S. Hanba,
Numerical nonlinear observers using pseudo-Newton-type solvers,
International Journal of Robust and Nonlinear Control, Vol. 18 no. 17, pp. 1592–1606, 2008
doi: 10.1002/rnc.1296

Numerical nonlinear observers using pseudo-Newton type solvers

Shigeru Hanba[1,*]

[1] *Department of Electrical and Electronic Engineering, University of the Ryukyus, 1 Senbaru Nishihara, Nakagami-gun, Okinawa 903-0213, Japan*

SUMMARY

In constructing a globally convergent numerical nonlinear observer of Newton-type for a continuous-time nonlinear system, a globally convergent nonlinear equation solver with a guaranteed rate of convergence is necessary. In particular, the solver should be Jacobian-free, because an analytic form of the state transition map of the nonlinear system is generally unavailable. In this paper, two Jacobian-free nonlinear equation solvers of pseudo-Newton type that fulfill these requirements are proposed. One of them is based on the finite difference approximation of the Jacobian with variable step size together with the line search. The other uses similar idea, but the estimate of the Jacobian is mostly updated through a BFGS-type law. Then, by using these solvers, globally stable numerical nonlinear observers are constructed. Numerical results are included to illustrate the effectiveness of the proposed methods.

KEY WORDS: nonlinear systems, observers, Jacobian-free, nonlinear equations, pseudo-Newton method

1. Introduction

In recent years, observers for continuous-time nonlinear systems have been studied extensively, and many constructions have been reported, such as those in [2, 4, 8, 11, 13, 14, 15, 17, 18, 22, 25], to name a few. Among them, the numerical nonlinear observers proposed by Moraal and Grizzle in their seminal paper[22] have especially useful features: they require only the sample-valued measurements of the output signal, and they are applicable to a wide class of nonlinear systems. In the context of moving horizon control, a similar observer was proposed by Michalska and Mayne almost simultaneously[21]. Due to the importance of this concept, similar or related constructions have been studied by several researchers[1, 3, 12, 24]. However, most of them have a common drawback: they require the analytic expressions of the sample-valued state transition maps, which are difficult to construct for general nonlinear systems.

This difficulty was resolved by Biyik and Arcak's brilliant idea of using a Jacobian-free solver for nonlinear equations, while assuming that only the values of the state transition maps are available [5]. Their contribution made a numerical nonlinear observer readily implementable, because the values of the maps can be obtained through time-scaled continuous nonlinear filters (analog implementation) or numerical integration (digital implementation).

Nevertheless, their method has two drawbacks: first, it is not always easy to estimate the required step size for the finite difference approximation used in generating an approximated

Jacobian *a priori*; and second, their line search technique, based on Wolfe-like conditions, is not always well defined. The objective of this paper is to resolve these drawbacks.

This paper is arranged as follows. In Section 2, we first introduce the idea of numerical nonlinear observers based on [22], and then describle the technical problems involved in Biyik and Arcak's construction[5] more closely. In Section 3, two globally convergent Jacobian-free solvers of pseudo-Newton type for nonlinear equations with guaranteed rates of convergence are proposed. In Section 4, by using these solvers, globally stable numerical nonlinear observers are constructed. Section 5 contains examples on nonlinear equation solvers and numerical nonlinear observers.

In what follows, the symbol $\|\cdot\|$ denotes the Euclidean norm for a vector and a matrix norm compatible to the Euclid norm for a matrix. The symbol I denotes identity matrices of appropriate dimensions.

2. Structure of the numerical nonlinear observers

Consider a smooth continuous-time nonlinear system of the form

$$\dot{x} = \phi(x, u), \quad y = h(x), \tag{1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $y \in \mathbb{R}^p$. For simplicity, we assume that $p = 1$ henceforth.

Given a sampling period T, we assume that $y(t)$ at each sampling instant kT is available; henceforth, kT is denoted by t_k; $x(kT)$ and $y(kT)$ are denoted by x_k and y_k, respectively. The state transition map from x_k to x_{k+1} is denoted by f_k, while omitting the dependence on $u(t)$. Hence, the discrete-time counterpart for (1) is of the form

$$x_{k+1} = f_k(x_k), \quad y_k = h(x_k). \tag{2}$$

Let $\eta_k = (y_k, \ldots, y_{k-n+1})^T$ and

$$\Phi_k(x_{k-n+1}) = \begin{pmatrix} h \circ f_{k-1} \circ \cdots \circ f_{k-n+1}(x_{k-n+1}) \\ \cdots \\ h(x_{k-n+1}) \end{pmatrix}.$$

Then,

$$\eta_k = \Phi_k(x_{k-n+1}). \tag{3}$$

In what follows, we assume that Φ_k are bijective for all k.

Moraal and Grizzle's nonlinear observer first solves the nonlinear equation (3) numerically (hence, at the time instant t_k, the state estimate $n-1$ -steps prior to the current time, $\widehat{x}_{k-n+1}$, is obtained), and then generates the state estimate at t_k by the forward transition

$$f_{k-1} \circ \cdots \circ f_{k-n+1}(\cdot). \tag{4}$$

Let the initial value of $\widehat{x}_{k-n+1}$ for solving (3) be $\widehat{x}^0_{k-n+1}$. In Moraal and Grizzle's framework, $\widehat{x}^0_0$ is set freely, and $\widehat{x}^0_{k+1}$ is given by $f_k(\widehat{x}_k)$.

Assume that, for all k, f_k, Φ_k, and Φ_k^{-1} are Lipschitz continuous (uniformly with respect to k) with Lipschitz constants Γ_f, Γ_Φ, and $\Gamma_{i\Phi}$, respectively. Then,

$$\|x_{k-n+2} - \widehat{x}^0_{k-n+2}\| \le \Gamma_f \|x_{k-n+1} - \widehat{x}_{k-n+1}\|. \tag{5}$$

Moreover, by letting $\widehat{\eta}_k^0 = \Phi_k(\widehat{x}_{k-n+1}^0)$ and $\widehat{\eta}_k = \Phi_k(\widehat{x}_{k-n+1})$, we obtain

$$\begin{aligned} \|x_{k-n+1} - \widehat{x}_{k-n+1}\| &\leq \Gamma_{i\Phi}\|\eta_k - \widehat{\eta}_k\|, \\ \|\eta_k - \widehat{\eta}_k^0\| &\leq \Gamma_{\Phi}\|x_{k-n+1} - \widehat{x}_{k-n+1}^0\|. \end{aligned} \tag{6}$$

Let us assume that a nonlinear equation solver satisfying

$$\|\eta_k - \widehat{\eta}_k\| \leq \gamma\|\eta_k - \widehat{\eta}_k^0\| \tag{7}$$

is given. Then, from (5), (6), and (7), we obtain

$$\|x_{k-n+2} - \widehat{x}_{k-n+2}^0\| \leq \gamma\,\Gamma_f\Gamma_{i\Phi}\Gamma_{\Phi}\|x_{k-n+1} - \widehat{x}_{k-n+1}^0\|. \tag{8}$$

Hence, the problem of constructing a stable nonlinear observer is reduced to constructing a nonlinear equation solver satisfying

$$\gamma\,\Gamma_f\Gamma_{i\Phi}\Gamma_{\Phi} < 1. \tag{9}$$

Remark 1
It is also possible to construct a numerical nonlinear observer based on the inverse-time state transition map

$$x_k = f_k^-(x_{k+1}), \quad y_k = h(x_k), \tag{10}$$

which is obtained by solving (1) with respect to the inverted time axis. The only difference between using (2) and using (10) is that, if one uses (10), then the forward transition (4) is not necessary.

If the Jacobian of the nonlinear equation to be solved is known, and if sufficient CPU power is available, a solver satisfying requirement (9) can be constructed in a straightforward manner. Moraal and Grizzle used the continuous Newton method[22]; alternatively, any globally convergent discrete Newton method can be used. However, the use of Newton methods requires the analytic expressions of (2), which are difficult to construct for general nonlinear systems.

This difficulty was overcome by Biyik and Arcak[5], who only assumed that the values of (2) are available, and introduced a Jacobian-free (semi-)globally convergent solver for (3). Nevertheless, there construction has several drawbacks, all of which are related to the line search. We shall now analyze these problems more closely.

Temporally, let $G(x) = 0$ be the nonlinear equation to be solved in the numerical nonlinear observer, and define $g(x) = G(x)^T G(x)/2$. Assume that $\nabla g(x) \neq 0$, and that a search vector s satisfying $\nabla g(x)s < 0$ is already obtained. The step size λ for updating x as $x = x + \lambda s$ to assure the global convergence is yet to be determined. A common method for determining this λ is to attempt to find a positive λ that satisfies the Wolfe conditions:

$$g(x + \lambda s) \leq g(x) + \kappa_1\lambda\nabla g(x)s, \quad \nabla g(x + \lambda s)s \geq \kappa_2\nabla g(x)s, \tag{11}$$

where $0 < \kappa_1 < \kappa_2 < 1$, by some search method such as the backtracking. However, the inequalities (11) may not be used in the Jacobian-free setup, because $\nabla g(x)$ contains the Jacobian of $G(x)$. Hence, Biyik and Arcak replaced the gradient $\nabla g(x)$ in (11) with the approximated gradient $\nabla_a g(x) = G(x)^T \widehat{J}_G(\rho, x)$, where $\widehat{J}_G(\rho, x)$ is the approximated Jacobian

of $G(x)$ generated by the finite difference approximation with a step size ρ. Thus, their new Wolfe-like conditions are as follows:

$$g(x+\lambda s) \le g(x) + \kappa_1 \lambda \nabla g_a(x) s, \quad \nabla g_a(x+\lambda s) s \ge \kappa_2 \nabla g_a(x) s. \tag{12}$$

In Biyik and Arcak's method, the search vector s is also obtained by the approximated Jacobian through $s = -\widehat{J}_G(\rho, x)^{-1} G(x)$. Hence, the effect of the approximation error appears on both sides of the former and the latter inequalities in (12), which violates the existence of a positive λ satisfying (12). Therefore, their line search may not be well defined, making their stability analysis imperfect. This also causes a practical problem. Biyik and Arcak used the backtracking search (update λ by $\lambda = \beta_B \lambda$, with $0 < \beta_B < 1$) for determining λ, but because the approximation makes the set of positive λs satisfying (12) smaller, the backtracking search readily overlooks a λ even when one exists. To decrease the risk of overlooking a solution, it is necessary to let β_B nearly equal to unity; this would make the algorithm inefficient. Moreover, their algorithm requires that a sufficiently small step size for the finite difference approximation of the Jacobian is known, because otherwise the condition $\nabla g(x) s < 0$ may not be satisfied, but it is not easy to estimate the size *a priori*. To overcome these difficulties, the nonlinear equation solver itself should be reconsidered.

In nonlinear programming literature, several globally convergent Jacobian-free nonlinear minimizers or nonlinear equation solvers are available[9, 19, 20]. However, none of them give an explicit rate of convergence; hence, they are not suitable for numerical nonlinear observers.

On the other hand, if we assume that, for each x, a search vector p satisfying the pseudo-Newton condition

$$\left\| \frac{\partial f}{\partial x} p + f(x) \right\| \le \eta \|f(x)\|, \quad 0 \le \eta < 1 \tag{13}$$

is available, it is possible to construct a globally convergent pseudo-Newton type nonlinear equation solver which gives an explicit rate of convergence[6, 10, 16]. Therefore, if it is possible to remove the assumption on the availability of p, then the resulting solver is directly applicable to the numerical nonlinear observers, and the difficulties in Biyik and Arcak's method are resolved. This type of algorithm is proposed in the following section.

3. Nonlinear equation solvers of pseudo-Newton type

In this section, we consider the problem of constructing globally convergent solvers for a nonlinear equation

$$f(x) = 0, \quad f : \mathbb{R}^n \to \mathbb{R}^n. \tag{14}$$

We assume the following throughout this section.

Assumption 1
The function f and the initial value x_0 satisfy the following conditions.

1. The nonlinear equation (14) has a unique solution x_*.
2. The function f is continuously differentiable.
3. The set $\Omega_0 = \{x : \|f(x)\| \le \|f(x_0)\|\}$ is compact.
4. There exists a sufficiently large compact set Ω containing Ω_0, in which the Jacobian of f (denoted by $J(x)$) is Lipschitz continuous, with a Lipschitz constant L, that is, $\forall x, y \in \Omega, \|J(x) - J(y)\| \le L\|x - y\|$.

5. $J(x)$ is invertible for all $x \in \Omega$.
6. $J(x)^{-1}$ is continuous on Ω as a function of x.

We begin with an easy lemma on the norm of a perturbed nonsingular matrix.

Lemma 1
Let A and B be square matrices with A being nonsingular and $|\rho| < 1/(2\|A^{-1}\|\|B\|)$. Then $A - \rho B$ is nonsingular and

$$\|(A-\rho B)^{-1}\| \le \|A\|^{-1} + |\rho|(\|A^{-1}\|)^2\|B\|(\|I\|+1). \tag{15}$$

Proof. Let $Q = I + \rho(A^{-1}B) + \rho^2(A^{-1}B)^2 + \cdots$. If $|\rho| < 1/(2\|A^{-1}\|\|B\|)$, then Q converges to $(I - \rho A^{-1}B)^{-1}$ with $\|Q\| \le \|I\| + 1$. Because $Q = I + \rho A^{-1}BQ$, we obtain

$$(A-\rho B)^{-1} = QA^{-1} = A^{-1} + \rho A^{-1}BQA^{-1}. \tag{16}$$

Taking the norm of the right hand side of (16) gives (15). □

In what follows, we assume that a point x satisfying

$$\|f(x)\| \le \|f(x_0)\| \tag{17}$$

is already given, and that all the following calculations are executable inside Ω.

Lemma 2
For a search vector p and step size ρ,

$$f(x+\rho p) - f(x) - \rho J(x)p = \int_0^\rho tR_p(t,x,p)pdt \tag{18}$$

for some $R_p(t,x,p)$ with $\|R_p(t,x,p)\| \le L\|p\|$, and

$$\|f(x+\rho p) - f(x) - \rho J(x)p\| \le \frac{1}{2}\rho^2 L\|p\|^2. \tag{19}$$

Proof. Let $J(x+tp) = J(x) + R(x,t,p)$. Since J is Lipschitz continuous, it follows that $\|R(x,t,p)\| \le Lt\|p\|$. Because

$$f(x+\rho p) - f(x) = \int_0^\rho \frac{d}{dt}f(x+tp)dt = J(x)\rho p + \int_0^\rho R(x,t,p)pdt, \tag{20}$$

we obtain

$$\|f(x+\rho p) - f(x) - J(x)\rho p\| \le \int_0^\rho \|R(x,t,p)p\|\,dt \le \int_0^\rho Lt\|p\|^2 dt. \tag{21}$$

The desired result is obtained by integrating the rightmost term of (21). □

Let $\widehat{J}(\rho,x)$ be the finite difference approximation of $J(x)$ with the step size ρ, and

$$\sigma_R = \sup_{r_i\in\mathbb{R}^n, \|r_i\|\le 1, i=1,\ldots,n} \|(r_1,\ldots,r_n)\|.$$

Then, by Lemma 2,we obtain

$$\|\widehat{J}(\rho,x) - J(x)\| \le \frac{1}{2}\rho L\sigma_R. \tag{22}$$

We are trying to generate a search vector p by solving the linear equation

$$\widehat{J}(\rho,x)p = -f(x); \tag{23}$$

thus, whether $\widehat{J}(\rho,x)$ is nonsingular or not is significant. From (22), for some $R_J(\rho,x)$ with $\|R_J(\rho,x)\| < \sigma_R$, $\widehat{J}(\rho,x) = J(x) + (1/2)\rho L R_J(\rho,x)$. Let $\sigma_{iJ} = \sup_{x\in\Omega}\|J^{-1}(x)\|$. From Assumption 1, σ_{iJ} is finite. We temporally assume that

$$\rho L < \frac{1}{\sigma_{iJ}\sigma_R}. \tag{24}$$

Then, by Lemma 1, $\widehat{J}(\rho,x)$ is nonsingular. Moreover, from (16), by letting

$$\begin{aligned} Q(\rho,x) &= \left(I + \frac{1}{2}\rho L J(x)^{-1}R_J(\rho,x)\right)^{-1}, \\ R_{iJ}(\rho,x) &= J(x)^{-1}R_J(\rho,x)Q(\rho,x)J(x)^{-1}, \end{aligned} \tag{25}$$

we obtain

$$\widehat{J}(\rho,x)^{-1} = J(x)^{-1} - \frac{1}{2}\rho L R_{iJ}(\rho,x). \tag{26}$$

Further, since $\|Q(\rho,x)\| \le \|I\| + 1$, it follows that

$$\|R_{iJ}(\rho,x)\| \le \sigma_{iJ}^2\sigma_R(\|I\| + 1). \tag{27}$$

Next, we evaluate the norm of the solution p of (23). Because

$$\|p\| \le \|\widehat{J}(\rho,x)^{-1}\|\|f(x)\|,$$

by using (26), we obtain

$$\|p\| \le \left(\sigma_{iJ} + \frac{1}{2}\rho L\sigma_{iJ}^2\sigma_R\left(\|I\| + 1\right)\right)\|f(x)\|, \tag{28}$$

which is simplified as

$$\|p\| \le \sigma_{iJ}\left(\frac{3}{2} + \frac{1}{2}\|I\|\right)\|f(x)\| \tag{29}$$

by using (24).

Now, we assume that $\rho < 1$ and evaluate the effect of letting $x \to x + \rho p$. From (19), it follows that

$$\|f(x+\rho p)\| \le \|f(x) + \rho J(x)p\| + \frac{1}{2}\rho^2 L\|p\|^2. \tag{30}$$

On the other hand, from (23) and (26), we obtain

$$p = -J(x)^{-1}f(x) + \frac{1}{2}\rho L R_{iJ}(\rho,x)f(x). \tag{31}$$

Substituting (31) into (30), using (27), (28), and (29), and letting $\sigma_J = \sup_{x\in\Omega}\|J(x)\|$ gives

$$\begin{aligned} \|f(x+\rho p)\| \le{}& (1-\rho)\|f(x)\| + \frac{1}{2}\rho^2 L\sigma_J\sigma_R\sigma_{iJ}^2(\|I\| + 1)\|f(x)\| \\ &+ \frac{1}{2}\rho^2 L\sigma_{iJ}^2\left(\frac{3}{2} + \frac{1}{2}\|I\|\right)^2\|f(x)\|^2. \end{aligned} \tag{32}$$

From (17), $\|f(x)\|^2 \leq \|f(x_0)\|\|f(x)\|$. Hence, for some κ satisfying $0 < \kappa < 1$, if

$$\frac{1}{2}\rho L\sigma_{iJ}^2 \left(\sigma_J \sigma_R \left(\|I\| + 1 \right) + \left(\frac{3}{2} + \frac{1}{2}\|I\| \right)^2 \|f(x_0)\| \right) < 1 - \kappa, \tag{33}$$

then (32) is reduced to

$$\|f(x + \rho p)\| \leq (1 - \rho)\|f(x)\| + \rho(1 - \kappa)\|f(x)\| = (1 - \kappa\rho)\|f(x)\|. \tag{34}$$

We have obtained the following lemma.

Lemma 3
Choose a $\kappa \in (0, 1)$ and fix a $\rho \in (0, 1)$ satisfying (24) and (33). Let $x_k \in \Omega$ be such that $\|f(x_k)\| \leq \|f(x_0)\|$. Then, letting

$$x_{k+1} = x_k + \rho p_k \tag{35}$$

by using the solution p_k of (23) gives

$$\|f(x_{k+1})\| \leq (1 - \kappa\rho)\|f(x_k)\|. \tag{36}$$

By using Lemma 3, an algorithm for solving (14) is obtained.

Algorithm 1 (Pseudo-Newton Method)
Initialization. Choose positive initial values of ρ (the step size), β (the decay rate of the step size), ε (the accuracy parameter), β_B (the backtracking parameter*) and a positive constant κ with $0 < \beta < 1$, $0 < \beta_B < 1$ and $0 < \kappa < 1$;
Main loop.
while ($\|f(x)\| > \varepsilon$) **do**
  if ($\widehat{J}(\rho, x)$ is nonsingular) **then**
    Solve $\widehat{J}(\rho, x)p = -f(x)$;
    $\lambda = 1$; final=FALSE;
    while (TRUE) **do**
      if ($\|f(x + \lambda p)\| \leq (1 - \kappa\rho)\|f(x)\|$) **then**
        $x = x + \lambda p$; success=TRUE; **break**;
      else
        if (final == TRUE) **then**
          success=FALSE; **break**;
        else
          $\lambda = \beta_B \lambda$;
          if ($\lambda < \rho$) **then**
            $\lambda = \rho$; final=TRUE;
  else
    success = FALSE;
  if (success == FALSE) **then**

*One may wonder why nearly-exact differentiation-free line search methods (such as the Fibonacci method and the golden section method) are not used here. The reason is that, in pseudo-Newton methods, the search vectors are not always good approximations of $-(J(x))^{-1}f(x)$, and searching a nearly-exact minimum along a bad search vector is sometimes inefficient.

$\rho = \beta\rho$;

Theorem 1
The sequence $(x_k)_{k\in\mathbb{N}}$ generated by Algorithm 1 converges to x_*.

Proof. Let the supremum of ρ satisfying (24) and (33) be ρ_*, the initial value of ρ be ρ_0, $N_0 = \min\{k \in \mathbb{N} : \beta^k\rho_0 < \max\{\rho_*, 1\}\}$, and $\beta^{N_0}\rho_0 = \overline{\rho}$. Lemma 3 states that the step size does not decrease below $\overline{\rho}$. Therefore, for $k > N_0$, there are at least $k - N_0$ iterations satisfying (36) with $\rho \geq \overline{\rho}$; hence,

$$\|f(x_k)\| \leq (1 - \kappa\overline{\rho})^{k-N_0}\|f(x_0)\|. \tag{37}$$

Thus, the sequence $(f(x_k))_{k\in\mathbb{N}}$ converges to zero. From Assumption 1, (14) has a unique solution x_*, and f is locally homeomorphic because $J(x)$ is nonsingular. Hence, $(x_k)_{k\in\mathbb{N}}$ converges to x_*. □

Now, recall that the application we have in mind is numerical nonlinear observers, in which the function values are obtained through numerical integration. Algorithm 1 requires a new $\widehat{J}(\rho, x)$ at each step. This necessitates n function calls at each step, a process that is time-consuming and not desirable. To remedy this situation, it is effective to update an estimate B of $\widehat{J}(\rho, x)$ through a BFGS-type rule[7].

Algorithm 2 (BFGS-type Pseudo-Newton Method)
Initialization. In addition to the parameters in Algorithm 1, choose a nonsingular initial value of B;
newB=TRUE;
Main loop.
while ($\|f(x)\| > \varepsilon$) **do**
  if (B is nonsingular) **then**
    Solve $Bp = f(x)$; $\lambda = 1$; final=FALSE;
    while (TRUE) **do**
      if ($\|f(x + \lambda p)\| \leq (1 - \kappa\rho)\|f(x)\|$) **then**
        $s = \lambda p$; $B = B + (f(x + s) - f(x) - Bs)s^T/\|s\|^2$;
        $x = x + s$; newB=FALSE; success=TRUE; **break**;
      else
        if (final == TRUE) **then**
          success=FALSE; **break**;
        else
          $\lambda = \beta_B\lambda$;
          if ($\lambda < \rho$) **then**
            $\lambda = \rho$; final=TRUE;
  else
    success=FALSE;
  if ((newB==TRUE) && (success==FALSE)) **then**
    $\rho = \beta\rho$;
  if (success==FALSE) **then**
    $B = \widehat{J}(x, \rho)$; newB=TRUE;

Remark 2
Algorithm 2 converges globally, similar to Algorithm 1. As a matter of fact, the mechanism that makes Algorithm 2 globally convergent is exactly the same as that of Algorithm 1, and it

is independent of the convergence property of standard BFGS-type algorithms. Note that the step size ρ is updated only if the line search fails for a newly generated B (which is $\widehat{J}(\rho, x)$). Hence, once ρ satisfies $\rho < \min\{\rho_*, 1\}$, the line search for the newly generated B is always successful for $\lambda = \rho$. Thus, ρ is not decreased any further. By the same reason, the steps required for ρ to achieve the desired value (that is, $\overline{\rho}$) is $2N_0$ (we are using the same symbols as those used in the proof of Theorem 1). Therefore, at the k -th iteration ($k > 2N_0$), the worst-case value of $\|f(x_k)\|$ is

$$\|f(x_k)\| \leq (1 - \kappa\overline{\rho})^{[(k-2N_0)/2]} \|f(x_0)\|, \tag{38}$$

where $[\alpha]$ denotes the maximum integer that does not exceed α.

4. Stability condition for the numerical nonlinear observers

Since the algorithms constructed in Section 3 are linearly convergent (exponentially stable) except for the delay terms, it is applicable to the observer described in Section 2. If sufficient number of iterations are permitted in the solver, the condition (9) is fulfilled, and a globally stable numerical nonlinear observer is obtained. However, the number of iterations is yet to be determined. The objective of this section is to give a worst case estimate of this number of iterations.

First, we consider Algorithm 1. For simplicity, we assume that all the parameters in the algorithms are reset at t_k. Then, in the worst case, the number of iterations for the step size to decrease to an appropriate level is

$$[-\log_\beta(\rho_0/\rho^*)] + 1, \tag{39}$$

and the number of iterations for (9) to be fulfilled is

$$[-\log_{(1-\kappa\overline{\rho})} \Gamma_f \Gamma_{i\Phi} \Gamma_\Phi] + 1 \tag{40}$$

(note that $\beta < 1$, $1 - \kappa\overline{\rho} < 1$). Hence, the estimate of the required number of iterations at each step is

$$\left[-\log_\beta \frac{\rho_0}{\rho^*}\right] + \left[-\log_{(1-\kappa\overline{\rho})} \Gamma_f \Gamma_{i\Phi} \Gamma_\Phi\right] + 2. \tag{41}$$

If the parameters of Algorithm 1 at t_k succeed to t_{k+1}, then it may sometimes be possible to ignore the delay terms (39).

If Algorithm 2 is used, then the worst case estimate of the necessary number of iterations is twice that in (41). Practically, it would be effective to succeed the value of B at t_k to t_{k+1}, but evaluating its effect is not straightforward.

Finally, it is to be noted that, since (41) contains many parameters that are difficult to estimate *a priori*, it is not practical to determine the number of iterations by using (41), which only provides theoretical assurance. Practically, the number of iterations should be determined by a cut-and-try method.

5. Numerical examples

Example 1

This example is taken from [19], but has been modified slightly from the original.

Given $x \in \mathbb{R}^n$, consider the problem of solving the equation $Ax + g(x) = 0$ with $g_i(x) = \tan^{-1}(x_i) - 1$, where $A = (a_{ij})$, $a_{ii} = 2$, $a_{ij} = a_{ji} = -1 (i = j + 1)$, and $a_{ij} = 0$ otherwise.

In this example, we compare Li and Fukushima's algorithm[19], Biyik and Arcak's algorithm[5], Algorithm 1 and Algorithm 2 by solving the equation for $n = 9$ and $n = 99$ with the termination condition that $\|Ax + g(x)\| \leq 10^{-7}$ for twelve initial values of x, namely, $x_0^1 = (1, \ldots, 1)^T$, $x_0^2 = 10x_0^1$, $x_0^3 = 100x_0^1$, $x_0^4 = 1000x_0^1$, $x_0^5 = -x_0^1$, $x_0^6 = -x_0^2$, $x_0^7 = -x_0^3$, $x_0^8 = -x_0^4$, $x_0^9 = (1, \ldots, n)^T$, $x_0^{10} = -x_0^9$, $x_0^{11} = (n, \ldots, 1)^T$, and $x_0^{12} = -x_0^{11}$. The experiment was performed on an IBM compatible PC with Scilab.

For each algorithm, the parameters were varied within some sets, which will be described below. The results of the best parameters with respect to the average number of function calls are listed in Table I, where "Itr" denotes the average number of iterations and "Fcall" denotes the average number of function calls over all initial conditions.

In Li and Fukushima's algorithm, σ_1 and σ_2 were fixed at 0.001, ρ was fixed at 0.9 (these values are taken from [19]; see [19] for the meanings of these symbols), and only the backtracking parameter β_B was varied from 0.01 to 0.99 in steps of 0.01. For $n = 9$, $\beta = 0.12$ was best, whereas for $n = 99$, $\beta = 0.01$ was best.

In Biyik and Arcak's algorithm, κ_1 and κ_2 were fixed at 10^{-4} and 0.9, respectively, which are the typical values taken from [23]; ρ was varied within 10^i, $i \in \{-12, \ldots, -1\}$ (see [5] for the meanings of these symbols); and the backtracking parameter for the line search (β_B; not specified in [5]) was varied from 0.01 to 0.99 in steps of 0.01. To avoid the runaway, the backtracking line search was terminated when the step size was below 10^{-10}. For $n = 9$, two parameter pairs produced the same results, namely, $\rho = 10^{-11}$, $\beta_B = 0.73$ and $\rho = 10^{-11}$, $\beta_B = 0.73$. For $n = 99$, $\rho = 10^{-11}$ and $\beta_B = 0.91$ was best.

In both Algorithms 1 and 2, κ and ρ_0 are fixed at 0.01 and 1, respectively; β was varied within $4^i, i \in \{-12, \ldots, -1\}$; and β_B was varied from 0.01 to 0.99 in steps of 0.01.

For Algorithm 1 with $n = 9$, $\beta = 4^i$, $i = -6, -7, -8, -9, -10, -11, -12$ and $\beta_B = 0.72$ produced the same results, whereas with $n = 99$, $\beta = 4^i$, $i = -6, -7, -8$ and $\beta_B = 0.79$ produced the same results. For Algorithm 2 with $n = 9$, $\beta = 4^{-1}$ and $\beta_B = 0.58$ produced the best result, whereas with $n = 99$, $\beta = 4^{-12}$ and $\beta_B = 0.24$ produced the best result.

In this example, for both $n = 9$ and $n = 99$, Algorithm 2 outperformed the other algorithms. The behaviors of Biyik and Arcak's algorithm and Algorithm 1 were similar, but for $n = 9$, Biyik and Arcak's algorithm was better, whereas for $n = 99$, Algorithm 1 was better, as long as the number of function calls were considered. The performance of Li and Fukushima's algorithm was next to that of Algorithm 2 for $n = 9$, but was the worst for $n = 99$.

Table I. Comparison of Li & Fukushima's algorithm, Biyik & Arcak's algorithm, Algorithm 1, and Algorithm 2

	$n = 9$		$n = 99$	
	Itr	Fcall	Itr	Fcall
Li&Fukushima	25.83	36.67	1267.00	3580.75
Biyik&Arcak	6.33	74.00	11.92	1344.67
Algorithm1	10.58	107.50	12.33	1246.75
Algorithm2	12.00	22.92	36.25	155.67

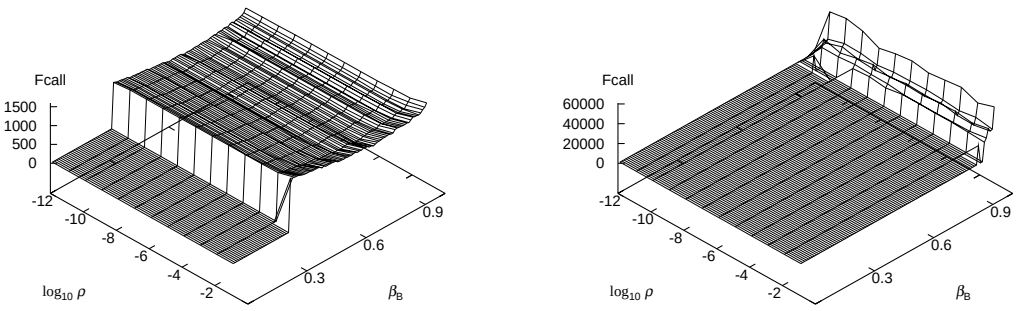
.

Algorithms 1 and 2 do not contain any mechanisms for preventing ρ from becoming too small; therefore, ρ may become unacceptably small before convergence. This, however, did not occur in this example. For Algorithm 1, the value of ρ ranged within $[4^{-6}, 4^{0}]$ when the algorithm terminated, whereas for Algorithm 2, it ranged within $[4^{-12}, 4^{0}]$ (the values were different for different initial conditions).

As explained in Section 2, Biyik and Arcak's algorithm has an issue that concerns the well-posedness of the line search. Therefore, to what extent certain badly selected parameters affect the performance is of some interest. Accordingly, the dependence of the number of function calls on some typical parameters of Biyik and Arcak's algorithm, Algorithm 1, and Algorithm 2 are shown in Figures 1, 2, and 3, respectively. In these figures, the vertical axis represents the number of function calls, but the value should be read with some care. In this experiment, the trial for each parameter was terminated and was marked as "failure" immediately after the number of iterations exceeded 10^5, and the corresponding number of function calls was reset to zero. Therefore, the value "0" on the vertical axis implies that the trial has failed.

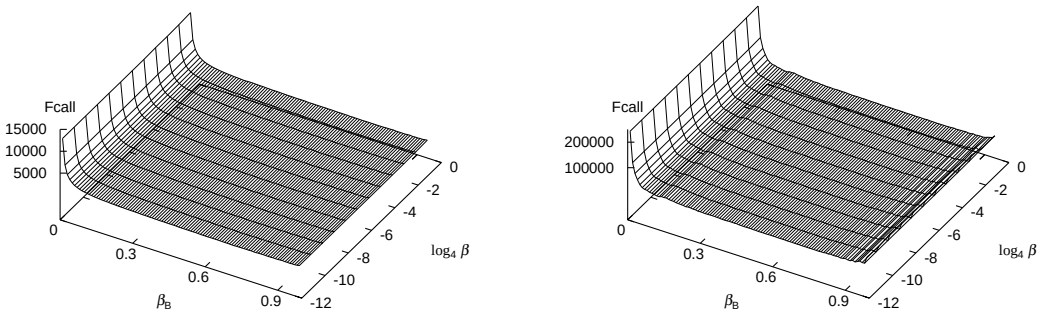
The result of Biyik and Arcak's algorithm is shown in Figure 1; for the parameters of the horizontal axes, the step size for the finite difference approximation (ρ, log-scaled) and the backtracking parameter (β_B) are selected. For $n = 9$, the algorithm failed for one thirds of the parameters, but the performance was almost the same for all successful parameters. For $n = 99$, the algorithm failed for almost all parameters, and β_B significantly affected the performance. Unfortunately, the performance was better for smaller value of β_B, but the solution was not obtained if β_B was too small.

Figures 2 and 3 show the results of Algorithms 1 and 2, respectively. The decay rate for the step size of the finite difference approximation (β, log-scaled) and the backtracking parameter (β_B) are selected as the parameters of the horizontal axes. These algorithms were successful for all parameters (this cannot be seen clearly from the figures, though), and their qualitative behavior was similar. When the value of β_B was close to zero, the performance was significantly affected; however, when the value was around $0.3 \sim 0.6$, the performance was good almost all the time. The effect of β, on the other hand, was not significant.



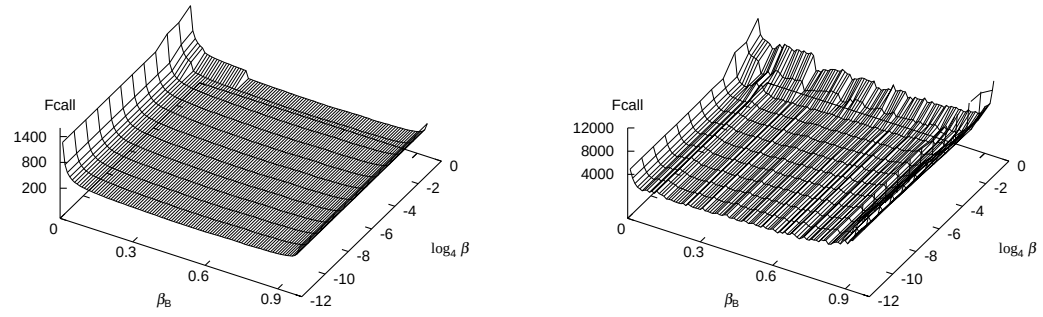
$n = 9$ $n = 99$

Figure 1. Biyik and Arcak's algorithm– dependence of the performance on parameters.



$n = 9$ $n = 99$

Figure 2. Algorithm 1– dependence of the performance on parameters.



$n = 9$ $n = 99$

Figure 3. Algorithm 2– dependence of the performance on parameters.

Example 2

In this example, we consider the following four-dimensional system

$$\dot{x} = A(x - r^0) + g(x_r^0), \tag{42}$$

with $x = (x_1, x_2, x_3, x_4)^T$, $r^0 = (r_1^0, r_2^0, r_3^0, r_4^0)^T$,

$$g(x) = (c_1 \tan^{-1}(d_1 x_1), c_1 \tan^{-1}(d_1 x_2), c_2 \tan^{-1}(d_2 x_3), c_2 \tan^{-1}(d_2 x_4))^T,$$

and

$$A = \begin{pmatrix} A_1 & A_2 \\ 0 & A_4 \end{pmatrix}, \quad A_1 = \begin{pmatrix} -a_1 & -w_1 \\ w_1 & -a_1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad A_3 = \begin{pmatrix} -a_2 & -w_2 \\ w_2 & -a_2 \end{pmatrix},$$

which is a cascade connection of two-dimensional systems exhibiting one globally attractive limit cycle.

Numerical nonlinear observers based on Biyik and Arcak's algorithm, Algorithm 1 and Algorithm 2 were tested for the system with $a_1 = a_2 = 1$, $c_1 = 100$, $c_2 = 50$, $d_1 = 1$, $d_2 = 3$, $w_1 = 1$, $w_2 = 7$, $r^0 = (500, 500, -500, -500)^T$, and the initial condition $(100, 100, 100, 100)^T$. The experiment was performed on an IBM compatible PC with Mathematica.

In each observer, the initial value of the state estimates at $t_0 = 0$ was $(1, 1, 1, 1)^T$, the maximum number of iterations at each step was 5, the sampling period was 0.2, and the simulation was performed within the time interval (0, 10.6), producing a sequence of state estimates of length 50.

In each algorithm, the following parameter values were used. In Biyik and Arcak's algorithm, $\kappa_1 = 10^{-4}$, $\kappa_2 = 0.9$, $\beta_B = 0.1$, $\rho = 0.01$, with a lower bound of the step size in the backtracking line search equal to 10^{-10}. In Algorithm 1 and Algorithm 2, $\kappa = 10^{-4}$, $\rho_0 = 1$, and $\beta = \beta_B = 0.1$. In each algorithm, the estimation at t_k was terminated once $\|\eta_k - \widehat{\eta}_k\|$ was below 10^{-8}.

Figure 4 shows the trajectory of the state estimates of Algorithm 2; the solid line indicates the states, while the points indicate their estimates. The figures for Biyik and Arcak's algorithm and Algorithm 1 are omitted because they are almost the same as Figure 4. In all cases, the state estimate converged to the state extremely fast; therefore, it is not possible to read the state estimation error from the figure.

For Biyik and Arcak's algorithm, Algorithm 1 and Algorithm 2, the average state estimation errors over 50 trials were 1.03×10^{-10}, 7.56×10^{-11}, and 8.43×10^{-11}, respectively. The total number of function calls were 2457, 1650, and 1404, respectively. As far as the number of the function calls is considered, the performance of Algorithm 2 was the best. As for the state estimation error, there was no significant difference.

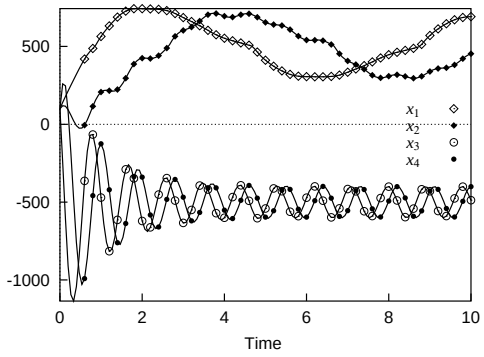


Figure 4. State estimation using Algorithm 2, the solid line represents the state, while the points represents their estimates.

6. Conclusions

In this paper, new globally convergent nonlinear equation solvers were proposed, and then globally stable numerical nonlinear observers were constructed by using the algorithms.

The proposed observers have a drawback: they are not robust against modeling errors, because the modeling error directly affects the state estimate through the inverse of the functions. To overcome this problem, observers using the nonlinear least squares method, treated axiomatically in [12], seem to be promising; however, to the author's knowledge, a Jacobian-free algorithm for solving the nonlinear least squares problems with a guaranteed (globally linear) rate of convergence, which is necessary in constructing a globally stable observer, is not available in literature. Therefore, the construction of a more robust nonlinear observer based on the nonlinear least squares methods is left to further research.

ACKNOWLEDGEMENTS

The author would like to express his gratitude toward the anonymous referees for their helpful and constructive comments.

REFERENCES

1. Alamir M, Calvillo-Corona LA. Further results on nonlinear receding-horizon observers. IEEE Transactions on Automatic Control 2002; **47**(7): 1184 – 1188.
2. Andrieu V, Praly L. On the existence of a Kazantzis–Kravaris/Luenberger observer. SIAM Journal on Control and Optimization 2006; **45**(2): 432–456.
3. Arcak M, Nešic D. A framework for nonlinear sampled-data observer design via approximate discrete-time models and emulation. Automatica 2004; **40**(11): 1931–1938.
4. Astolfi A, Praly L. Global complete observability and output-to-state stability imply the existence of a globally convergent observer. Mathematics of Control Signals and Systems 2006; **18**(1): 32–65.
5. Biyik E, Arcak M. A hybrid redesign of Newton observers in the absence of an exact discrete-time model. System & Control Letters 2006; **55**(6): 429–436.
6. Brown PN, Saad Y. Convergence theory of nonlinear Newton-Krylov algorithms. SIAM Journal on Optimization 1994; **4**(2): 297–330.
7. Broyden CG. A class of methods for solving nonlinear simultaneous equations. Mathematics of Computation 1965; **19**: 577–593.
8. Busvelle E, Gauthier J-P. Observation and identification tools for nonlinear systems application to a fluid catalytic cracker. International Journal of Control 2005; **78**(3): 208–234.
9. Conn AR, Scheinberg K, Toint PL. On the convergence of derivative-free methods for unconstrained optimization. in Approximation Theory and Optimization Tributes to M. J. D. Powell edited by M. D. Buhmann and A. Iserles Cambridge University Press Cambridge pp. 83 – 108 1997.
10. Eisenstat SC, Walker HF. Globally convergent inexact Newton methods. SIAM Journal on Optimization 1994; **4**(2): 393–422.
11. Hammouri H, Targui B, Armanet F. High gain observer based on a triangular structure. International Journal of Robust and Nonlinear Control 2002; **12**(6): 497–518.
12. Kang W. Moving horizon numerical observers of nonlinear control systems. IEEE Transactions on Automatic Control 2006; **51**(2): 344–350.
13. Karafyllis I, Kravaris C. Robust output feedback stabilization and nonlinear observer design. Systems & Control Letters 2005; **54**(10): 925–938.
14. Kazantzis N, Kravaris C. Nonlinear observer design using Lyapunovs auxiliary theorem. Systems & Control Letters 1998; **34**(5): 241–247.
15. Keller H. Non-linear observer design by transformation into a generalized. observer canonical form International Journal of Control 1987; **46**(6): 1915–1930.
16. Kelley CT. Iterative methods for linear and nonlinear equations. Society for Industrial and Applied Mathematics 1995.
17. Krener AJ, Respondek W. Nonlinear observers with linearizable error dynamics. SIAM Journal on Control and Optimation 1985; **23**(2): 197–216.
18. Krener AJ, Kang W. Locally convergent nonlinear observers. SIAM Journal on Control and Optimization 2003; **42**(1): 155–177.

19. Li D-H, Fukushima M. A derivative-free line search and global convergence of Broyden-like method for nonlinear equations. Optimization Methods & Software 2000; **13**: 181–200.
20. Lucidi S, Sciandrone M. On the global convergence of derivative-free methods for unconstrained optimization. SIAM Journal on Optimization 2002; **13**(1): 97–116.
21. Michalska H, Mayne DQ. Moving horizon observers and observer-based control. IEEE Transactions on Automatic Control 1995; **40**(6): 995–1006.
22. Moraal PE, Grizzle JW. Observer design for nonlinear systems with discrete-time measurements. IEEE Transactions on Automatic Control 1995; **40**(3): 395–404.
23. Nocedal J, Wright J. Numerical Optimization. Springer-Verlag 1999.
24. Rao CV, Rawlings JB, Mayne DQ. Constrained state estimation for nonlinear discrete-time systems stability and moving horizon approximation. IEEE Transactions on Automatic Control 2003; **48**(2): 246–258.
25. Zeitz M. The extended Luenberger observer for nonlinear systems. Systems & Control Letters 1987; **9**(2): 149–156.