Adaptive Backstepping Tuning Functions Control Design for Industrial Robot Manipulators

Tran Xuan Kien[1], Do Duc Hanh[2]

[1]R&D Department, Military Institute of Science and Technology, Hanoi, Vietnam
Email: txkien2003@gmail.com
[2]Control Engineering Department, Military Technical Academy, Hanoi, Vietnam
Email: hanh@aitcv.ac.vn

Abstract— *In this paper, an adaptive backstepping control design with tuning functions and K-filters for robot manipulators is developed. A stronger stability and convergence performance of the designed control in comparison with backstepping observer is achieved despite the presence of disturbances, parameter uncertainties, system nonlinearities for a real-time system of a single-link flexible-joint manipulator.*

Keywords— Tuning functions, K-filters, Adaptive Observer Backstepping, Robot Manipulator Control.

I. INTRODUCTION

The adaptive backstepping solution to the problem of nonlinear stabilization and tracking in the presence of unknown parameters is a starting point for more elaborate adaptive control designs for feedback systems including robot manipulators [2-4]. One of the improvements to be achieved with the tuning functions design [1] is to reduce the dynamic order of the adaptive controller to its minimum (the number of parameter estimates is equal to the number of unknown parameters). This minimum-order design is advantageous not only for implementation, but also because it guarantees the strongest achievable stability and convergence properties.

In the tuning functions procedure the parameter update law is designed recursively. At each consecutive step we design a tuning function as a potential update law. In contrast to adaptive backstepping in [2], these intermediate update laws are not implemented. Instead, the controller uses them to compensate for the effect of parameter estimation transients. Only the final tuning function is used as the parameter update law.

In this paper we presented an adaptive backstepping tuning functions control design for systems in the output feedback form. In the design, different filter structures (K-filter) and identifiers are applied.

The rest of the paper is structured as follows. A design with tuning functions for a single-link flexible-joint robot manipulator is presented in section II. Experiment design and the performance of the designed real-time control for the flexible-joint robot arm are presented in section III. We conclude in section IV.

II. TUNING FUNCTIONS FOR A SINGLE-LINK FLEXIBLE-JOINT ROBOT MANIPULATOR

2.1 System Modeling

We consider a single-link flexible-joint robot manipulator actuated by a DC motor as shown in Fig. 1 (This is depicted in [2] but repeated here for convenience).

The dynamic equations of the system are as follows:

$$
\begin{aligned}
&J_1\ddot{q}_1 + F_1\dot{q}_1 + K\left(q_1 - \frac{q_2}{N}\right) + mgd\cos q_1 = 0\\
&J_2\ddot{q}_1 + F_2\dot{q}_2 - \frac{K}{N}\left(q_1 - \frac{q_2}{N}\right) = K_t i\\
&LDi + Ri + K_b\dot{q}_2 = u
\end{aligned}
\tag{1}
$$

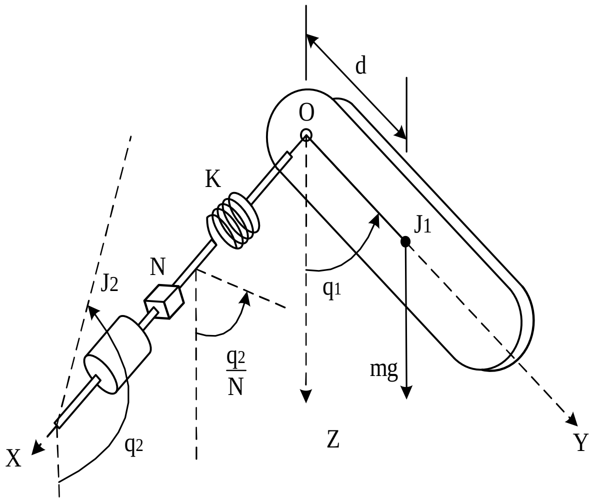

FIG. 1. A SINGLE-LINK FLEXIBLE ROBOT MANIPULATOR

Where q_1, q_2 are the angular positions of the link and the motor shaft, *i* is the armature current, and *u* is the armature voltage. The inertials J_1, J_2, the viscous friction constants F_1, F_2, the spring constant *K*, the torque constant K, the torque constant K_t, the back-emf constant K_b, the armature resistance *R* and inductance *L*, the link mass *M*, the position of link's centre of gravity *d*, the gear ratio *N* and the acceleration of gravity *g* can all be unknown.

We assume that only the link position q_1 is measured. The choice of state variables:

$$\zeta_1 = q_1 ,\ \zeta_2 = \dot{q}_1,\ \zeta_3 = q_2,\ \zeta_4 = \dot{q}_2,\ \zeta_5 = i$$

The dynamic equations of the system become:

$$\begin{aligned}
&\dot{\zeta}_1 = \zeta_2 \\
&\dot{\zeta}_2 = -\frac{mgd}{J_1}\cos y - \frac{F_1}{J_1}\zeta_2 - \frac{K}{J_1}\left(\zeta_1 - \frac{\zeta_2}{N}\right) \\
&\dot{\zeta}_3 = \zeta_4 \\
&\dot{\zeta}_4 = -\frac{K}{J_2 N}\left(\zeta_1 - \frac{\zeta_2}{N}\right) - \frac{F_2}{J_2}\zeta_4 + \frac{K_t}{J_2}\zeta_5 \\
&\dot{\zeta}_5 = -\frac{R}{L}\zeta_5 - \frac{K_b}{L}\zeta_4 - \frac{1}{L}u \\
&y = \zeta_1
\end{aligned} \tag{2}$$

Clearly, (2) is not in the output-feedback. Differentiating y twice, we obtain $\zeta_2 = Dy$ ($D = {}^{d}\!/_{dt}$ is the differentiation operator) and

$$D^2 y = -\frac{mgd}{J_1}\cos y - \frac{F_1}{J_1}Dy - \frac{K}{J_1}\left(y - \frac{\zeta_3}{N}\right)$$

It implies that:

$$\zeta_3 = \frac{J_1 N}{K}\left(D^2 y + \frac{mgd}{J_1}\cos y + \frac{F_1}{J_1}Dy - \frac{K}{J_1}y\right) \zeta_4 = D\zeta_3 = \frac{J_1 N}{K}\left(D^3 y + \frac{mgd}{J_1}D\cos y + \frac{F_1}{J_1}D^2 y + \frac{K}{J_1}Dy\right)$$

Differentiating and substituting ζ_3, ζ_4 , we obtain

$$\zeta_5=\frac{J_1J_2N}{K_1K}\left(D^4y+\left(\frac{F_1}{J_1}+\frac{F_2}{J_2}\right)D^3y+\left(\frac{K}{J_1}+\frac{K}{J_2N^2}+\frac{F_1F_2}{J_1J_2}\right)\right)D^2y+$$
$$+\frac{mgd}{J_1}D^2\cos y+\left(\frac{F_1K}{J_1J_2N^2}+\frac{F_2K}{J_1J_2}\right)Dy\ +\frac{mgd}{J_1J_2}D\cos y+\frac{mgdK}{J_1J_2N^2}\cos y$$

Finally, differentiating and substituting ζ_5,ζ_4 we arrive at the input-output description

$$\begin{aligned}D^5y=&\frac{K_tK}{J_1J_2NL}u-\left(\frac{R}{L}+\frac{F_1}{J_1}+\frac{F_2}{J_2}\right)D^4y-\frac{mgd}{J_1}D^3\cos y+\left[\frac{R}{L}\left(\frac{F_1}{J_1}+\frac{F_2}{J_2}\right)+\frac{K_tK_b}{J_2L}+\left(\frac{K}{J_1}+\frac{K}{J_2N^2}+\frac{F_1F_2}{J_1J_2}\right)\right]D^3y\\&+\left[\frac{R}{L}\left(\frac{K}{J_1}+\frac{K}{J_2N^2}+\frac{F_1F_2}{J_1J_2}\right)+\frac{F_1K}{J_1J_2N^2}+\frac{F_2K}{J_1J_2}+\frac{K_tK_bF_1}{J_1J_2L}\right]D^2y-\left(\frac{R}{L}+\frac{F_2}{J_2}\right)\frac{mgd}{J_1}D^2\cos y-\frac{R}{L}\left(\frac{F_1K}{J_1J_2N^2}+\frac{F_2K}{J_1J_2}\right)+\frac{K_tK_b}{J_1J_2L}\Bigg]Dy\\&+\left(\frac{K}{N^2}+\frac{RF_2}{L}+\frac{K_tK_b}{L}\right)\frac{mgd}{J_1}D\cos y-\frac{R}{L}\frac{mgd}{J_1J_2N^2}\cos y\end{aligned}\tag{3}$$

It is tedious but straightforward using (3) to find a choice of state variables.

$$\begin{aligned}\dot{x}_1&=x_2+a_1.y\\\dot{x}_2&=x_3+a_{2.}y+a_{3.}.\cos y\\\dot{x}_3&=x_4+a_{4.}y+a_{5.}.\cos y\\\dot{x}_4&=x_5+a_{6.}y+a_{7.}.\cos y\\\dot{x}_5&=b_0.u+a_{8.}.\cos y\\y&=x_1\end{aligned}\tag{4}$$

Where the unknown parameters a_1 , a_2 , a_3 ,.....a_8 are defined as:

$$\begin{aligned}&a_1=-\left(\frac{R}{L}+\frac{F_1}{J_1}+\frac{F_2}{J_2}\right);a_3=-\frac{mgd}{J_1};b_0=\frac{K_tK}{J_1J_2NL};a_2=-\left[\frac{R}{L}\left(\frac{F_1}{J_1}+\frac{F_2}{J_2}\right)+\frac{K_tK_b}{J_2L}+\left(\frac{K_1}{J_1}+\frac{K}{J_2N^2}+\frac{F_1F_2}{J_1J_2}\right)\right]\\&a_4=-\left[\frac{R}{L}\left(\frac{K}{J_1}+\frac{K}{J_2N^2}+\frac{F_1F_2}{J_1J_2}\right)+\frac{F_1K}{J_1J_2N^2}+\frac{F_2K}{J_1J_2}+\frac{K_bK_tF_1}{J_1J_2L}\right];a_5=-\left(\frac{R}{L}+\frac{F_2}{J_2}\right)\frac{mgd}{J_1};\\&a_6=-\left[\frac{R}{L}\left(\frac{F_1K}{J_1J_2N^2}+\frac{F_2K}{J_1J_2}\right)+\frac{K_tK_b}{J_1J_2L}\right];a_7=-\left(\frac{K}{N^2}+\frac{RF_2}{L}+\frac{K_tK_b}{L}\right)\frac{mgd}{J_1J_2};a_8=\frac{R}{L}\frac{mgd}{J_1J_2N^2}.\end{aligned}\tag{5}$$

Hence, the design procedure of theorem is applicable to (4), and an adaptive controller that achieves bounded asymptotic position tracking from all initial conditions and for all position values of the constant *J₁, J₂, F₁, F₂ K, Kₜ K_b, R, L.*

We consider systems in the output-feedback form:

$$\begin{aligned}\dot{x}_1&=x_2+\varphi_{0,1}(y)+\sum_{j=1}^{q}a_j\varphi_{j,1}(y)\\\dot{x}_2&=x_3+\varphi_{0,2}(y)+\sum_{j=1}^{q}a_j\varphi_{j,2}(y)\\\dot{x}_{p-1}&=x_p+\varphi_{0,p-1}(y)+\sum_{j=1}^{q}a_j\varphi_{j,p-1}\\\dot{x}_p&=x_{p+1}+\varphi_{0,p+1}(y)+\sum_{j=1}^{q}a_j\varphi_{j,\mathrm{p}}(y)+b_m\pm\ (y)u\\\dot{x}_n&=\varphi_{0,n}(y)+\sum_{j=1}^{q}a_j\varphi_{j,n}(y)+b_0\pm\ (y)u\\y&=x_1\end{aligned}\tag{6}$$

$\varphi_{0,1}(y)=0\,;\varphi_{0,2}(y)=0\,;\varphi_{0,3}(y)=0\,;\varphi_{0,4}(y)=0\,;\varphi_{0,5}(y)=0\,;\qquad q=8;\, n=p=5;\, m=0;$

$\varphi_{i,1}(y)=[\mathrm{y}\ 0\ 0\ 0\ 0\ 0\ 0\ 0]\,;\varphi_{i,2}(y)=[0\ \mathrm{y}\ \cos y\ 0\ 0\ 0\ 0\ 0]\,;\varphi_{i,3}(y)=[0\ 0\ 0\ \mathrm{y}\ \cos y\ 0\ 0\ 0]\,;$
$\varphi_{i,4}(y)=[0\ 0\ 0\ 0\ 0\ \mathrm{y}\ \cos y\ 0]\,;\varphi_{i,5}(y)=[0\ 0\ 0\ 0\ 0\ 0\ 0\ \cos y]\,;$

Where xϵR is the state, uϵR is the input, yϵR is the output, φ, σ are smooth nonlinear functions, and

$$a=[a_1,a_2\ ......a_7,a_8];\quad b=b_0 \tag{7}$$

are vectors of unknown constant parameters. Only the output y is available for measurement. We rewrite as

$$\begin{aligned}\dot{x}&=Ax+\Phi(y)a+b_0u\\ y&=e_1^{\,T}x\end{aligned} \tag{8}$$

$$A=\begin{bmatrix}0 & & \\ \vdots & & I_{(4)x(4)}\\ 0 & \dots & 0\end{bmatrix},\ \Phi(y)=\begin{bmatrix}\ddot{}_{1,1}(y) & \dots & \ddot{}_{8,1}(y)\\ \dots & \dots & \dots\\ \ddot{}_{1,5}(y) & \dots & \ddot{}_{8,5}(y)\end{bmatrix},\ c=1. \tag{9}$$

Filters and observer

We start by rewriting (8) as

$$\begin{aligned}\dot{x}&=Ax+\Phi(y)a+b_0u\\ y&=e_1^{\,T}x\end{aligned} \tag{10}$$

Where the $p = q+m+1 = 9$ dimensional parameter vector is defined by $\quad -=\begin{bmatrix}b_0\\ a_{8x1}\end{bmatrix}$ (11)

And $\quad F(y,u)^T=\begin{bmatrix}\underbrace{u}_{row1},\underbrace{\Phi(y)}_{row2}\end{bmatrix}$ (12)

If $-$ were known, we would design an observer $\quad \dot{\hat{x}}=A_0\hat{x}+ky+F(y,u)^{T-}$ (13)

With the vector $k=[k_1,k_2,....k_n]^T$ chosen so that the matrix $\quad A_0=A-ke_1^T$

By Hurwitz, that is $PA_0+A_0^TP=-I,\quad P=P^T>0$

Then, the observer error $\tilde{x}=x-\hat{x}$ would be governed by the exponentially stable system $\dot{\tilde{x}}=A_0\tilde{x}$

Since $-$ is not known, the observer is not implementable but it provides motivation for the subsequent development. We define the state estimate

$$\tilde{x}=\P+\Omega^{T\hat{-}} \tag{14}$$

Which employs the filters $\quad \begin{aligned}\dot{\P}&=A_0\P+ky\\ \dot{\Omega}^T&=A_0\Omega^T+F(y,u)^T\end{aligned}$ (15)

The state estimate error $\quad £=x-\hat{x}$ (16)

is readily shown to satisfy $£=A.£$

The nonminimal observer (14), (15) is still nonimplementable because it depends on ¯ . However, it has a key property not present in (13): the state x satisfies a static relationship with ¯ , that is

$$x = \text{¶} + \Omega^{T\text{¯}} + \text{£}$$

This is easily verified by substituting (14) into (16).

Remark 1. The certainty equivalence counterpart of the estimate (14) is

$$\dot{x} = \text{¶} + \Omega^{T\hat{\text{¯}}} \tag{17}$$

it can alternatively be generated via

$$\dot{\tilde{x}} = A_0\tilde{x} + ky + F(y,u)^{T\hat{\text{¯}}} + \Omega^{T\dot{\hat{\text{¯}}}} \tag{18}$$

This observer is not a certainty equivalence version of (13) because of the term $\Omega^{T\dot{\hat{\text{¯}}}}$.

To reduce the dynamic order of the filter (15), we exploit the structure of *F(y,u)*. Denote the first m+1 columns of Ω^T by υ_0 satisfy the equations

$$\dot{\text{³}}_0 = A_0\text{³}_0 + e_5 u \tag{19}$$

It is easy to show that

$$A_0 e_5 = e_5$$

Therefore, the vectors are generated by only one input filter

$$\dot{\text{ª}} = A_0\text{ª} + e_5 u \tag{20}$$

with the algebraic expressions $\text{³}_j = A_0^j\text{ª} \quad , j = 0,...5$

While we always implement the filter (20), for analysis we use the equations (19) considering (12), (15), (20). Ω is obtained as

$$\Omega^T = \left[\text{³}_5, ..., \text{³}_1, \text{³}_0, \Xi\right] \tag{21}$$

where the matrix is generated by

$$\dot{\Xi} = A_0\Xi + \Phi(y) \quad , \Xi \in \Re^{nxq} \tag{22}$$

TABLE 1

K-filter	$\dot{\text{¶}} = A_0\text{¶} + ky$ $\dot{\Xi} = A_0\Xi + \Phi(y) \quad , \Xi \in \Re^{nxq}$ $\dot{\text{ª}} = A_0\text{ª} + e_5 u$	(23)
	$\text{³}_j = A_0^j\text{ª} \quad , j = 0,....,m$ $\Omega^T = \left[\text{³}_{0,j}, \Xi\right]$	(24)

The implemented filters are summarized in Table 1. The total dynamic order of the K-filters is *nx(q+2)*. As explained in [1], a further reduction is possible by using the reduced-order observer technique, so that the total filter dynamic order becomes *(n-1)(q+2)*.

To prepare for the backstepping procedure in the next subsection, we consider the equation for the output rewritten from (6)

$$\dot{y} = x_2 + \Phi_{(1)} a \tag{25}$$

we need to replace the unavailable state x_2 by available filter signals. We have

$$x_2 = ¶_2 + \Omega_2^{T}¯ + £_2 \quad = ¶_2 + \left[³_{0,2}, \Xi_{(2)} \right]¯ + £_2 \quad = b_0 .³_{0,2} + ¶_2 + \left[0, \Xi_{(2)} \right]¯ + £_2 \tag{26}$$

we obtain the following two important expressions for y

$$\dot{y} = µ_0 + µ^{T}¯ + £_2 \quad = µ_0 + \bar{µ}^{T}¯ + £_2 \tag{27}$$

where the regressor and truncated regressor are defined as

$$µ = \left[³_{0,2}, \Xi_{(2)} \right]; \quad \bar{µ} = \left[0, \Xi_{(2)} \right] \tag{28}$$

and $\ddot{}_0 = ¶_2 + £_2$

2.2 Adaptive controller design with tuning functions

The tuning functions design with K-filters has many similarities with the state-feedback design presented in section 4.2 in [1]. It also uses the same technique for dealing with unknown high frequency gain b_m as in section 4.5 in [1].

The first obstacle for applying backstepping with output feedback is that the state x_2 is not measured. For this reason, (27) is written in a form which suggests that the filter signal be used for backstepping. Indeed, by comparing (6) with

$$ʒ_m = A_0 ³_m + e_5 ± u \tag{29}$$

we see that both x_2 and $³_{m,2}$ are separated from the control u by 4 integrators. Therefore, the system to which we apply backstepping is

$$\begin{aligned} \dot{y} &= µ_0 + \bar{µ}^{T}¯ + £_2 \\ ʒ_{0,i} &= ³_{m,i+1} - k_i ³_{0,1} \qquad i = 2, \ldots, 4. \\ ʒ_{0,5} &= b_0 u + ³_{0,5} - k_{\circ} ³_{0,1} \end{aligned} \tag{30}$$

Our analysis will show that once we stabilize the system (30), all the close-loop signals remain bounded. In the backstepping procedure while multiplying by nonlinear terms, we will include nonlinear damping terms in our stabilizing functions.

Next we need to choose the stabilizing functions and the tuning functions to achieve a design skew-symmetric form.

For system (30) we change coordinates

$$\begin{aligned} z_1 &= y - y_r \\ z_i &= ³_{0,i} - \hat{S} y_r^{(i-1)} - \sim_{i-1} \qquad i = 2, \ldots 4 \end{aligned} \tag{31}$$

where $\hat{S}$ is an estimate of $S = 1/b_0$

The complete tuning function design with K-filters is summarized in Table 2, where the filters given in Table 1 are employed.

TABLE 2

$z_1 = y - y_r$ $z_i = {}^{3}{}_{m,i} - \hat{s}\,y_r^{(i-1)} - \sim_{i-1} \qquad i = 2,\ldots 4$	(32)
$\sim_1 = \bar{s}\,\bar{\sim}_1$ $\bar{\sim}_1 = -c_1 z_1 - d_1 z_1 - \mu_0 - \bar{\mu}^{T-}$ $\sim_2 = -\hat{b}_0 z_1 - \left\|-c_2 + d_2\left(\frac{\partial\sim_{i-1}}{\partial y}\right)^2 + k_2\left\|\frac{\partial\sim_1}{\partial y}\mu\ \ - z_1 e_1\right\|^2\right\| z_2 + (\dot{y}_r + \frac{\partial\sim_1}{\partial\hat{s}})\dot{\hat{s}} - \frac{\partial\sim_1}{\partial y}\Gamma^{2}{}_2 + \ {}_2$ $\sim_i = -z_{i-1} - c_i z_i - d_i\left(\frac{\partial\sim_{i-1}}{\partial y}\right)^2 z_i - k_i\left\|\frac{\partial\sim_1}{\partial y}\mu\ \ \right\|^2 z_i + z_{i+1} - \sum_{j=i+1}^{\circ}\frac{\partial\sim_{i-1}}{\partial\hat{-}}\Gamma\frac{\partial\sim_{i-1}}{\partial y}\mu z_j + \frac{\partial\sim_{i-1}}{\partial y}\Gamma^{2}{}_i + \ {}_i$ ${}_i = \frac{\partial\sim_{i-1}}{\partial y}(\mu_0 + \mu^{T\hat{-}}) + \frac{\partial\sim_{i-1}}{\partial\P}(A_0\P + ky) + \frac{\partial\sim_{i-1}}{\partial\Xi}(A_0\Xi + \Phi(y)) + \sum_{j=1}^{i-1}\frac{\partial\sim_{i-1}}{\partial y_r^{(j-1)}}y_r^{(j)}$ $+\sum_{j=1}^{m+i-1}\frac{\partial\sim_{i-1}}{\partial \mathrm{a}_j}(-k_j\mathrm{a}_j + \mathrm{a}_{j+1})$	(33)
${}^{2}{}_1 = (\mu\ \ - \hat{s}(\dot{y}_r + \sim_1)e_1^T)z_1$ ${}^{2}{}_i = {}^{2}{}_{i-1} - \frac{\partial\sim_{i-1}}{\partial y}\mu z_i$	(34)
Adaptive control law: $u = \frac{1}{\pm(y)}\left(\sim_4 - {}^{3}{}_{0,5} + \mathrm{s}\,y_r^{(5)}\right)$	(35)
Parameter update laws: $\dot{\bar{-}} = -\Gamma W_{-}(z,t)z$ $\dot{\hat{s}} = ¥\ \mathrm{sgn}(b_0)(\dot{y}_r + \bar{\sim}_1)e_1^T z$	(36)

III. EXPERIMENT DESIGN

3.1 Experiment and simulation setup

A 2-DOF flexible-joint robot arm control is implemented based on algorithms shown in section II. The control algorithms are modelled in Simulink, run in real-time under Real Time Window Target utility of Real Time Workshop, and interface through a PCI1711 card from Advantech with the real world robot arm (see Fig. 2).

The model (constructed the same as in [2] and we repeat here for convenience) consisted of MATLAB/Simulink algorithm, PCI1711 card, DC motor with PWM driver and IMU. Torque transmission between the actuator (DC motor) and the plant (industrial robot arm) is of spring drive as a flexible-joint.

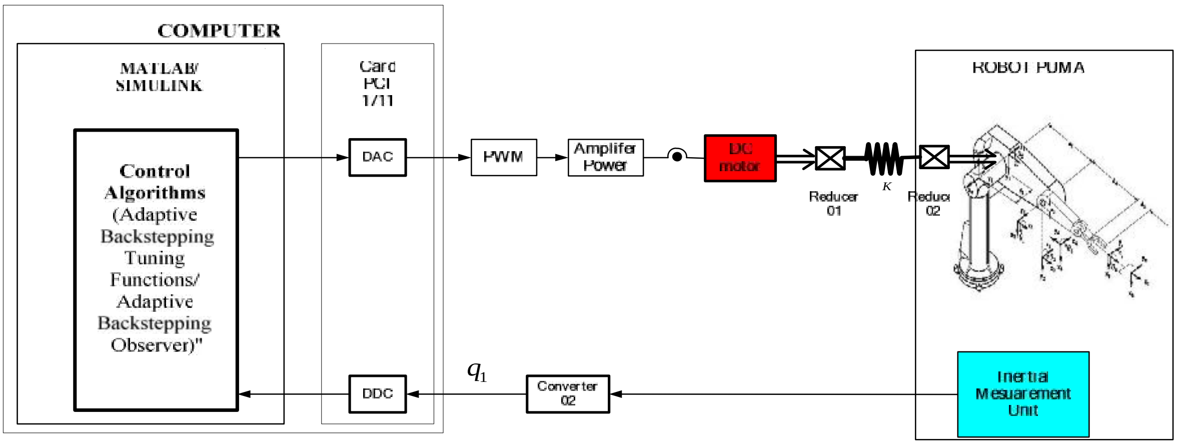
COMPUTER
MATLAB/ SIMULINK
Control Algorithms (Adaptive Backstepping Tuning Functions/ Adaptive Backstepping Observer)"
Card PCI 1711
DAC
DDC
PWM
Amplifer Power
DC motor
Reducer 01
K
Reducer 02
ROBOT PUMA
Inertial Mesuarement Unit
Converter 02
q_1

FIG. 2. MODEL OF CONTROL ALGORITHMS

3.2 Simulation Environment and Experiment Results

The proposed algorithms are simulated in MATLAB/Simulink, implemented in a real-time hardware using Real-time Workshop with DAQ cards. Model parameters can be viewed, modified in run-time allowing to find an optimal control parameter set, and controller types as well.

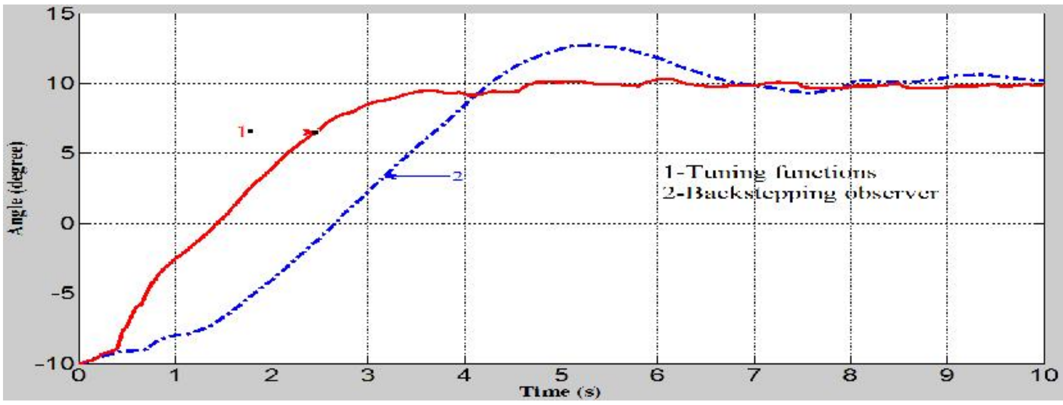

FIG. 3. EXPERIMENT RESULT

The experiment result for control algorithms is shown in Fig. 3, where the setpoint (for robot arm angle) changes from -10^0 to +10^0, lines 1 and 2 show changes of the robot arm angle output signals for backstepping tuning functions with K-filters and backstepping observer controllers respectively. The tuning functions controller has shorter response time and better control performance in terms of stability and convergence.

IV. CONCLUSION

In this paper, an adaptive backstepping control with tuning functions and K-filters for robot manipulators is developed and implemented. Applying tuning functions with K – filters, the strongest stability and convergence properties can be achieved. The result is shown for a real-time system of a single-link flexible-joint manipulator, but the proposed algorithms can be extended and applied to more complex industrial robot manipulators.

REFERENCES

[1] J M. Krstic, I. Kanellakopoulos, P. Kokotovic, “Nonlinear and Adaptive Control Design”, Wiley, 1995.

[2] Tran Xuan Kien, Do Duc Hanh, “Adaptive Observer Backstepping Control for Industrial Robot Manipulators Using IMU”. International Journal of Emerging Technology and Advanced Engineering Volume 6, Issue 1, January 2016, 102-111.

[3] Bo Zhou, “Backstepping Based Global Exponential Stabilizaton of a Tracked Mobile Robot with Slipping Pertubation”, Journal of Bionic Engineering, 2011, 8:69-76.

[4] Yongming Li, “Adaptive Fuzzy Output Feedback Control for a Single-Link Flexible Robot Manipulator Driven DC Motor via Backstepping”, Nonlinear Analysis:Real World Applications, 2013, 14: 483-494.