Exponential Convergence of a Unified CLF Controller for Robotic Systems under Parameter Uncertainty

Shishir Kolathaya[1] and Aaron D. Ames[2]

Abstract— **This paper presents a novel method for achieving exponential convergence of a Control Lyapunov Function (CLF) based controller in a n-DOF robotic system in the presence of parameter uncertainty. Utilizing the linearity of parameters in the equations of motion, we construct the regressor and augment the state space of the robot to include a vector of unknown parameters, called base inertial parameters. The augmented state space can be utilized to realize an optimal controller that is exponentially stable while simultaneously estimating the parameters online. To achieve this result, acceleration data for a given torque input is measured and used to compute the regressor. This, in turn, is used to compute the set of base inertial parameters in the form of linear equality constraints. By demonstrating that it is not necessary for the estimated parameters to converge to the actual parameters, but rather convergence is only needed on a specified space, we are able to construct a quadratic program enforcing convergence. The end result is that exponential convergence of a Control Lyapunov Function can be guaranteed, in an optimal fashion, without prior knowledge of the parameters.**

I. INTRODUCTION

During the 1980's and 1990's, control of robotic manipulators under parameter uncertainty was studied in depth; see, for example, [12], [13]. The special property of the equations of motion of these n-link manipulators, viz. linearity of the uncertain terms in the expression, was utilized to generate the so called regressors (see [3], [13], [16]). By knowing the acceleration data of the robot manipulators the regressor was computed and several adaptive control schemes were used in [6], [9] to realize asymptotic convergence of joint angle outputs. [12] suggested a different method to realize asymptotic convergence without using the acceleration data. The general idea is to use the unknown parameters as the adaptive variables in the controller, and by using a suitable control law for these adaptive variables, the derivative of the Lyapunov candidate function for the outputs is rendered negative semi-definite.

These adaptive control controllers would achieve asymptotic convergence, but, in applications like bipedal locomotion, stronger bounds on convergence are necessary: exponential convergence. This is due to the time duration of completing a step being small. In addition, any of the adaptive schemes mentioned do not determine the parameters of the robot explicitly. To be more precise, these control schemes are designed such that determining the parameters of the robot is completely eliminated. During the 1990's learning methods were adopted with adaptive control to achieve exponential convergence. [7], [8] uses a learning controller and exponential convergence is achieved by having persistence of excitation in the training tasks [8]. These controllers are specific to particular learning or a repetitive task.

This paper presents a method to extend the method used in [9] to realize an exponentially stable controller for a robot with uncertain parameters. In other words, joint angular position, velocity and acceleration will be used to compute the regressor matrix. The regressor matrix, coupled with torque data will be posed as an equality constraint to determine the parameters of the robot. In this framework, a higher number of samples yields better estimate of the robot parameters. Utilizing this estimate of the parameters, a state space model of the uncertain robot in terms of the outputs will be constructed. By augmenting the state space with the estimated parameters, a suitable Lyapunov candidate will be chosen which takes into consideration both the outputs and the parameters. With this Lyapunov candidate, an optimal control law will be proposed through the use of a novel quadratic program-this will guarantee exponential convergence of the outputs. In other words, a suitable controller will be chosen such that this Lyapunov candidate is strictly bounded by two decaying exponentials.

The contents of the paper are divided in the following manner: Section II starts with a brief introduction to the n-link rigid body robotic systems and the types of controllers that will be considered in this paper. Specifically, two types of model based contollers are discussed: feedback linearization and the method of computed torque. The use of Control Lyapunov Functions (CLFs) as a means to propose control laws as a quadratic program will be discussed. Section III will decribe the robot model with uncertain parameters. Specifically the concept of using the Regressor (Y) and the base inertial parameters will be explained in detail. Section III will also present the main result of the paper: a controller that yields exponential stability even in the presence of parameter uncertainty. The state space model for the uncertain robot model in terms of the outputs will be introduced first, and then the Control Lyapunov Function (CLF) for realizing the aforementioned objective will be introduced. Finally, Section IV will show the simulation results of this controller, demonstrating tracking performances of both the outputs and the parameters for a 5-link fully actuated serial chain manipulator.

This work is supported by the National Science Foundation through grants CNS-0953823 and CNS-1136104

[1]Shishir Kolathaya is with Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA shishirny@tamu.edu

[2]Aaron D. Ames is with the Faculty of Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA aames@tamu.edu

II. Robot Model and Control

A robotic model can be modeled as a n-link manipulator. Given the configuration space $\mathbb{Q} \subset \mathbb{R}^n$, with the coordinates $q \in \mathbb{Q}$, and velocities $\dot{q} \in T_q\mathbb{Q}$, the equations of motion (EOM) can be written as:

$$(D(q)+M)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)+E\dot{q}=B\mathrm{T}, \tag{1}$$

where $D(q) \in \mathbb{R}^{n\times n}$ is the mass matrix, $M \in \mathbb{R}^{n\times n}$ is the matrix consisting of the mass and inertia terms, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is the coriolis matrix, $G(q) \in \mathbb{R}^n$ is the gravity matrix, $E \in \mathbb{R}^{n\times n}$ is the matrix of damping terms, $\mathrm{T} \in \mathbb{R}^k$ is the torque input with k the number of actuators and $B \in \mathbb{R}^{n\times k}$ is the mapping from torque to joints.

From the equations of motion, the dynamics of the robot can be stated as: $\dot{x} = f(x) + g(x)\mathrm{T}$, with $x = (q^T, \dot{q}^T)^T \in \mathbb{R}^{2n}$, $f(x) \in \mathbb{R}^n$, $g(x) \in \mathbb{R}^{n\times k}$. If the number of degrees of freedom is more than the number of actuators, i.e., $k < n$, then the robotic system is underactuated. Accordingly, $k = n$ for the fully actuated case. For simplicity, it is assumed that the torques acting on the actuators are completely decoupled between the degrees of freedom. This is an assumption since, if there is any coupling, then the torques can be relabeled into decoupled torques. In such systems, if $k = n$ then the torque map B is just the identity matrix.

Control Implementation. Since the objective is to achieve tracking, a convenient step is to make the joint angles track a set of trajectories. We would like to generalize this by picking k functions of joint angles referred to as actual outputs $y_a : \mathbb{Q} \to \mathbb{R}^k$, which are made to track functions termed the desired outputs $y_d : \mathbb{Q} \to \mathbb{R}^k$. The objective is to drive the error $y = y_a - y_d \to 0$. These outputs are also termed virtual constraints in [14]. The outputs are picked such that they are relative degree two outputs ([10]). In other words, y_a will be functions of joint angles, and not angular velocities. If relative degree one outputs are chosen, then the controller can still be formulated in a slightly different manner [1].

We can adopt two types of model based controllers: feedback linearization and the method of computed torque.

Feedback Linearization. For the vector field already defined, $\dot{x} = f(x) + g(x)\mathrm{T}$, we can take the Lie derivatives of the outputs:

$$\dot{y} = L_f y, \quad \ddot{y} = L_f^2 y + L_g L_f y\,\mathrm{T}, \tag{2}$$

where L_f, L_g are the Lie derivatives.

Choosing outputs so that the decoupling matrix, $A(q,\dot{q}) = L_g L_f y(q,\dot{q})$ is nonsingular, we can define the following torque control law:

$$\mathrm{T}_{\mathrm{fb}} = A^{-1}(q,\dot{q})\left(-L_f^2 y(q,\dot{q}) + \mu\right), \tag{3}$$

where $\mu \in \mathbb{R}^k$ is the linear control input. The resulting dynamics will be $\ddot{y} = \mu$ (see [10]), and the resulting state space representation is:

$$\underbrace{\begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix}}_{\dot{\eta}} = \underbrace{\begin{bmatrix} 0_{k\times k} & 1_{k\times k} \\ 0_{k\times k} & 0_{k\times k} \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} y \\ \dot{y} \end{bmatrix}}_{\eta} + \underbrace{\begin{bmatrix} 0_{k\times k} \\ 1_{k\times k} \end{bmatrix}}_{G} \mu, \tag{4}$$

where μ can be conveniently picked such that the fully controllable linearized system is stabilized.

Computed Torque. The method of computed torque utilizes the idea of achieving the desired acceleration in the robot. Therefore the torque controller for a n-DOF robot can be defined as:

$$\mathrm{T}_{\mathrm{ct}} = (D(q)+M)\ddot{q}_d + C(q,\dot{q})\dot{q} + G(q) + E\dot{q}, \tag{5}$$

where $\ddot{q}_d \in \mathbb{R}^n$ is the desired acceleration required in the robot, and $\mathrm{T}_{\mathrm{ct}} \in \mathbb{R}^n$. Formally, the left hand side of (5) should be $B\mathrm{T}_{\mathrm{ct}} \in \mathbb{R}^n$ to match with (1). Since this equation of motion has n rows, the desired acceleration will lead to the range space of the mapped torque $B\mathrm{T}_{\mathrm{ct}}$ to have size n. This will be true only if $k = n$ (since the torques are assumed to be decoupled, B then is an identity matrix implying $B\mathrm{T}_{\mathrm{ct}} = \mathrm{T}_{\mathrm{ct}}$). But, if $k < n$, then T_{ct} needs to be reduced to k rows so that the operator B can be used. In other words, special care needs to be taken to pick $\ddot{q}_d$ such that the torque acting on the unactuated degree of freedom is identically zero. In other words, $n-k$ entries in T_{ct} must be identically equal to zero such that

$$\mathrm{T}_{\mathrm{ct}} = B\mathrm{T}^*_{\mathrm{ct}}, \tag{6}$$

when B is not invertible; here $\mathrm{T}^*_{\mathrm{ct}}$ is the reduced vector of torques with the zero entries removed.

Since the objective of tracking is to drive the error y to zero, we need to employ the method of computed torque utilizing these outputs. In other words, the idea is to use the outputs y to obtain the desired acceleration. Given the output y, it follows that:

$$\begin{aligned} \dot{y} &= \frac{\partial y}{\partial q}\dot{q} \\ \ddot{y} &= \underbrace{\frac{\partial y}{\partial q}}_{J}\ddot{q} + \underbrace{\dot{q}^T\frac{\partial^2 y}{\partial q^2}}_{\dot{J}}\dot{q}. \end{aligned} \tag{7}$$

If $k = n$ then the desired acceleration can be calculated as:

$$\ddot{q}_d = J^{-1}\left(\mu - \dot{J}\dot{q}\right), \tag{8}$$

where μ is control law with J being invertible due to the choice of the outputs. If $k < n$, then the rank of J reduces to k rendering it non-invertible. Therefore, for underactuated systems $\ddot{q}_d$ is picked in a different manner in order to satisfy (6). Since this paper only requires the case $k = n$, the procedure for obtaining $\ddot{q}_d$ will be omitted.

Note that with $\ddot{q}_d$ defined as in (8) and the torque controller defined as in (5), if $n = k$, the resulting dynamics of the output will be the same as (4). In other words the two controllers—feedback linearization and the method of computed torque—are equivalent. Therefore, the two controllers lead to the same output dynamics of the robot irrespective of controller chosen (see [11]).

Both the controllers, (3) and (5), assume that the model is known. If there are differences in the actual and the assumed model, then the outputs y may not be driven to zero. In other words, $\ddot{y} \neq \mu$. The dynamics of the output with model uncertainties will be discussed in the next Section.

Computing μ. One of the possible control laws for μ which renders the system $(f(x), g(x))$ stable would be:

$$\mu = -2\varepsilon L_f y(q,\dot{q}) - \varepsilon^2 y(q), \tag{9}$$

where $\varepsilon > 0$ denotes the rate of convergence. The resulting linear system on the outputs is: $\ddot{y} = -2\varepsilon\dot{y} - \varepsilon^2 y$, which exponentially drives the output $y \to 0$. More details can be found in [15]. Even though this controller gives exponential convergence, (9) is not optimal and might require high torque inputs. Therefore, a different control law, based upon Control Lyapunov Functions, is proposed which optimally drives the outputs to zero.

Control Lyapunov Function (CLF). For the control law presented in (3), if the goal is instead to find an optimal value of μ for the system $\ddot{y} = \mu$, then the problem can be formulated in the form of a Control Lyapunov Function (CLF) based controller (see [1], [2]). The control law μ is chosen such that the following quadratic cost is minimized:

$$\int_0^\infty \left(\eta^T Q \eta + \mu^T R \mu\right) dt, \;\; Q > 0, \;\; R > 0. \tag{10}$$

We can choose P to construct an *Exponentially Stable Control Lyapunov Function (ES-CLF)* that can be used to stabilize the output dynamics in an exponential fashion (see [1] for more details). Choosing the CLF:

$$V(\eta) = \eta^T P \eta. \tag{11}$$

It can be verified that this is a ES-CLF; here P is the solution to the Continuous Algebraic Riccati Equation (CARE):

$$F^T P + PF - PGG^T P + Q = 0, \tag{12}$$

where F, G are given in (4). Differentiating the function (11) yields:

$$\dot{V}(\eta) = L_f V(\eta) + L_g V(\eta)\mu, \tag{13}$$

where

$$\begin{aligned} L_F V(\eta) &= \eta^T (F^T P + PF)\eta, \\ L_G V(\eta) &= 2\eta^T PG. \end{aligned} \tag{14}$$

To find a specific value of μ, we can utilize a minimum norm controller (see [4]) which minimizes $\mu^T\mu$ subject to the inequality constraint:

$$\dot{V} = L_F V(\eta) + L_G V(\eta)\mu \leq -\gamma V, \tag{15}$$

where

$$\gamma = \frac{\lambda_{min}(Q)}{\lambda_{max}(P)}, \tag{16}$$

with λ denoting the eigenvalue of the matrix. Satisfying (15) implies exponential convergence.

These formulations motivate the introduction of a quadratic program that balances the control objective subject to torque bounds [5]:

$$\begin{aligned} \underset{(\delta,\mu)\in\mathbb{R}^{n+1}}{\text{argmin}} \;\; & p\delta^2 + \mu^T\mu \\ \text{s.t.} \;\; & \psi_0 + \psi_1 \mu \leq \delta \\ & A^{-1}(-L_f^2 y(q,\dot{q}) + \mu) \leq \mathrm{T}_{max} \\ & A^{-1}(-L_f^2 y(q,\dot{q}) + \mu) \leq -\mathrm{T}_{max}, \end{aligned} \tag{17}$$

where $p > 0$ is the penalty that allows small deviation from the desired outputs in order to facilitate use of reduced torques, $\psi_0 = L_f V(\eta) + \gamma V(\eta)$ and $\psi_1 = L_g V(\eta)$. It is important to note that similar to (9), even with this controller, if $\delta \equiv 0$, then the exponential convergence of the outputs is guaranteed given that the parameters of the robot are known.

III. Model Uncertainty

Unlike the previous controllers (3), (5), which required an accurate model of the robot, this section will introduce controllers which consider model uncertainty as well. Since the parameters are not perfectly known, the equation of motion, (1), computed with the given set of parameters will henceforth have $\hat{}$ over the symbols. Therefore, D_a, M_a, C_a, G_a, E_a represent the actual model of the robot, and $\hat{D}, \hat{M}, \hat{C}, \hat{G}, \hat{E}$ represent the assumed model of the robot.

It is a well known fact that the inertial parameters of a robot are affine in the EOM (see [13]). Therefore (1) can be restated as:

$$\mathrm{Y}(q,\dot{q},\ddot{q})\Theta = B\mathrm{T}, \tag{18}$$

$\mathrm{Y}(q,\dot{q},\ddot{q})$ is the regressor [13], and Θ is the set of base inertial parameters. If Θ is the minimal representation of the parameters, then it is called the Base Parameter Set (BPS) [16].

Estimating Θ. Given the angle (q), velocity ($\dot{q}$) and acceleration ($\ddot{q}$) for the applied torque input T, it is possible to estimate Θ of length d by the following method:

$$\begin{aligned} \underset{\Theta\in\mathbb{R}^d}{\text{argmin}} \;\; & \Theta^T\Theta \\ \text{s.t.} \;\; & \mathrm{Y}(q,\dot{q},\ddot{q}) \;\Theta = \; B\mathrm{T} \;. \end{aligned} \tag{19}$$

If s samples are used instead of one, then Θ can more accurately approximate the actual set of base parameters Θ_a. Therefore, the following optimization problem is proposed:

$$\begin{aligned} \underset{\Theta\in\mathbb{R}^d}{\text{argmin}} \;\; & \Theta^T\Theta \\ \text{s.t.} \;\; & \underbrace{\begin{bmatrix} \mathrm{Y}(q_1,\dot{q}_1,\ddot{q}_1) \\ \mathrm{Y}(q_2,\dot{q}_2,\ddot{q}_2) \\ \vdots \\ \mathrm{Y}(q_s,\dot{q}_s,\ddot{q}_s) \end{bmatrix}}_{\mathrm{Y_C}} \Theta = \underbrace{\begin{bmatrix} B\mathrm{T}_1 \\ B\mathrm{T}_2 \\ \vdots \\ B\mathrm{T}_s \end{bmatrix}}_{\mathrm{T}_C}, \end{aligned} \tag{20}$$

with $\mathrm{Y_C}$ denoting the collection of s regressor matrices such that the rank of $\mathrm{Y_C}$ is not less than s, and $\mathrm{T_C}$ denotes the collection of torque vectors. The following Lemma will state the conditions required for computing the actual set of parameters Θ_a from the optimization problem (20).

Lemma 1: *Using the optimization problem presented in* (20)*, if the number of independent samples s^* is sufficient such that the rank $\mathscr{R}(\mathrm{Y_C}) \geq d(\Theta)$, with $d(\Theta)$ denoting the size (dimension) of Θ, then $\Theta = \Theta_a$. In other words, the actual set of parameters can be computed as:*

$$\Theta_a = (\mathrm{Y_C}^T \mathrm{Y_C})^{-1} \mathrm{Y_C}^T \mathrm{T}_C. \tag{21}$$

Control with Model Uncertainty. We will now define model based controllers for robotic systems with uncertain parameters. Feedback linearization can be implemented as:

$$\mathrm{T_{fbU}} = \hat{A}^{-1}(q,\dot{q}) \left(-\hat{L}_f^2 y(q,\dot{q}) + \mu\right), \tag{22}$$

where $\hat{A}$ and $\hat{L}_f^2$ can be obtained from the estimated parameters $\hat{\Theta}$. Similarly, the method of computed torque can be implemented as:

$$\begin{aligned} \mathrm{T_{ctU}} &= (\hat{D}(q) + \hat{M})\ddot{q}_d + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) + \hat{E}\dot{q}, \\ &= \mathrm{Y}(q,\dot{q},\ddot{q}_d)\hat{\Theta}. \end{aligned} \tag{23}$$

Again, if $k = n$, then $\mathrm{T_{fbU}} = \mathrm{T_{ctU}}$ (see [11]). Therefore, we will only consider the method of computed torque which can be conveniently evaluated through the regressor. The following lemma will show how the dynamics of the outputs evolve with the assumed set of parameters $\hat{\Theta}$.

Lemma 2: *Assume that the inertia matrix $(\hat{D}(q) + \hat{M})$ is bounded and invertible. If $k = n$, then the output dynamics $\ddot{y}$ will evolve as:*

$$\ddot{y} = \mu + J(\hat{D}(q) + \hat{M})^{-1} \mathrm{Y}(q,\dot{q},\ddot{q})(\hat{\Theta} - \Theta_a) \tag{24}$$

With the method of computed torque shown in (23), and with the minimum number of independent samples s^* achieved according to Lemma 1, we have:

$$\mathrm{T_{ctU}} = \mathrm{Y}(q,\dot{q},\ddot{q})\hat{\Theta} = \mathrm{Y}(q,\dot{q},\ddot{q})\Theta_a \implies \hat{\Theta} = \Theta_a. \tag{25}$$

Therefore, if $k = n$, then according to Lemma 2, $\ddot{y} = \mu$, which is an important observation. But, it is not always possible to get s^* samples which make $\mathrm{Y_C}^T \mathrm{Y_C}$ invertible. Under such conditions, we will consider the following theorem:

Theorem 1: *There exists p^* independent samples of $q, \dot{q}$ and $\ddot{q}$ such that the optimization problem* (20) *provides $\Theta = \Theta^*$ such that*

$$\mathrm{Y}(q,\dot{q},\ddot{q})\Theta^* = \mathrm{Y}(q,\dot{q},\ddot{q})\Theta_a \tag{26}$$

This theorem is aligned with the concept of persistence of excitation ([8]). If $p^* \geq s^*$, then the controller used is considered persistently exciting.

Theorem 1 can be utilized to rewrite (24). Specifically, the term $Y(q,\dot{q},\ddot{q})\Theta_a$ can be replaced with $Y(q,\dot{q},\ddot{q})\Theta^*$, yielding:

$$\ddot{y} = \mu + J(\hat{D}(q) + \hat{M})^{-1} \mathrm{Y}(q,\dot{q},\ddot{q})(\hat{\Theta} - \Theta^*). \tag{27}$$

This is significant because it means that it is not necessary to know the actual model of the robot to have the desired dynamics, and the sample space required to determine Θ^* depends on the dynamics itself. This property can be used to frame the Control Lyapunov Function for the uncertain model in such a way that the knowledge of Θ_a can be entirely eliminated.

Control Lyapunov Function with Parameter Uncertainty. Define:

$$\Phi = (\hat{D}(q) + \hat{M})^{-1} \mathrm{Y}(q,\dot{q},\ddot{q}). \tag{28}$$

With p^* samples, we have $\Phi\Theta^* = \Phi\Theta_a$ according to Theorem 1. Also declare the following variables:

$$\tilde{\Theta} = \hat{\Theta} - \Theta_a, \;\; \tilde{\Theta}^* = \hat{\Theta} - \Theta^* \tag{29}$$

and the control law for the assumed set of base parameters:

$$\dot{\hat{\Theta}} = \beta. \tag{30}$$

From (29) it is also evident that $\dot{\tilde{\Theta}}^* = \dot{\hat{\Theta}} = \dot{\tilde{\Theta}}$. Here $\tilde{\Theta}$ is not known, but $\tilde{\Theta}^*$ is known. With these substitutions, and using (4) and (27), we have the following state space representation for the uncertain model:

$$\begin{aligned} \dot{\eta} &= F\eta + G\mu + GJ\Phi\tilde{\Theta}^* \\ \dot{\tilde{\Theta}}^* &= \beta, \end{aligned} \tag{31}$$

where F and G are obtained from (4). For this representation, we could apply a simple control law:

$$\beta = -\varepsilon\tilde{\Theta}^*, \; \mu = -2\varepsilon L_f y(q,\dot{q}) - \varepsilon^2 y(q)$$

in order to drive $\eta \to 0$. We could also pick μ, β by using Control Lyapunov Functions. In this case, consider a Lyapunov candidate for the uncertain model:

$$V_{\mathrm{U}}(\eta, \tilde{\Theta}^*) = \eta^T P \eta + \tilde{\Theta}^{*T} \Gamma \tilde{\Theta}^*, \tag{32}$$

with P the solution to CARE given in (12), and Γ a positive definite matrix. Taking the derivative of V_{U}, substituting for $\dot{\eta}$ and $\dot{\tilde{\Theta}}^*$, and picking $\kappa = 2\eta^T PGJ\Phi\tilde{\Theta}$ yields:

$$\begin{aligned} \dot{V}_{\mathrm{U}}(\eta, \tilde{\Theta}^*, \mu, \beta) &= \dot{\eta}^T P \eta + \eta^T P \dot{\eta} + 2\tilde{\Theta}^{*T} \Gamma \dot{\tilde{\Theta}}^*, \\ &= \eta^T (F^T P + PF)\eta + 2\eta^T PG\mu \\ &\quad \cdots + \kappa + 2\tilde{\Theta}^{*T} \Gamma \beta. \end{aligned} \tag{33}$$

With this CLF candidate, we can realize an optimal controller which will render the Lyapunov function V_{U} exponentially convergent by imposing the constraint:

$$\dot{V}_{\mathrm{U}} \leq -\gamma V_{\mathrm{U}}, \tag{34}$$

where γ is given by (16). In other words, we can design the following optimal controller:

$$\begin{aligned} &\underset{(\mu,\beta) \in \mathbb{R}^{n+d}}{\mathrm{argmin}} \;\; \mu^T \mu + \beta^T \beta \\ &\text{s.t. } L_F V_{\mathrm{U}} + L_G V_{\mathrm{U}} \mu + L_\Theta V_{\mathrm{U}} \beta \leq 0, \end{aligned} \tag{35}$$

where

$$\begin{aligned} L_F V_{\mathrm{U}}(\eta, \tilde{\Theta}^*) &= \eta^T (F^T P + PF)\eta + \kappa + \gamma V_{\mathrm{U}}(\eta, \tilde{\Theta}^*), \\ L_G V_{\mathrm{U}}(\eta, \tilde{\Theta}^*) &= 2\eta^T PG, \\ L_\Theta V_{\mathrm{U}}(\eta, \tilde{\Theta}^*) &= 2\tilde{\Theta}^{*T} \Gamma. \end{aligned} \tag{36}$$

The dimension of μ is $k = n$ since full actuation needs to be assumed to utilize (27).

It is important to note that the term κ is a function of $\ddot{q}$. Therefore, the term $\ddot{q}$ appears in $\dot{V}_{\mathrm{U}}$. In particular, $\ddot{q}$ appears on the right hand side of (27). So the state space for the given system $(f(x), g(x))$ has an algebraic loop which needs to be considered while simulating this controller. One way to do it is by specifying μ separately and substituting into (27). By picking:

$$\mu = -G^T P \eta, \tag{37}$$

we obtain the following simplification to $\dot{V}_{\mathrm{U}}$:

$$\dot{V}_{\mathrm{U}}(\eta, \tilde{\Theta}^*, \mu, \beta) = -\eta^T Q \eta + \kappa + 2\tilde{\Theta}^{*T} \Gamma \beta, \tag{38}$$

where Q is obtained from the CARE (12). μ is used to evaluate (31) which gives the $\ddot{q}$ required to evaluate κ.

The minimum norm controller for the parameters can thus be formulated as:

$$\begin{aligned} &\underset{\beta \in \mathbb{R}^d}{\operatorname{argmin}} \ \beta^T \beta \qquad (39) \\ &\text{s.t.} \ -\eta^T Q \eta + \kappa + 2\tilde{\Theta}^{*T} \Gamma \beta \leq -\gamma V_{\mathrm{U}}, \end{aligned}$$

where κ is known, $\tilde{\Theta}^*$ is also known. Therefore β can be computed accordingly in every iteration. With this controller we can introduce the following theorem:

Theorem 2: *Let μ be chosen as in* (38)*, and utilized in the computed torque controller,* $\mathrm{T}_{\mathrm{ctU}}$*, or the feedback linearization controller,* $\mathrm{T}_{\mathrm{fbU}}$*, as given in* (23) *and* (22) *with parameters determined by the quadratic program* (39)*. Then the control Lyapunov function,* V_{U}*, given in* (32) *is an exponentially stabilizing Control Lyapunov Function. In particular,*

$$\begin{aligned} &c_1 \|(\eta, \tilde{\Theta}^*)\|^2 \leq V_{\mathrm{U}}(\eta, \tilde{\Theta}^*) \leq c_2 \|(\eta, \tilde{\Theta}^*)\|^2 \\ &\inf_{(\mu,\beta) \in \mathbb{R}^{n+d}} [\dot{V}_{\mathrm{U}}(\eta, \tilde{\Theta}^*, \mu, \beta) + c_3 V_{\mathrm{U}}(\eta, \tilde{\Theta}^*)] \leq 0 \qquad (40) \end{aligned}$$

for

$$\begin{aligned} c_1 &= \min\{\lambda_{\min}(P), \lambda_{\min}(\Gamma)\}, \\ c_2 &= \max\{\lambda_{\max}(P), \lambda_{\max}(\Gamma)\}, \\ c_3 &= \gamma \end{aligned}$$

Due to space constraints the proof is omitted, but the result is established by finding a specific value of μ and β. We can choose $\mu = -G^T P \eta$, as given in (37), and

$$\beta = -\Gamma^{-1} \Phi^T J^T G^T P \eta - \gamma \tilde{\Theta}^*, \tag{41}$$

and evaluate (39) which satisfies (40).

IV. Simulation Results

We consider a 5 link robot manipulator Fig. 1. It specifically originates from the 5 link robot, AMBER, used to achieve walking via human-inspired control (see [15]). Since only continuous time systems are considered in this paper, we will not consider impact dynamics, and therefore not consider locomotion. Instead, we will consider the control objective of stabilizing the robot upright in the presence of uncertain parameters. In other words, the objective is to drive all the joint angles and angular velocities to zero.

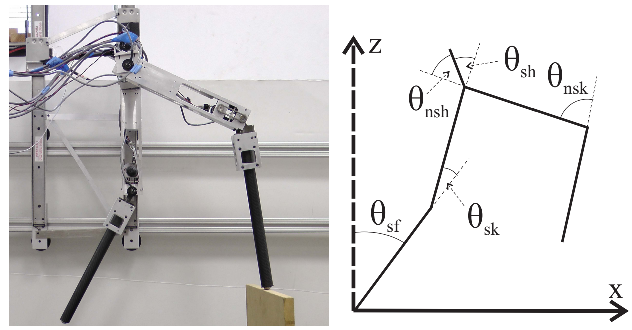

Fig. 1: The biped AMBER (left) and the stick figure of AMBER showing the configuration angles (right).

We can define the configuration space : $q = (q_{sf}, q_{sk}, q_{sh}, q_{nsh}, q_{nsk})^T \in \mathbb{R}^5$, which is the vector of stance ankle, stance knee, stance hip, non-stance hip and non-stance knee angles in order. Full actuation ($n = k$) is assumed for this robot.

Outputs. For this robot, the outputs y are defined in the following manner: joint angles of the robot are the actual outputs, i.e., $y_a = q$. Since the objective here is to drive the robot to a zero angle configuration, the desired output, y_d, is zero. Therefore, the outputs become:

$$\eta = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}, \tag{42}$$

and the objective is to drive the output η to zero, i.e., $\eta \to 0$.

We will consider the vector of parameters $\hat{\Theta}$ which contain the masses and inertia of each link, inertia of the motors and damping friction. The left hand side of (1) can be rewritten such that the unknown variables can be linearly separated from the equations. The set of base parameters Θ^* is computed with p^* samples as indicated in Theorem 1. p^* depends on the size of Θ^*. 58 parameters are considered here to verify convergence of the controller. It is found that $p^* = 5$ for the robot manipulator considered here. Accordingly, the state variables, η and $\hat{\Theta} - \Theta^*$, form the state space representation given in (31). These states are simulated in MATLAB with the controller given by (37) for evaluating μ and (39) for evaluating β, and the following results are obtained.

Parameters $\hat{\Theta}$ are started with a 40% error and convergence to Θ^* is observed. Plots of the actual and desired outputs, y_a and y_d, are shown in Fig. 2. Plots of the parameter error, $\tilde{\Theta}^*$, and the Lyapunov function, V_{U} are shown in Fig. 3. The tiles of the robot at different instances of time until it reaches zero angle configuration is shown in Fig. 4. Hyper-extension of the knee joints of the robot is observed in the tiles, but is ignored since the objective here is to achieve exponential convergence in tracking under parameter uncertainty.

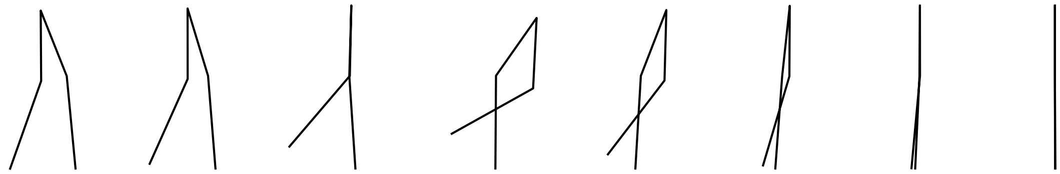

Fig. 4: Tiles of the robot showing different configurations over time until the angles converge to zero.

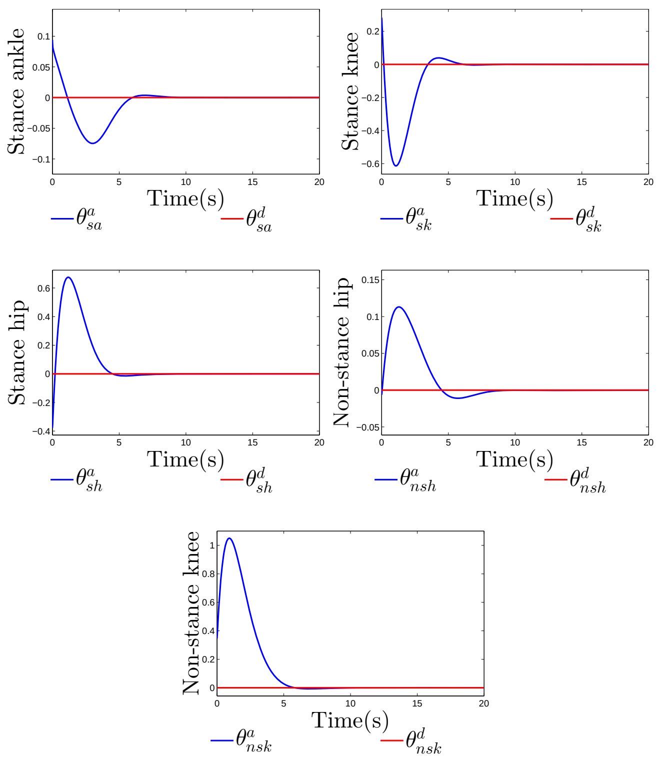

Fig. 2: Figures showing the comparison between actual (blue) and desired (red) joint angles over time. The joint angles are seen to be converging to zero.

V. Conclusions

The paper presents a method for achieving exponential convergence under parameter uncertainty by using a unified CLF controller, both in theory and simulation. This is achieved by using the property that the parameters are affine in the dynamics. It is also shown that the parameters do not have to converge to the actual set of base inertial parameters Θ_a to realize exponential convergence, and the parameter set obtained is primarily decided by the robot design and the state space in which the robot manipulator is operating. One major drawback with this method is that it uses acceleration data which is noisy and could lead to bad parameter estimation.

References

[1] Aaron D Ames, Kevin Galloway, JW Grizzle, and Koushil Sreenath. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Trans. on Automatic Control*, 2014.

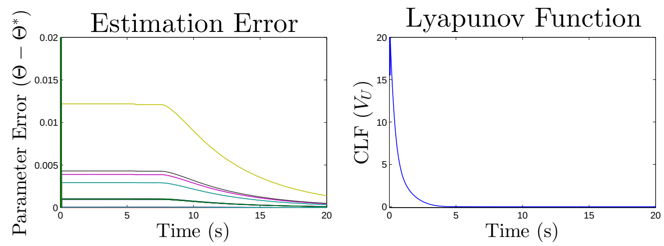

Fig. 3: The parameter errors (left) and the CLF (right) are shown here. The CLF is converging exponentially to zero.

[2] Aaron D Ames and Matthew Powell. Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs. In *Control of Cyber-Physical Systems*, pages 219–240. Springer International Publishing, 2013.

[3] Chae H An, Christopher G Atkeson, and John M Hollerbach. *Model-based control of a robot manipulator*, volume 16. MIT press Cambridge, MA, 1988.

[4] Randy A Freeman and Petar V Kokotovic. *Robust nonlinear control design: state-space and Lyapunov techniques*. Springer, 2008.

[5] Kevin Galloway, Koushil Sreenath, Aaron D Ames, and Jessy W Grizzle. Torque saturation in bipedal robotic walking through control lyapunov function based quadratic programs. *arXiv preprint arXiv:1302.7314*, 2013.

[6] Fathi Ghorbel, John Y Hung, and Mark W Spong. Adaptive control of flexible-joint manipulators. *Control Systems Magazine, IEEE*, 9(7):9–13, 1989.

[7] Roberto Horowitz, William Messner, and John B Moore. Exponential convergence of a learning controller for robot manipulators. *Automatic Control, IEEE Transactions on*, 36(7):890–894, 1991.

[8] John B Moore, Roberto Rorowitz, and William Messner. Functional persistence of excitation and observability. In *American Control Conference, 1990*, pages 308–314. IEEE, 1990.

[9] Romeo Ortega and Mark W Spong. Adaptive motion control of rigid robots: A tutorial. *Automatica*, 25(6):877–888, 1989.

[10] Shankar S Sastry. *Nonlinear Systems: Analysis, Stability and Control*. Springer, New York, 1999.

[11] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2008.

[12] Jean-Jacques E Slotine and Weiping Li. On the adaptive control of robot manipulators. *The International Journal of Robotics Research*, 6(3):49–59, 1987.

[13] Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*. John Wiley & Sons Hoboken^ eNJ NJ, 2006.

[14] Eric R Westervelt, Jessy W Grizzle, Christine Chevallereau, Jun H Choi, and Benjamin Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, Boca Raton, 2007.

[15] Shishir Nadubettu Yadukumar, Murali Pasupuleti, and Aaron D Ames. From formal methods to algorithmic implementation of human inspired control on bipedal robots. In *Tenth International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, Cambridge, MA, 2012.

[16] Koji Yoshida, Koichi Osuka, Hirokazu Mayeda, and Toshiro Ono. When is the set of base parameter values physically impossible? In *Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on*, volume 1, pages 335–342. IEEE, 1994.