}\hat{\mathrm{M}} - \mathrm{I})\left(\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f)\right) - \mathrm{M}^{-1}\hat{\mathrm{M}}.\Lambda.sign(s) \\ + \mathrm{M}^{-1}\Delta\mathrm{N} \leq -\eta.\ sign(s).\end{aligned} \quad (19)$$

By multiplication of both sides of Eq. (19) by $\hat{\mathrm{M}}^{-1}\mathrm{M}$, one gets

$$\begin{aligned}(\mathrm{I} - \hat{\mathrm{M}}^{-1}\mathrm{M})\left(\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f)\right) - \Lambda.\ sign(s) \\ + \hat{\mathrm{M}}^{-1}\Delta\mathrm{N} \leq -\hat{\mathrm{M}}^{-1}\mathrm{M}.\eta.\ sign(s).\end{aligned} \quad (20)$$

So the condition which guarantees the stability can be expressed as follow

$$\begin{aligned}\Lambda > \left|(\mathrm{I} - \hat{\mathrm{M}}^{-1}\mathrm{M})\left(\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f)\right)\right. \\ \left. + \hat{\mathrm{M}}^{-1}\Delta\mathrm{N}\right| + \hat{\mathrm{M}}^{-1}\mathrm{M}\eta.\end{aligned} \quad (21)$$

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

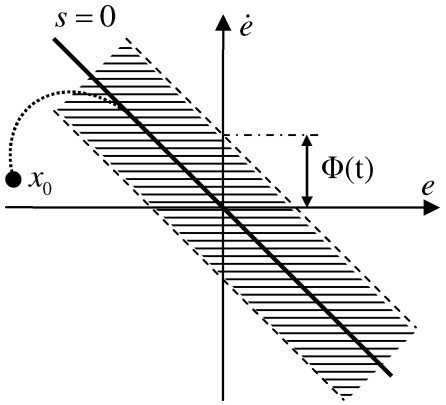

Fig. 2. Variable boundary layer (Φ(t) is boundary layer width, s is sliding surface and e is error).

Since M is unknown one can define the following known bounds

$$\mathrm{M}_{\min} \leq \mathrm{M} \leq \mathrm{M}_{\max}. \tag{22}$$

Since M acts multiplicatively in the dynamic of manipulator, it is reasonable to choose the estimate $\hat{\mathrm{M}}$ of M as the geometric mean of the above bounds,[23]

$$\hat{\mathrm{M}} = (\mathrm{M}_{\min}\mathrm{M}_{\max})^{1/2}. \tag{23}$$

Therefore bounds for $\hat{\mathrm{M}}^{-1}\mathrm{M}$ can be defined as follow

$$\Psi^{-1} \leq \hat{\mathrm{M}}^{-1}\mathrm{M} \leq \Psi, \tag{24}$$

where

$$\Psi = \left(\frac{\mathrm{M}_{\max}}{\mathrm{M}_{\min}}\right)^{1/2}. \tag{25}$$

Equation (21) can be rewritten in terms of Ψ

$$\Lambda > \left|(\mathrm{I} - \Psi)\left(\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f)\right) + \hat{\mathrm{M}}^{-1}\Delta\mathrm{N}\right| + \Psi\eta. \tag{26}$$

3.4. Variable boundary layer

By introducing a Boundary Layer (Φ) from both sides of the sliding surface $s = 0$, as shown in Fig. 2., Eq. (15) is written as below

$$u = \hat{M}\left[\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f) - \Lambda\, sat\left(\frac{s}{\Phi}\right)\right] + \hat{N}, \tag{27}$$

where "*sat*" is saturation function.

If we rewrite $\dot{s}$ based on "*sat*" function, we have

$$\dot{s} = (\mathrm{M}^{-1}\hat{\mathrm{M}} - \mathrm{I})\left(\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f)\right) - \mathrm{M}^{-1}\hat{\mathrm{M}}.\Lambda.sat\left(\frac{s}{\Phi}\right) + \mathrm{M}^{-1}\Delta\mathrm{N}. \tag{28}$$

By considering the system trajectories inside the boundary layer

$$\dot{s} = (\mathrm{M}^{-1}\hat{\mathrm{M}} - \mathrm{I})\left(\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f)\right) - \mathrm{M}^{-1}\hat{\mathrm{M}}.\Lambda.\left(\frac{s}{\Phi}\right) + \mathrm{M}^{-1}\Delta\mathrm{N}, \tag{29}$$

and Eq. (29) can be rewritten as

$$\dot{s} + \frac{\mathrm{M}^{-1}\hat{\mathrm{M}}.\Lambda}{\Phi}s = (\mathrm{M}^{-1}\hat{\mathrm{M}} - \mathrm{I})\left(\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f)\right) + \mathrm{M}^{-1}\Delta\mathrm{N}. \tag{30}$$

In fact Eq. (30) shows that the smoothing of control discontinuity inside boundary layer essentially assigns a low pass filter structure to the local dynamics of the variable s, thus eliminate chattering. Furthermore the sliding condition is redefined as below

$$\dot{s} \leq (\dot{\Phi} - \Lambda)\, sign(s), \tag{31}$$

because in the presence of boundary layer we need to guarantee that the distance to the boundary layer always decreases.

System robustness is a function of the boundary layer, in other words thinner boundary layer gives more robust control, but larger chattering. Therefore, there is a trade off between the chattering and the robustness of the system for the width of the boundary layer. In order to improve the robustness of the system while at the same time reducing the chattering, Φ can be tuned according to tracking errors through the fuzzy logic systems. The fuzzy system adopts the distance of trajectories to the sliding surface ($|s|$) as input and the thickness of the boundary layer as output. The membership used in fuzzy systems is shown in Fig. 3.

Fuzzy system is constructed based on "singleton" fuzzifier, "Mamdani" fuzzy inference system, "centroid" defuzzification and the rule-base including 4 IF-THEN rules as below

$\mathrm{R}^{(1)}$: IF $|s|$ is "Z" THEN Φ is "Z"
$\mathrm{R}^{(2)}$: IF $|s|$ is "S" THEN Φ is "S"
$\mathrm{R}^{(3)}$: IF $|s|$ is "M" THEN Φ is "M"
$\mathrm{R}^{(4)}$: IF $|s|$ is "B" THEN Φ is "B"

By applying the fuzzy variable boundary layer, the control law can be expressed as

$$u = \hat{M}\left[\ddot{q}_r^{ref} - \lambda(\dot{\tilde{q}}_r + \lambda'\dot{\tilde{q}}_f) - (\Lambda - \dot{\Phi}(t))\, sat\left(\frac{s}{\Phi(t)}\right)\right] + \hat{N}. \tag{32}$$

4. State Observer Design

By extracting $\ddot{q}_r$ from Eq. (2-I), and substituting in Eq. (2-II), we have

$$M_{fr}M_{rr}^{-1}[u - M_{rf}\ddot{q}_f - N_r] + M_{ff}\ddot{q}_f + N_f = 0, \tag{33}$$

and it can be rearranged to

$$\left[M_{ff} - M_{fr}\, M_{rr}^{-1}M_{rf}\right]\ddot{q}_f + N_f - M_{fr}\, M_{rr}^{-1}N_r + M_{fr}M_{rr}^{-1}u = 0. \tag{34}$$

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

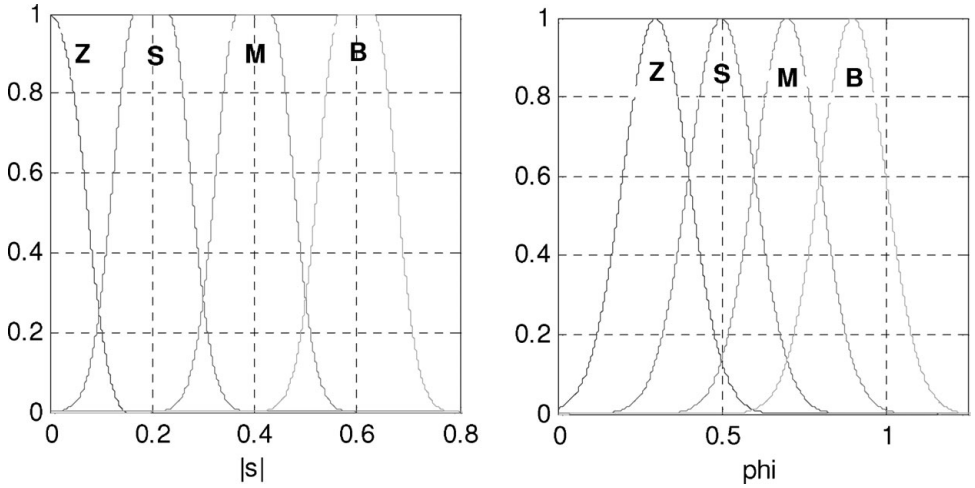

Fig. 3. The membership function for $|s|$ and Φ (Z is Zero, S is small, M is medium, B is big).

Equation (34) can be expressed in state space form

$$\begin{cases} \dot{x}_{f_1} = x_{f_2} \\ \dot{x}_{f_2} = \left[M_{ff} - M_{fr}M_{rr}^{-1}M_{rf}\right]^{-1}\left[-N_f \right. \\ \qquad \left. + M_{fr}M_{rr}^{-1}N_r - M_{fr}M_{rr}^{-1}u\right] \end{cases}, \tag{35}$$

where $x_{f_1} = q_f$ and $x_{f_2} = \dot{q}_f$. Using sliding mode observer technique, the dynamic of observer is written as

$$\begin{cases} \dot{\hat{x}}_{f_1} = \hat{x}_{f_2} + k_{11}\tilde{x}_{f_1} + k_{12}sign(\tilde{x}_{f_1}) \\ \dot{\hat{x}}_{f_2} = \hat{f}(x_r, x_{f_1}, \hat{x}_{f_2}) + k_{21}\tilde{x}_{f_1} + k_{22}sign(\tilde{x}_{f_1}) \end{cases}, \tag{36}$$

where $\hat{f}(x_r, x_{f_1}, \hat{x}_{f_2}) = [M_{ff} - M_{fr}M_{rr}^{-1}M_{rf}]^{-1}[-N_f + M_{fr}M_{rr}^{-1}N_r - M_{fr}M_{rr}^{-1}u]$ and k_{ij} are positive gains. $\tilde{x}_{f_1}$ is the estimation error and equal to $x_{f_1} - \hat{x}_{f_1}$. The dynamic of error is achieved by subtracting Eq. (35) by Eq. (36)

$$\begin{cases} \dot{\tilde{x}}_{f_1} = \tilde{x}_{f_2} - k_{11}\tilde{x}_{f_1} - k_{12}sign(\tilde{x}_{f_1}) \\ \dot{\tilde{x}}_{f_2} = \tilde{f} - k_{21}\tilde{x}_{f_1} - k_{22}sign(\tilde{x}_{f_1}) \end{cases}, \tag{37}$$

where can be written in a simple form of

$$\dot{e} = \tilde{f}^* - K_e e - K_s sign(e), \tag{38}$$

where $\tilde{f}^* = \begin{bmatrix} \tilde{x}_{f_2} \\ \tilde{f} \end{bmatrix}$ and $e = \begin{bmatrix} \tilde{x}_{f_1} \\ \tilde{x}_{f_2} \end{bmatrix}$. Using Taylor expansion around $e = 0$, Eq. (38) can be given as

$$\dot{e} = Ae + O(\mathrm{e}^2) - K_s sign(e), \tag{39}$$

where

$$A = \left(\frac{\partial \tilde{f}^*}{\partial e} - K_e\right)_{e=0} = \begin{bmatrix} 0 & 1 \\ \left.\frac{\partial \tilde{f}}{\partial \tilde{x}_{f_1}}\right|_{\tilde{x}_{f_1}=0} & \left.\frac{\partial \tilde{f}}{\partial \tilde{x}_{f_2}}\right|_{\tilde{x}_{f_2}=0} \end{bmatrix} + \begin{bmatrix} -k_{11} & 0 \\ -k_{21} & 0 \end{bmatrix}. \tag{40}$$

The eigenvalues of A can be specifically placed by properly choosing of K_e. If matrix A has negative eigenvalues then the error will converge to zero. From matrix algebra we know that a square matrix is negative definite if determinants of all principal minors have the following pattern:

$$|D_1| < 0, |D_2| > 0, |D_3| < 0, \ldots, \tag{41}$$

where D_i is the ith principle minor. So, by applying the above conditions we have

$$-k_{11}\left.\frac{\partial \tilde{f}}{\partial \tilde{x}_{f_2}}\right|_{\tilde{x}_{f_2}=0} + k_{21} - \left.\frac{\partial \tilde{f}}{\partial \tilde{x}_{f_1}}\right|_{\tilde{x}_{f_1}=0} > 0. \tag{42}$$

If k_{21} is chosen big enough, the above condition is satisfied. Now for stability analysis of the proposed state observer the following Lyapunov candidate is considered

$$V(e) = \mathrm{e}^T P e, \tag{43}$$

where P is a positive definite matrix. The time derivative of this Lyapunov function can be expressed as

$$\dot{V}(e) = \mathrm{e}^T(A^T P + PA)e + 2\mathrm{e}^T P(O(\mathrm{e}^2) - K_s sign(e)). \tag{44}$$

By replacement, $A^T P + PA$ by a positive definite matrix Q such that

$$A^T P + PA = -Q. \tag{45}$$

Equation (44) can be simplified to be

$$\dot{V}(e) = -\mathrm{e}^T Qe + 2\mathrm{e}^T P(O(\mathrm{e}^2) - K_s\ sign(e)). \tag{46}$$

To get a more simplified form, $sign(e)$ is replaced by $\frac{e}{|e|}$, moreover it's assumed that $O(\mathrm{e}^2)$ is a Lypschitz function with a constant γ, so we have

$$\dot{V}(e) = -\mathrm{e}^T Qe + 2\gamma\mathrm{e}^T Pe - 2PK_s\,|e|\,. \tag{47}$$

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

From Eq. (47), we can have the following inequality

$$\dot{V}(e) \leq -\lambda_{\min}(Q)\,\|e\|^2 + 2\gamma\lambda_{\max}(P)\,\|e\|^2 - 2\lambda_{\min}(PK_s)\,|e|\,, \tag{48}$$

where $\lambda_{\min}(Q)$ and $\lambda_{\max}(P)$ are the minimum and maximum eigenvalues of Q and P respectively. By proper choice of K_e and K_s the negativeness of $\dot{V}(e)$ can be guaranteed.

5. Maximum Load Carrying Capacity

MLCC can be obtained in either open-loop or closed-loop case. In open loop, the controller is not considered and only dynamic equation is used. In closed-loop case, MLCC is obtained while both dynamic equation and controller are considered. The actuator torque constraint is formulated on the basis of typical torque–speed characteristics of DC motors.

$$\begin{cases} \tau_U = K_1 - K_2\dot{q}_r \\ \tau_L = -K_1 - K_2\dot{q}_r \end{cases}, \tag{49}$$

where τ_U and τ_L are the upper bound and the lower bound of actuator constraint, respectively. The coefficients K_i are defined as

$$\begin{cases} K_1 = T_s \\ K_2 = \dfrac{T_s}{\omega_{nl}} \end{cases}, \tag{50}$$

where T_s is the stall torque and ω_{nl} is the maximum no-load speed of the motor.

5.1. Determining the MLCC in open-loop case

For computing the MLCC in open-loop condition, the following steps must be done:

1. Determining the actuator path in which the arms are in fully extended configurations.
2. Finding $q_r, \dot{q}_r, \ddot{q}_r$ by solving inverse dynamic for the same rigid manipulator.
3. Determining $q_f, \dot{q}_f, \ddot{q}_f$ from Eq. (2-II).
4. Computing of the actuators torque (τ_{nl}) and end effector path for no load manipulator.
5. Choosing an initial value for $m_{\max}$.
6. Putting $m_p = m_{\max}$ and computing the actuators torque (τ_l) and end effector path.
7. Computing the actuators bounds based on Eq. (49) and Eq. (50).
8. Determining the load coefficient C_a based on actuator constraints:[10]

$$C_a = \min\Big(\min\big(C_a^{\text{first joint}}(1:n)\big), \min\big(C_a^{\text{second joint}}(1:n)\big)\Big). \tag{51}$$

9. Determining the load coefficient C_p based on accuracy constraints:

$$C_P(k) = \frac{R_P - \Delta_e(k)}{\max(\Delta_e(k)) - \max(\Delta_n(k))}, \tag{52}$$

where $\Delta_e(k)$ is the error of end effector in the present of load and $\Delta_n(k)$ is the error of end effector without load (Fig. 3).

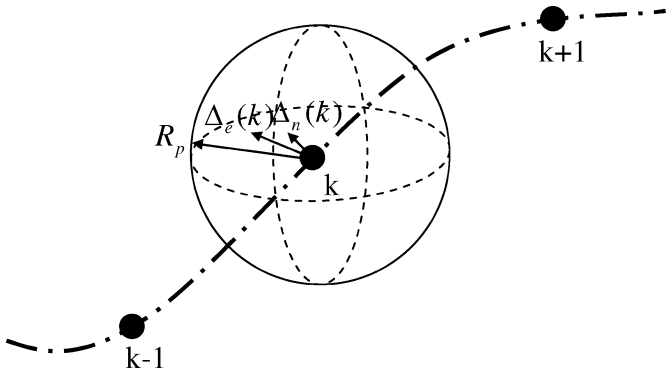

Fig. 4. Boundary of end effector deflection.

10. Determining the load coefficient C

$$C = \min(C_P, C_a). \tag{53}$$

11. If $|C^{i+1} - C^i| \leq$ error then $m_{\max} = C \times m_p$, otherwise $m_p = C \times m_p$ and go to 6.

5.2. Determining the MLCC in closed-loop case

Since in closed-loop case the system input is computed by controller, so for applying the actuator constraints instead of defining the load coefficient we put a constraint on the controller output such that: if controller output does not violate the actuator constraint the system input is equal to controller output otherwise it is equal to the bounds of actuator constraints. The accuracy constraint is checked by the distance between desired trajectory and actual trajectory which must not violate the accuracy constraint (R_p). The algorithm used for finding MLCC in closed-loop case is shown in Fig. 4.

6. Simulation Studies

By applying the proposed algorithm for closed-loop plant the maximum allowable load computed as $m_{\text{load}} = 4.51$kg, meanwhile the maximum allowable loop for open loop in three iteration find as $m_{\text{load}} = 3.63$kg. The simulation results are shown in Figs. 5–10. The parameters used in simulation are given in Tables I and II.

Table I. Parameters of manipulator.

Parameter	Value	Unit
Length of Links	$L_1 = L_2 = 1$	m
Mass per length	$\rho_1 = \rho_2 = 4.68$	kg/m
Flexural rigidity	$E_1I_1 = E_2I_2 = 1025$	N m^2
Actuator stall torque	$T_{s1} = 81, T_{s2} = 29$	N m
Actuator no-load speed	$\omega_{n1} = \omega_{n2} = 3.5$	Rad/s

Table II. Parameters of controller.

Parameter	Value
Controller constants	$\lambda = \text{diag}\ (10, 10)\ \Lambda = \text{diag}\ (25, 25)$
Observer constants	$k_{11} = 100, k_{12} = 1\text{e}5\ k_{21} = 100, k_{22} = 1\text{e}5$

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

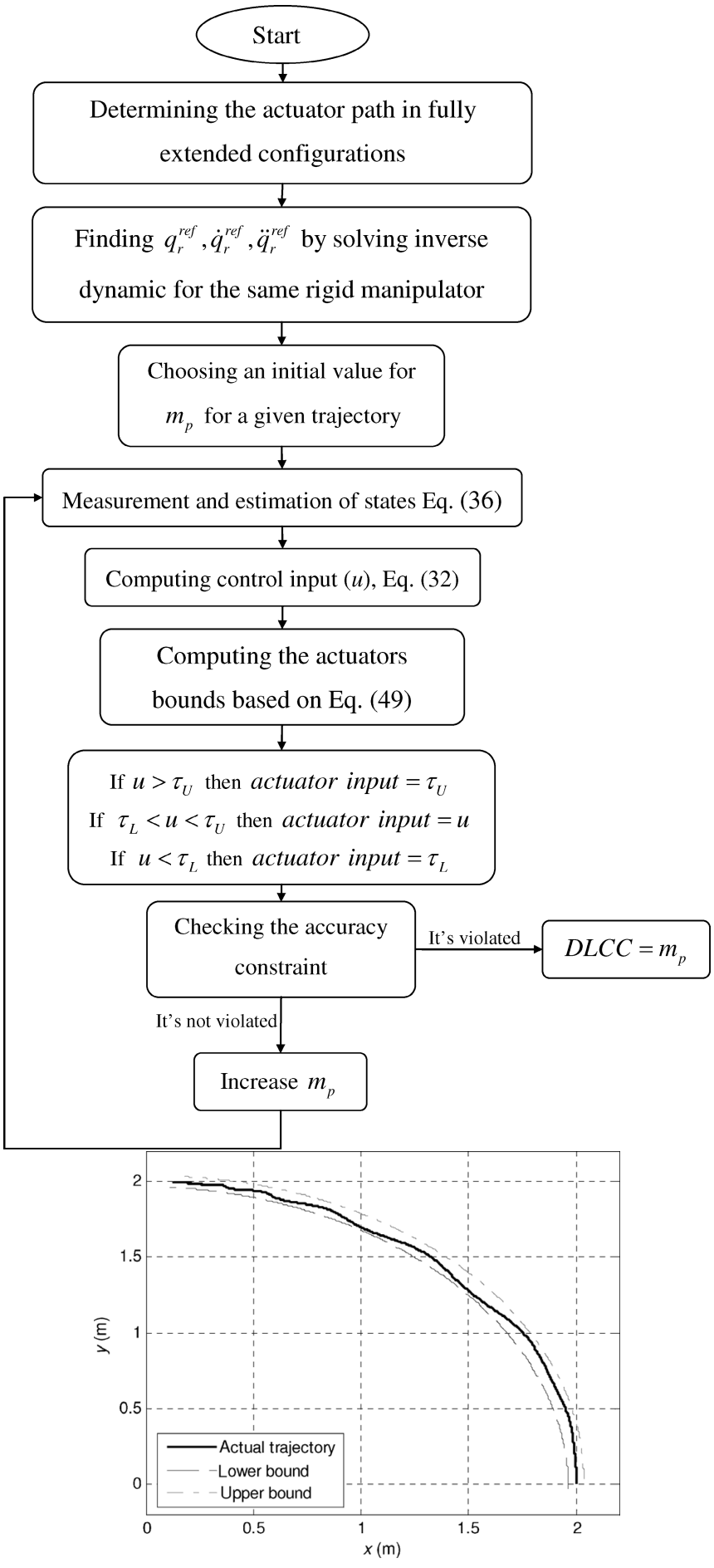

Fig. 5. Flowchart of computing DLCC. End effector path, (open-loop case).

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

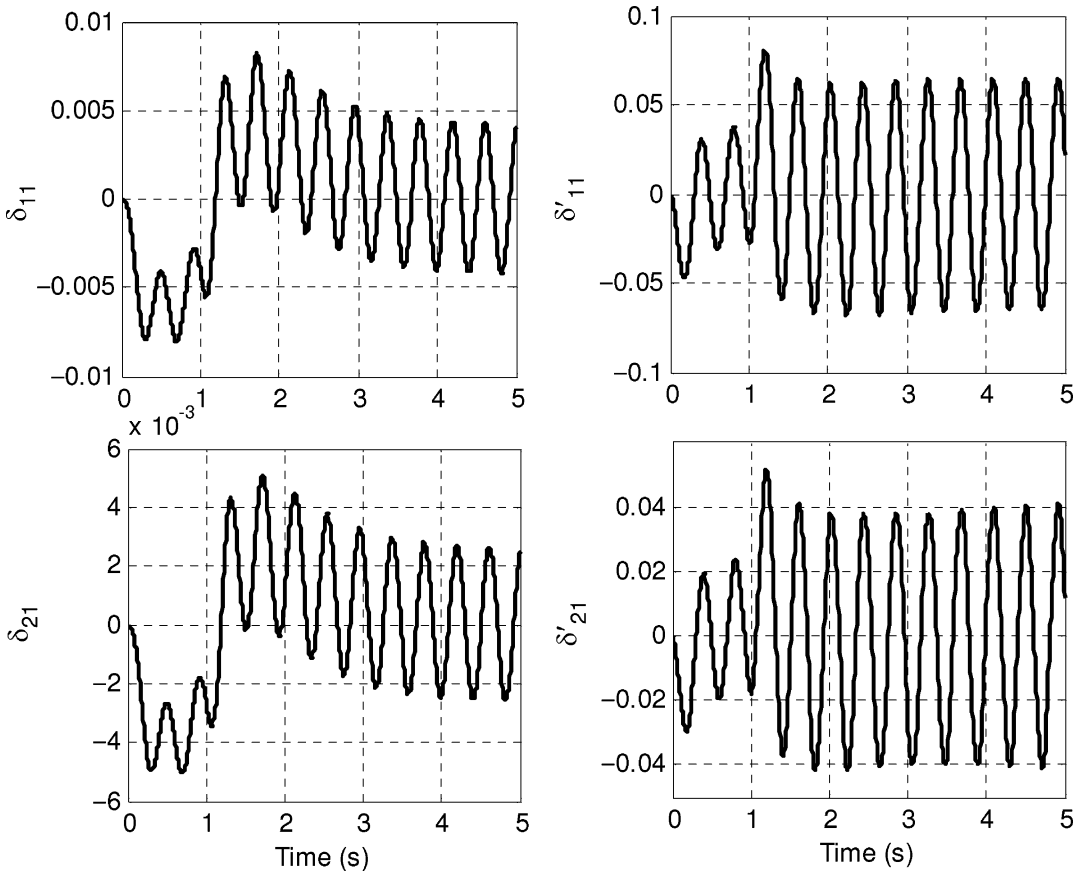

Fig. 6. Flexible variables in open-loop case (δ_{ij} is the jth flexible variable of the ith link and δ'_{ij} is the rate of change of that flexible variable).

Figure 6 shows the elastic variables in open-loop case which can be seen that these variables do not converge to zero. Figure 7 shows that in closed-loop case the capacity of actuators is better used respect to open-loop case. Figure 9 shows that the good performance of state observer in estimating of elastic variable velocities. Moreover it shows the convergence of flexible-link vibrations. Figure 10 is the elastic variables used in dynamic of system but it is neglected in controller and observer design to show the robustness of controller respect to unstructured uncertainties.

7. Conclusion

In this paper, a new formulation is developed to determine the maximum load for flexible-link manipulator in the presence

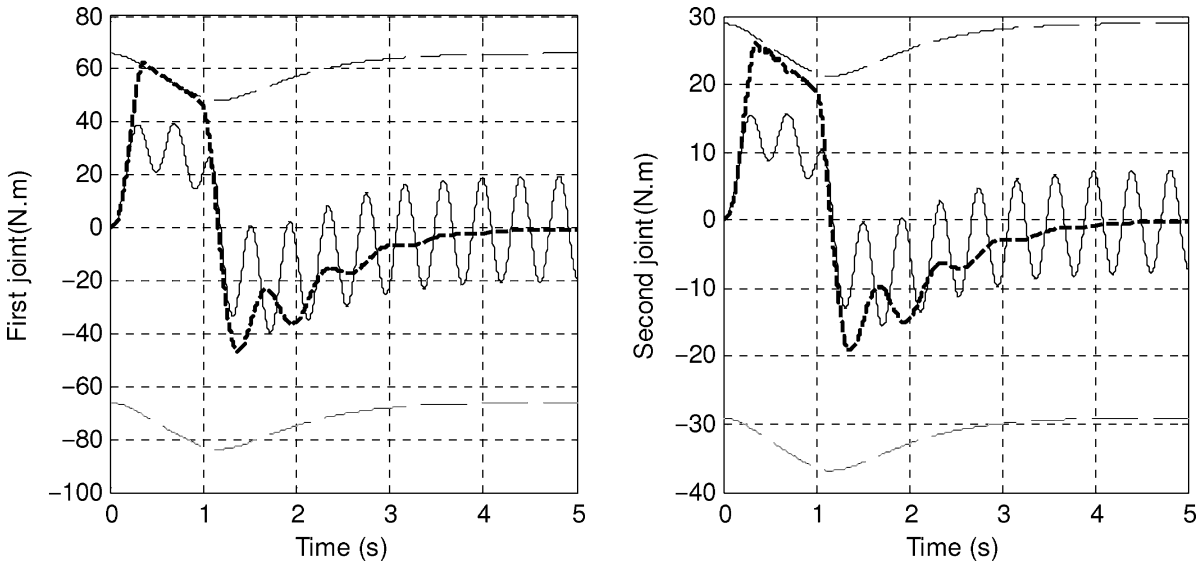

Fig. 7. Control input in two cases, open loop (solid thin line) and closed loop (dashed thick line).

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

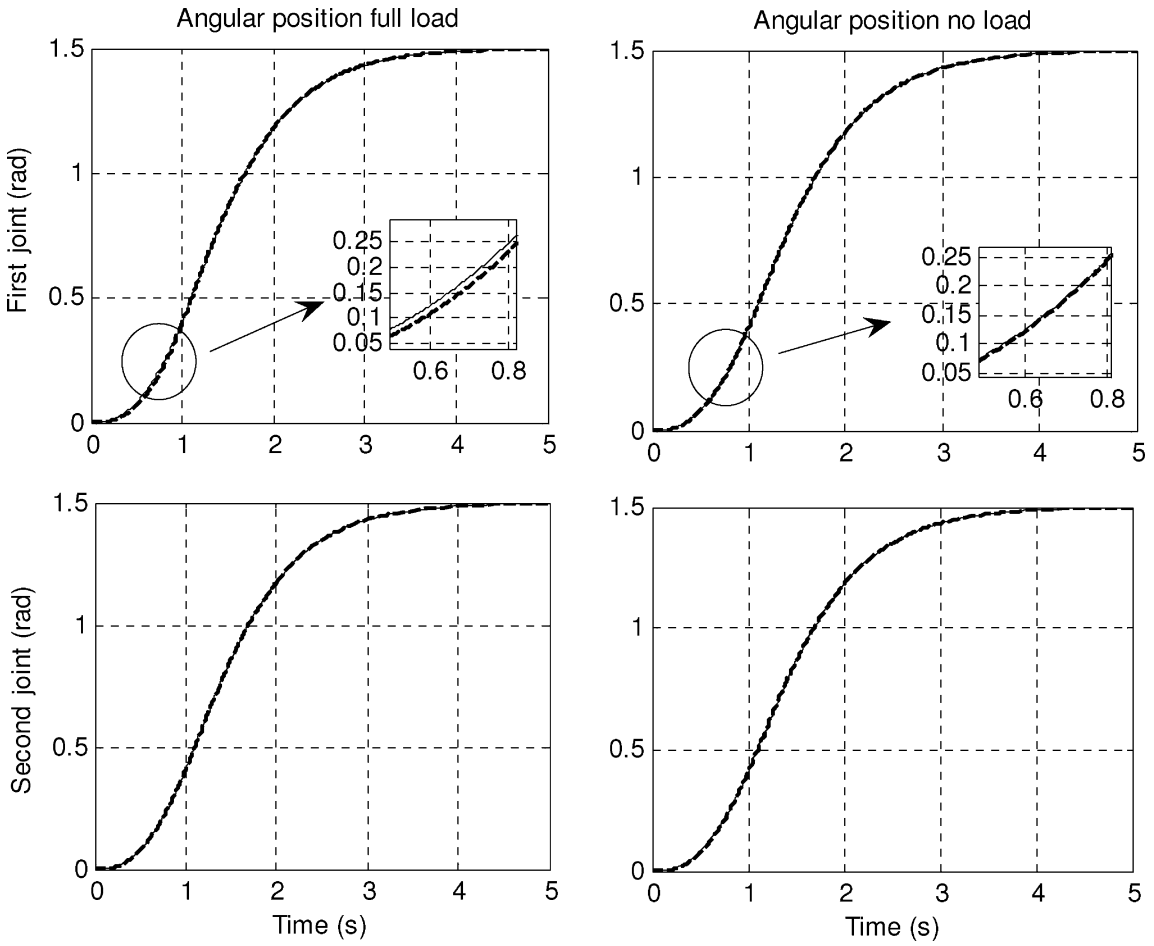

Fig. 8. Angular position in two cases (without load and full load), the reference signal is shown in solid thin line and the actual trajectories are shown in dashed thick line, (closed-loop case).

of controller. The controller is designed based on sliding mode method and for alleviation of chattering phenomena a fuzzy variable boundary layer is used. However in control law, the velocity of elastic variables is used which cannot be measured easily. So a nonlinear state observer is designed based on sliding mode approach to estimate these variables. The controllers and the observer have been designed in this study based on a simplified version of the model of the

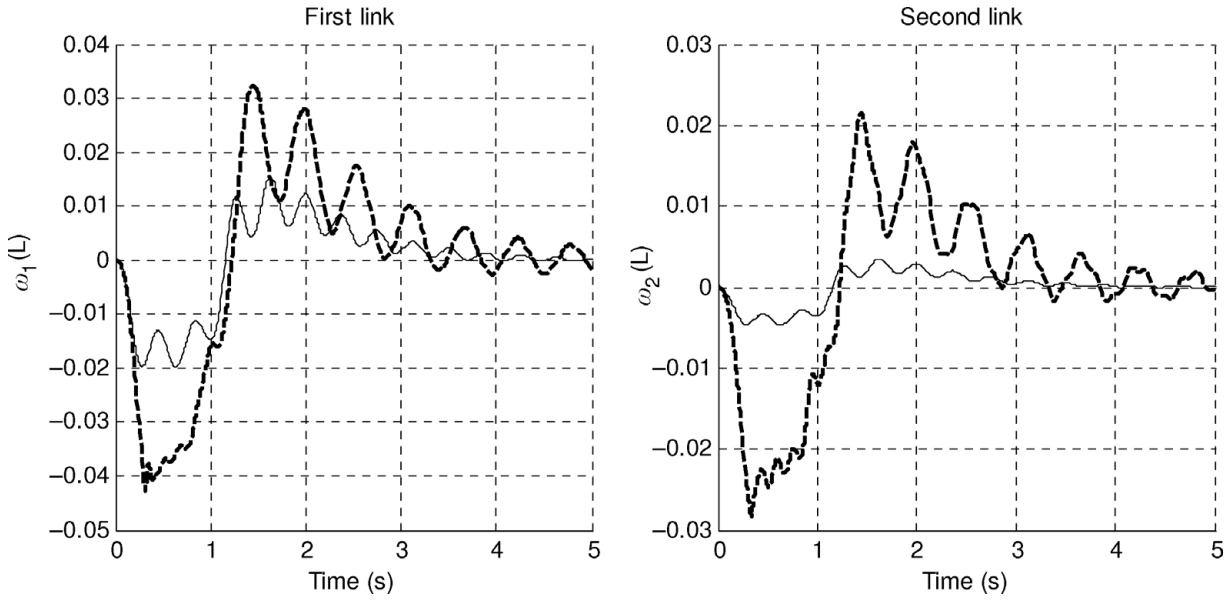

Fig. 9. Deflection of two links in two cases without load (solid thin line) and full load (dashed thick line), (closed-loop case).

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

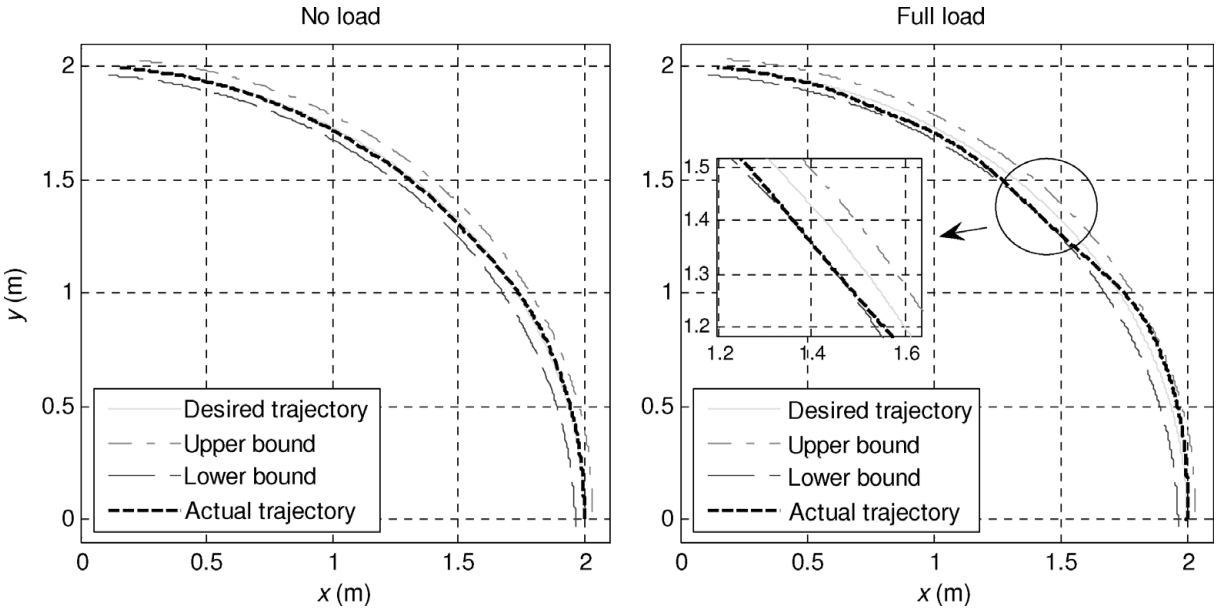

Fig. 10. End effector path in two cases (without load and full load), (closed-loop case).

arm, which only the first elastic mode of the link is taken into account while for the model the second mode shape is also considered to investigate the effects of unstructured uncertainties on the overall performance of the closed-loop system. By applying the proposed algorithm for closed-loop case the maximum allowable load computed as $m_{\text{load}} = 4.51$ kg, meanwhile the maximum allowable loop for open loop found as $m_{\text{load}} = 3.63$ kg.

References

1. L. T. Wang and B. Ravani, "Dynamic load carrying capacity of mechanical manipulators—Part 1," *J. Dyn. Sys. Meas. Control* **110**, 46–52 (1988).
2. Y. L. Yao, M. H. Korayem and A. Basu, "Maximum allowable load of flexible manipulators for given dynamic trajectory," *Rob. Comp.-Integrated Manuf.* **10**(4), 301–309 (1993).
3. M. H. Korayem and A. Basu, "Formulation and numerical solution of elastic robot dynamic motion with maximum load carrying capacity," *Robotica* **12**, 253–261 (1994).
4. Y. S. Zhao, L. Lu, T. S. Zhao, Y. H. Du and Z. Huang, "The novel approaches for computing the dynamic load-carrying capacity of multiple cooperating robotic manipulators," *Mech. Mach. Theory* **34**, 637–643 (1998).
5. M. H. Korayem, H. Ghariblu and A. Basu, "Maximum allowable load of mobile manipulators for two given end points of end effector," *Int. J. Adv. Manuf. Technol.* **24**, 743–751 (2004).
6. M. H. Korayem and A. Nikoobin, "Maximum payload for flexible joint manipulators in point-to-point task using optimal control approach," *Int. J. Adv. Manuf. Technol.* **38**(9–10), 1045–1060 (2008).
7. M. H. Korayem, A. Heidari and A. Nikoobin, "Maximum allowable dynamic load of flexible mobile manipulators using finite element approach," *Int. J. Adv. Manuf. Technol.* **36**(5–6), 606–617 (2008).
8. M. H. Korayem and M. Bamdad, "Dynamic load-carrying capacity of cable-suspended parallel manipulators," *Int. J. Adv. Manuf. Technol.* **44**(7–8), 1133–1143 (2009).
9. M. H. Korayem and A. Nikoobin, "Maximum payload path planning for redundant manipulator using indirect solution of optimal control problem," *Int. J. Adv. Manuf. Technol.* **44**(7–8), 725–736 (2008).
10. M. H. Korayem, F. Davarpanah and H. Ghariblu, "Load carrying capacity of flexible joint manipulators with feedback linearization," *Int. J. Adv. Manuf. Technol.* **29**, 389–397 (2006).
11. M. H. Korayem and A. Pilechian, "Maximum allowable load of elastic joint robots: sliding mode control approach," *Amirkabir J. Sci. & Technol.* **17**(65), 75–82 (2007).
12. A. Isidori, *Nonlinear Control Systems*, 3rd ed. (Springer-Verlag, London, 1995).
13. A. Arisoy, M. Gokasan and S. Bogosyant, "Sliding Mode Based Position Control of a Flexible-Link Arm," *12th International Power Electronics and Motion Control Conference, Proceedings*, art. no. 4061738, (2007) pp. 402–407.
14. J. Wang, A. B. Rad and P. T. Chan, "Indirect adaptive fuzzy sliding mode control: Part I: fuzzy switching," *Fuzzy sets sys.* **122**, 21–30 (2001).
15. N. Derbel and A. M. Alimi, "Design of a sliding mode controller by fuzzy logic," *Int. J. Rob. Autom.* **21**(4), 241–246 (2006).
16. C. Y. Liang and J. P. Su, "A new approach to the design of a fuzzy sliding mode controller," *Fuzzy Sets Sys.* **139**, 111–124 (2003).
17. M. M. Abdelhameed, "Enhancement of sliding mode controller by fuzzy logic with application to robotic manipulators," *Mechatronics* **15**, 439–458 (2005).
18. H. F. Ho, Y. K. Wong and A. B. Rad, "Robust fuzzy tracking control for robotic manipulators," *Simul. Model. Prac. Theory* **15**, 801–816 (2007).
19. A. L. Zuyev, "Observability of a flexible manipulator with a payload," *Phys. Con.* 523–526 (2005).
20. T. D. Nguyen and O. Egeland, "Observer design for a flexible robot arm with a tip load," *American Control Conf.* 1389–1394 (2005).
21. N. G. Chalhoub and G. A. Kfoury, "Development of a robust nonlinear observer for a single-link flexible manipulator," *Nonlinear Dyn.* **39**, 217–233 (2005).
22. S. H. Lee and C. W. Lee, "Hybrid control scheme for robust tracking of two-link flexible manipulator," *J. Intell. Rob. Sys.* **32**, 389–410 (2001).
23. J. J. Slotine and W. Li, *Applied Nonlinear Control* (Prentice-Hall, Englewood Cliffs, NJ, 1991).

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

Appendix: Equation of motion for a two-link planar flexible manipulator

The equation of motion for two-link flexible manipulator can be written as

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau, \tag{A 1}$$

where

$$q = \begin{bmatrix} thet_1 \\ thet_2 \\ q_{11} \\ q_{12} \\ q_{21} \\ q_{22} \end{bmatrix}, \tag{A 2}$$

$$\tau = \begin{bmatrix} u_1 \\ u_2 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{A 3}$$

where $thet_i$ is angular position of the ith link and q_{ij} is the jth flexible variable of the ith link. $N(q, \dot{q})$ can be expressed by

$$N(q, \dot{q}) = C(q, \dot{q}) + Kq, \tag{A 4}$$

The elements of $M(q)$, $C(q, \dot{q})$ and K are given as below

M(1, 1) = 4.1000 + q11^2 + q12^2 + (1. + (.92447*q11 −.92529*q12)^2)*(5.4800 + Mtip) + Mtip*(1. + (.92447*q11 − .92529*q12)*(.92447*q21 − .92529*q22)*cos(−thet1 + thet2) − (.92447* × q21 − .92529*q22 − .92447*q11 + .92529* × q12)*sin(−thet1 + thet2)) − 1.6939* × sin(−thet1 + thet2)*q21 − .93883* × sin(−thet1 + thet2)*q22 + 1.5660* × cos(−thet1 + thet2)*q11*q21 − .86769* × cos(−thet1 + thet2)*q12*q22–1.5672* × cos(−thet1 + thet2)*q12*q21 + .86800* × cos(−thet1 + thet2)*q11*q22 + 2.1633* × sin(−thet1 + thet2)*q11–2.1652* × sin(−thet1 + thet2)*q12;

M(1, 2) = Mtip*(1. + (.92447*q11−.92529*q12)*(.92447* × q21 − .92529*q22)*cos(−thet1 + thet2) − (.92447*q21 − .92529*q22 − .92447*q11 + .92529*q12)*sin(−thet1 + thet2))−1.6939* × sin(−thet1 + thet2)*q21 − .93883* × sin(−thet1 + thet2)*q22 + 1.5660* × cos(−thet1 + thet2)*q11*q21 − .86769* × cos(−thet1 + thet2)*q12*q22 + 4.1000 − 1.5672*cos(−thet1 + thet2)*q12*q21 + .86800*cos(−thet1 + thet2)*q11*q22 + 2.1633*sin(−thet1 + thet2)*q11–2.1652* × sin(−thet1 + thet2)*q12 + Mtip + Mtip* ×(.92447*q21−.92529*q22)^2+q21^2+q22^2;

M(1, 3) = 6.2967 + .92447*Mtip + 2.1633*cos(−thet1 + thet2) − 1.5660*sin(−thet1 + thet2)*q21 − .86801*sin(−thet1 + thet2)*q22 + Mtip* × (.92447*cos(−thet1 + thet2) − 1.*(.85465* × q21 − .85540*q22)*sin(−thet1 + thet2));

M(1, 4) = −4.8748 − .92529*Mtip − 2.1652*cos(−thet1 + thet2) + 1.5672*sin(−thet1 + thet2)*q21 + .86839*sin(−thet1 + thet2)*q22 + Mtip* × (−.92529*cos(−thet1 + thet2)−1.*(−.85540* × q21 + .85616*q22)*sin(−thet1 + thet2));

M(1, 5) = 1.6939*cos(−thet1 + thet2) + 1.5660* × sin(−thet1 + thet2)*q11–1.5672*sin(−thet1 + thet2)*q12 + Mtip*(.92447*cos(−thet1 + thet2) + (.85465*q11 − .85540*q12)* × sin(−thet1 + thet2)) + 1.2305 + .92447*Mtip;

M(1, 6) = .93881*cos(−thet1 + thet2) + .86801*sin(−thet1 + thet2)*q11 − .86839*sin(−thet1 + thet2)*q12 + Mtip*(−.92529*cos(−thet1 + thet2) + (−.85540*q11 + .85616*q12)*sin(−thet1 + thet2)) + .19581 − .92529*Mtip;

M(2, 1) = Mtip*(1. + (.92447*q11−.92529*q12)*(.92447* × q21 − .92529*q22)*cos(−thet1 + thet2) − 1.* × (.92447*q21 − .92529*q22 − .92447*q11 + .92529*q12)*sin(−thet1 + thet2)) − 1.6939* × sin(−thet1 + thet2)*q21−.93883*sin(−thet1 + thet2)*q22 + 1.5660*cos(−thet1 + thet2)*

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

× q11*q21 − .86769*cos(−thet1 + thet2)*q12*
× q22 + 2.3400–1.5672*cos(−thet1 + thet2)*
× q12*q21 + .86800*cos(−thet1 + thet2)*q11*
× q22 + 2.1633*sin(−thet1 + thet2)*q11
− 2.1652*sin(−thet1 + thet2)*q12;

M(2, 2) = 1.7600 + Mtip + Mtip*(.92447*q21 − .92529*
× q22)^2 + q21^2 + q22^2;

M(2, 3) = 2.1633*cos(−thet1 + thet2) − 1.5660*sin(−thet1
+ thet2)*q21 − .86801*sin(−thet1 + thet2)*q22
+ Mtip*(.92447*cos(−thet1 + thet2) − 1.*
×(.85465*q21−.85540*q22)*sin(−thet1 + thet2));

M(2, 4) = −2.1652*cos(−thet1 + thet2) + 1.5672*
× sin(−thet1 + thet2)*q21 + .86839*sin(−thet1
+ thet2)*q22 + Mtip*(−.92529*cos(−thet1
+ thet2) − 1.*(−.85540*q21 + .85616*q22)*
× sin(−thet1 + thet2));

M(2, 5) = 1.2305 + .92447*Mtip;

M(2, 6) = .19581 − .92529*Mtip;

M(3, 1) = 6.2967 + .92447*Mtip;

M(3, 2) = 2.1633*cos(−thet1 + thet2) − 1.5660*sin(−thet1
+ thet2)*q21 − .86801*sin(−thet1 + thet2)*q22
+ Mtip*(.92447*cos(−thet1 + thet2) − 1.*
×(.85465*q21−.85540*q22)*sin(−thet1 + thet2));

M(3, 3) = 5.6835 + .85465*Mtip;

M(3, 4) = −4.6876 − .85540*Mtip;

M(3, 5) = 1.5660*cos(−thet1 + thet2) + .85465*
× cos(−thet1 + thet2)*Mtip;

M(3, 6) = .86801*cos(−thet1 + thet2) − .85540*
× cos(−thet1 + thet2)*Mtip;

M(4, 1) = −4.8748 − .92529*Mtip;

M(4, 2) = −2.1652*cos(−thet1 + thet2) + 1.5672*
× sin(−thet1 + thet2)*q21 + .86839*sin(−thet1
+ thet2)*q22 + Mtip*(−.92529*cos(−thet1
+ thet2) − 1.*(−.85540*q21 + .85616*q22)*
× sin(−thet1 + thet2));

M(4, 3) = −4.6876 − .85540*Mtip;

M(4, 4) = 5.6917 + .85616*Mtip;

M(4, 5) = −1.5672*cos(−1.*thet1 + thet2) − .85540*
× cos(−1.*thet1 + thet2)*Mtip;

M(4, 6) = −.86839*cos(−thet1 + thet2) + .85616*
× cos(−thet1 + thet2)*Mtip;

M(5, 1) = 1.6939*cos(−thet1 + thet2) + 1.5660*
× sin(−thet1 + thet2)*q11–1.5672*sin(−thet1
+ thet2)*q12 + Mtip*(.92447*cos(−thet1
+ thet2) + (.85465*q11 − .85540*q12)*
× sin(−thet1 + thet2));

M(5, 2) = 1.2305 + .92447*Mtip;

M(5, 3) = 1.5660*cos(−thet1 + thet2) + .85465*
× cos(−thet1 + thet2)*Mtip;

M(5, 4) = −1.5672*cos(−thet1 + thet2) − .85540*
× cos(−thet1 + thet2)*Mtip;

M(5, 5) = 1. + .85465*Mtip;

M(5, 6) = −.85540*Mtip;

M(6, 1) = .93881*cos(−thet1 + thet2) + .86801*sin(−thet1
+ thet2)*q11 − .86839*sin(−thet1 + thet2)*q12
+ Mtip*(−.92529*cos(−thet1 + thet2)
+ (−.85540*q11 + .85616*q12)*sin(−thet1
+ thet2));

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

```
M(6, 2) = .19581 − .92529*Mtip;

M(6, 3) = .86801*cos(−thet1 + thet2) − .85540*
          × cos(−thet1 + thet2)*Mtip;

M(6, 4) = −.86839*cos(−thet1 + thet2) + .85616*
          × cos(−thet1 + thet2)*Mtip;

M(6, 5) = −.85540*Mtip;

M(6, 6) = 1. + .85616*Mtip;

C(1) = 1.6939*thetdot1^2*cos(−thet1 + thet2)*q21
       + .93858*thetdot1^2*cos(−thet1 + thet2)*q22
       + 7.9995*thetdot1*q11*qdot11–2.1633*thetdot1^2*
       × cos(−thet1 + thet2)*q11 − .93822*cos(−thet1
       + thet2)*thetdot2^2*q22–4.3303*thetdot1*
       × sin(−thet1 + thet2)*qdot12–8.0066*thetdot1*
       × q12*qdot11–8.0066*thetdot1*q11*qdot12
       + 8.0136*thetdot1*q12*qdot12 + 2.1633*thetdot2^2*
       × cos(−thet1 + thet2)*q11 + 2.1652*thetdot1^2*
       × cos(−thet1 + thet2)*q12 + 4.3265*thetdot1*
       × sin(−thet1 + thet2)*qdot11 + (1.4791*q21
       − 1.4805*q22)*(.92447*q21 − .92529*q22)*
       × thetdot2–1.6939*cos(−thet1 + thet2)*thetdot2^2*
       × q21–1.8768*sin(−thet1 + thet2)*thetdot2*qdot22
       − 2.1652*thetdot2^2*cos(−thet1 + thet2)*q12
       − 3.3879*sin(−thet1 + thet2)*thetdot2*qdot21
       + 1.7348*thetdot1*cos(−thet1 + thet2)*qdot11*q22
       + .86787*sin(−thet1 + thet2)*thetdot1^2*q11*q22
       + 3.1319*thetdot1*cos(−thet1 + thet2)*qdot11*q21
       − 1.7361*thetdot1*cos(−thet1 + thet2)*qdot12*q22
       − 1.5672*sin(−thet1 + thet2)*thetdot1^2*q12*q21
       − .86808*sin(−thet1 + thet2)*thetdot1^2*q12*q22
       + 1.5660*sin(−thet1 + thet2)*thetdot1^2*q11*q21
       − 3.1347*thetdot1*cos(−thet1 + thet2)*qdot12*q21
       − 1.5660*thetdot2^2*sin(−thet1 + thet2)*q11*q21
       + .86800*thetdot2^2*sin(−thet1 + thet2)*q12*q22
       + 1.7350*thetdot2*cos(−thet1 + thet2)*q11*qdot22
       − 3.1347*thetdot2*cos(−thet1 + thet2)*q12*qdot21
       + 3.1319*thetdot2*cos(−thet1 + thet2)*q11*qdot21
       − 1.7360*thetdot2*cos(−thet1 + thet2)*q12*qdot22
       − .86794*thetdot2^2*sin(−thet1 + thet2)*q11*q22
       + 1.5672*thetdot2^2*sin(−thet1 + thet2)*q12*q21
       + 2.3400*sin(−thet1 + thet2)*thetdot1^2 + Mtip*
       × ((1.8489*q21–1.8506*q22)*(.92447*q21
       − .92529*q22)*thetdot2 + (thetdot1 + 1.8489*
       × qdot11–1.8506*qdot12 + (.92447*q11 − .92529*
       × q12)*(.92447*q21 − .92529*q22)*thetdot1)*
       × thetdot1*sin(−thet1 + thet2) + ((.92447*q21
       − .92529*q22)*thetdot1–1.*(.92447*q11 − .92529*
       × q12)*thetdot1 + (1.8489*q11–1.8506*q12)*
       × (.92447*q21 − .92529*q22))*thetdot1*
       × cos(−1.*thet1 + thet2)) + (2.*q11*qdot11
       + 2.*q12*qdot12)*thetdot1 + Mtip*((1.8489*q11
       − 1.8506*q12)*(.92447*qdot11 − .92529*qdot12)*
       × thetdot1–1.*(thetdot2 + 1.8489*qdot21–1.8506*
       × qdot22 + (.92447*q11 − .92529*q12)*(.92447*
       × q21 − .92529*q22)*thetdot2)*thetdot2*
       × sin(−1.*thet1 + thet2) − 1.*((.92447*q21
       − .92529*q22)*thetdot2–1.*(.92447*q11 − .92529*
       × q12)*thetdot2 − (1.8489*q11–1.8506*q12)*
       × (.92447*qdot21 − .92529*qdot22))*thetdot2*
       × cos(−1.*thet1 + thet2)) − 2.3400*thetdot2^2*
       × sin(−thet1 + thet2) + (2.*q21*qdot21 + 2.*q22*
       × qdot22)*thetdot2 + (1.4791*q11–1.4805*q12)*
       × (.92447*qdot11 − .92529*qdot12)*thetdot1;

C(2) = Mtip*((1.8489*q21 − 1.8506*q22)*(.92447*q21
       − .92529*q22)*thetdot2 + (thetdot1 + 1.8489*
       × qdot11−1.8506*qdot12+(.92447*q11−.92529*
       × q12)*(.92447*q21 − .92529*q22)*thetdot1)*
       × thetdot1*sin(−1.*thet1 + thet2) + ((.92447*q21
       − .92529*q22)*thetdot1 − (.92447*q11 − .92529*
       × q12)*thetdot1 + (1.8489*q11–1.8506*q12)*
       × (.92447*q21 − .92529*q22))*thetdot1*cos(−thet1
       + thet2)) − 3.1347*thetdot1*cos(−thet1 + thet2)*
       × qdot12*q21 + 1.7348*thetdot1*cos(−1.*thet1
       + thet2)*qdot11*q22 − .86808*sin(−thet1 + thet2)*
       × thetdot1^2*q12*q22+3.1319*thetdot1*cos(−thet1
```

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

+ thet2)*qdot11*q21–1.7361*thetdot1*cos(−thet1
+ thet2)*qdot12*q22 + .86787*sin(−thet1 + thet2)*
× thetdot1^2*q11*q22–4.3303*thetdot1*sin(−thet1
+ thet2)*qdot12 + 4.3265*thetdot1*sin(−thet1
+ thet2)*qdot11–2.1633*thetdot1^2*cos(−thet1
+ thet2)*q11 + 1.6939*thetdot1^2*cos(−thet1
+ thet2)*q21 + 2.1652*thetdot1^2*cos(−thet1
+ thet2)*q12 + .93858*thetdot1^2*cos(−thet1
+ thet2)*q22–1.5672*sin(−thet1 + thet2)*
× thetdot1^2*q12*q21 + 1.5660*sin(−thet1
+ thet2)*thetdot1^2*q11*q21 + (1.4791*q21
− 1.4805*q22)*(.92447*q21−.92529*q22)*thetdot2
+ (2.*q21*qdot21 + 2.*q22*qdot22)*thetdot2
+ 2.3400*sin(−thet1 + thet2)*thetdot1^2;

C(3) = .80000*thetdot1^2*(−.85465*q11 + .85540*q12)
+ Mtip*(thetdot1^2*(−.85465*q11 + .85540*q12)
− sin(−1.*thet1 + thet2)*thetdot2*(.92447*thetdot2
+ 1.7093*qdot21–1.7108*qdot22) − cos(−1.*thet1
+ thet2)*thetdot2^2*(.85465*q21 − .85540*q22))
−thetdot1^2*q11–3.9998*thet1^2*q11 + 4.0033*
× thet1^2*q12–1.5660*cos(−thet1 + thet2)*
× thetdot2^2*q21–1.7348*sin(−thet1 + thet2)*
× thetdot2*qdot22–3.1319*sin(−thet1 + thet2)*
× thetdot2*qdot21 − .86793*cos(−thet1 + thet2)*
× thetdot2^2*q22–2.1633*thetdot2^2*sin(−thet1
+ thet2);

C(4) = 4.0033*thet1^2*q11 − 4.0068*thet1^2*q12
+ 1.5672*cos(−thet1 + thet2)*thetdot2^2*q21
+ 1.7360*sin(−thet1 + thet2)*thetdot2*qdot22
+ 3.1344*sin(−thet1 + thet2)*thetdot2*qdot21
+ .86800*cos(−thet1 + thet2)*thetdot2^2*q22
− thetdot1^2*q12 + .80000*thetdot1^2*(.85540*
× q11 − .85616*q12) + 2.1652*thetdot2^2*
× sin(−thet1 + thet2) + Mtip*(thetdot1^2*(.85540*
× q11 − .85616*q12) − 1.*sin(−thet1 + thet2)*
× thetdot2*(−.92529*thetdot2–1.7108*qdot21
+ 1.7123*qdot22) − 1.*cos(−thet1 + thet2)*
× thetdot2^2*(−.85540*q21 + .85616*q22));

C(5) = −1.*Mtip*(thetdot2^2*(.85465*q21 − .85540*q22)
− 1.*thetdot1*sin(−thet1 + thet2)*(.92447*thetdot1
+ 1.7093*qdot11–1.7108*qdot12) + thetdot1^2*
× cos(−thet1 + thet2)*(.85465*q11 − .85540*q12))
− 3.1347*thetdot1*sin(−thet1 + thet2)*qdot12
+ 3.1319*thetdot1*sin(−thet1 + thet2)*qdot11
− 1.5660*thetdot1^2*cos(−thet1 + thet2)*q11
+ 1.5672*thetdot1^2*cos(−thet1 + thet2)*q12
− 1.*thetdot2^2*q21 + 1.6939*sin(−thet1
+ thet2)*thetdot1^2;

C(6) = −Mtip*(thetdot2^2*(−.85540*q21 + .85616*q22)
− 1.*thetdot1*sin(−thet1 + thet2)*(−.92529*
× thetdot1–1.7108*qdot11 + 1.7123*qdot12)
+ thetdot1^2*cos(−thet1 + thet2)*(−.85540*q11
+ .85616*q12)) − 1.7357*thetdot1*sin(−thet1
+ thet2)*qdot12 + 1.7347*thetdot1*sin(−thet1
+ thet2)*qdot11 − .86792*thetdot1^2*cos(−thet1
+ thet2)*q11 + .86794*thetdot1^2*cos(−thet1
+ thet2)*q12–1.*thetdot2^2*q22 + .93858*
× sin(−thet1 + thet2)*thetdot1^2;

K = [zeros(2)zeros(2, 4); zeros(4, 2)K$_1$];

where the elements of K_1 are given as below

K1(1,1) = 2707.6;

K1(1,2) = 0.;

K1(1,3) = 0.;

K1(1,4) = 0.;

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press

K1(2,1) = 0.;

K1(2,2) = .10634e6;

K1(2,3) = 0.;

K1(2,4) = 0.;

K1(3,1) = 0.;

K1(3,2) = 0.;

K1(3,3) = 2707.6;

K1(3,4) = 0.;

K1(4,1) = 0.;

K1(4,2) = 0.;

K1(4,3) = 0.;

K1(4,4) = .10634e6;

https://doi.org/10.1017/S0263574709990798 Published online by Cambridge University Press