Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol. 4
Lipo Wang, Jagath C. Rajapakse, Kunihiko Fukushima, Soo-Young Lee, and Xin Yao (Editors)

A STABLE NEURAL NETWORK OBSERVER WITH APPLICATION TO FLEXIBLE-JOINT MANIPULATORS

*F. Abdollahi, H.A. Talebi**

Dept. of Electrical Engineering
AmirKabir University of Technology
Tehran, Iran, 15914

R.V. Patel

Dept. of Electrical and Computer Engineering
University of Western Ontario
London, Ontario, Canada, N6A 5B9

ABSTRACT

A stable neural network based observer for general multivariable nonlinear system is considered in this paper. A linearly parameterized neural network is employed for approximation of the unknown nonlinearities of the system. The recurrent network configuration is obtained by a combination of feedforward network architectures with dynamical elements in the form of stable filters. The weights of the network are updated according to a novel approach based on the modification of the backpropagation algorithm. The stability of the system is shown using Lyapunov direct method. No SPR assumption is imposed on the output error equation. The proposed observer is applied to a flexible-joint manipulator to evaluate its performance. The simulation results show the effectiveness of the proposed observer.

1. INTRODUCTION

The state estimation is one of the essential issues in modern control theory since most control methodologies assume that the states are available for feedback. This is especially true for nonlinear systems where the so-called inverse dynamics (feedback linearization) is the most common approach. However, in most realistic cases, only the outputs of the plant rather than the full state vector can be measured or using measurement tools is not economical. Therefore, designing a state estimator with a good accuracy is an essential step for achieving a high performance control system. The Luenberger observer and Kalman filter are the most popular linear observers whose properties are well defined and studied. The extended Kalman filter and Luenberger-like observer based on linearization or quasi-linearization techniques have also been suggested. However, the locally linearized model works satisfactorily only in the neighborhood of the operating point. Several conventional nonlinear observers have been suggested during the past several years [1, 2]. Nevertheless, a fundamental assumption underlying most of these methods is that the system nonlinearities are completely known *a priori*. Practical systems, however, are prone to uncertainty. Therefore, unmodeled dynamics and parameter variations are always present in system dynamics. This will render the performance of the model-based observers. Robot manipulators with joint or link flexibility are good examples of such systems. The flexibility of the joint causes extreme difficulty in modeling manipulator dynamics and becomes a potential source of uncertainty that can degrade the performance of the manipulator and in some cases can even destabilize the system [3]. Consequently, addressing this issue is essential for calibration as well as modeling and control of robot manipulators. The adaptive nature of a *neural network* and its ability to deal with uncertainty makes it a viable choice for identification and state estimation of nonlinear systems. In [4], a neural network observer was proposed for affine SISO nonlinear systems. A recurrent neural network was used for estimating the nonlinear functions with the strictly positive real (SPR) assumption imposed on the output error equation. However, they considered *scalar* valued nonlinear functions in system dynamics. This implies that similar nonlinear terms correspond to all state variables. In [5], the general nonlinear model was considered. It was claimed that every general nonlinear model can be described by an affine model plus a bounded unmodeled dynamic term. Hence, the affine model was used for observer design. Nonetheless, it is not clear how the error can be made arbitrary small. In [6],[7], an observer for general MIMO nonlinear systems was proposed and the SPR assumption was also relaxed. According to the authors, it is extremely difficult to select the proper values for design parameters such as the various gains and functional links of the neural network. Moreover, the observer has an open-loop structure. In [8], a general MIMO nonlinear model was also considered. In contrast to the other works, the weight updating mechanism was based on the steepest descent method. The observer was shown to be experimentally stable, but no mathematical proof was given to support the experiment.

In this paper, a neuro-observer based on linearly pa-

This research was supported in part by the Natural Science and Engineering Research Council (NSERC) of Canada grant RGPIN-1345.

rameterized dynamic recurrent neural network for general nonlinear systems is proposed. The neural network weights update online based on backpropagation algorithm plus e-modification. The SPR assumption imposed on the output error equation is also relaxed. The rest of the paper is organized as follows: In Section 2, the proposed neural network observer is introduced and the main result of the paper (the stability proof) is given. Section 3 gives the model for flexible-joint manipulators. The observer performance is tested by simulation carried on a single-link flexible-joint manipulator which is given in Section 4. Finally, Section 5 concludes the paper.

2. THE PROPOSED NEURAL NETWORK BASED OBSERVER

Consider the general model of nonlinear MIMO system

$$\begin{aligned} \dot{x}(t) &= f(x,u) \\ y(t) &= Cx(t) \end{aligned} \tag{1}$$

where $u \in R^m$ is the input, $y \in R^m$ is the output and $x \in R^n$ is the state vector of the system and f is an unknown nonlinear function. By adding and subtracting Ax, (1) becomes

$$\begin{aligned} \dot{x}(t) &= Ax + g(x,u) \\ y(t) &= Cx(t) \end{aligned} \tag{2}$$

where A is Hurwitz matrix and $g(x,u) = f(x,u) - Ax$. Then the observer model can be selected as

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x} + \hat{g}(\hat{x},u) + G(y - C\hat{x}) \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{3}$$

where $\hat{x}$ denotes the states of the observer and the observer gain $K \in R^{n\times m}$ is selected such that $A - GC$ is a Hurwitz matrix. The structure of the observer is shown in Figure (1). The key to designing a neuro-observer is to employ a neural network to identify the nonlinearity and a conventional observer to estimate the states. It was shown by many researchers (e.g. [9] and [10]) that for x restricted to a compact set S of $x \in R^n$ and for some sufficiently large number of hidden layer neurons, there exist weights and thresholds such that any continuous function on the compact set S can be represented as

$$g(x,u) = W^T\sigma(V^T\bar{x}^T) + \epsilon(x) \tag{4}$$

where W and V are the weight matrices and $\bar{x} = [x \ u]$, $\sigma(.)$ is the transfer function of the hidden neurons that usually is considered as a sigmoidal function and $\epsilon(x)$ is the neural network approximation error.

$$\sigma(\bar{x}) = \frac{2}{1 + exp^{-2\bar{x}}} - 1 \tag{5}$$

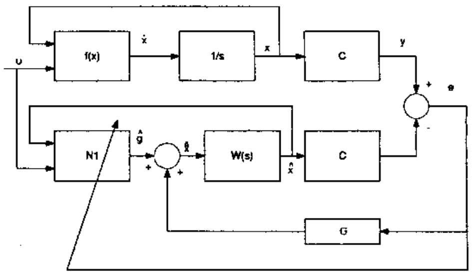

Figure 1: The structure of the proposed neural network observer.

The so-called linear in parameter neural network (LPNN) is obtained by fixing the weights of the first layer as $V = I$. Then the model can be expressed as

$$g(x,u) = W^T\sigma(\bar{x}) + \epsilon \tag{6}$$

Thus, the function g can be approximated by LPNN as

$$\hat{g}(\hat{x},u) = \hat{W}^T\sigma(\hat{\bar{x}}) \tag{7}$$

The proposed observer is then given by

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x} + \hat{W}^T\sigma(\hat{\bar{x}}) + G(y - C\hat{x}) \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{8}$$

Defining the state estimation error as $\tilde{x} = x - \hat{x}$ and by using (2), (7) and (8), the error dynamics can be expressed as

$$\begin{aligned} \dot{\tilde{x}}(t) &= A_c\tilde{x} + \tilde{W}^T\sigma(\hat{\bar{x}}) + w(t) \\ \tilde{y}(t) &= C\tilde{x}(t) \end{aligned} \tag{9}$$

where $\tilde{W} = W - \hat{W}$, $w(t) = W^T[\sigma(\bar{x}) - \sigma(\hat{\bar{x}})]$ is a bounded disturbance term i.e, $\|w(t)\| \leq \bar{w}$ for some positive constant $\bar{w}$, due to the sigmoidal function.

2.1. Stability Analysis

Backpropagation is widely used in classification, recognition, identification and control problems and it is shown to have promising results [11] and [12]. However, the main drawback of the previous works is the lack of mathematical proof of stability. The essential idea of the most stable neural network observers is to select the learning rule (the weight updating mechanism) such that the time derivative of the selected Lyapunov function becomes negative definite whereas in backpropagation, the learning rule is already set. It is proposed here to modify the backpropagation algorithm according to e-modification. The stability of the overall system is shown by Lyapunov direct method. The main results of the paper is now given in the following theorem.

Theorem 1 *Consider the plant model (1) and the observer model (8). If the weights of the LPNN are updated according to*

$$\dot{\hat{W}} = -\eta(\frac{\partial J}{\partial \hat{W}})^T - \rho\|\tilde{y}\|\hat{W} \quad (10)$$

where $\eta > 0$ is the learning rate, $J = \frac{1}{2}(\tilde{y}^T\tilde{y})$ is the objective function and ρ is a small positive number, then $\tilde{x}, \tilde{W}, \tilde{y} \in L_\infty$

The first term in (10) is the back propagation term and the second one is the e-modification term.

proof: First, note that $\frac{\partial J}{\partial \hat{W}}$ can be computed according to

$$\frac{\partial J}{\partial \hat{W}} = \frac{\partial J}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial \hat{W}} = -\tilde{y}^T C\frac{\partial \hat{x}}{\partial \hat{W}} \quad (11)$$

Now, by using the static approximation of $\frac{\partial \hat{x}}{\partial \hat{W}}$ as $-A_c^{-1}\frac{\partial g}{\partial \hat{W}}$ and by using (7) and (9), the learning rule (11) may be written as

$$\dot{\hat{W}} = \eta(\tilde{x}^T C^T C A_c^{-1})^T(\sigma(\hat{\bar{x}}))^T + \rho\|C\tilde{x}\|\hat{W} \quad (12)$$

Consider the Lyapunov function candidate as

$$V = \frac{1}{2}\tilde{x}^T p\tilde{x} + \frac{1}{2}tr(\tilde{W}^T\rho^{-1}\tilde{W}) \quad (13)$$

where $p = p^T > 0$. The time derivative of (13) is given by

$$\dot{V} = \frac{1}{2}\dot{\tilde{x}}^T p\tilde{x} + \frac{1}{2}\tilde{x}^T p\dot{\tilde{x}} + tr(\tilde{W}^T\rho^{-1}\dot{\tilde{W}}) \quad (14)$$

Lyapunov equation states that for a Hurwitz matrix A_c and a symmetric positive definite matrix p, there exists a symmetric positive definite matrix Q such that

$$A_c^T p + pA_c = -Q \quad (15)$$

Now, By substituting (9), (15), and (12) in to (14), one can get

$$\begin{aligned} \dot{V} &= -\frac{1}{2}\dot{\tilde{x}}^T Q\tilde{x} + \tilde{x}^T p(\tilde{W}\sigma(\hat{\bar{x}}) + w) \\ &+ tr(-\tilde{W}^T l\tilde{x}\sigma^T + \tilde{W}^T\|C\tilde{x}\|(W - \tilde{W})) \quad (16) \end{aligned}$$

where $l = \eta\rho^{-1}A_c^{-T}C^T C$. We have

$$\begin{aligned} tr(\tilde{W}^T(W - \tilde{W}) &\leq W_M\|\tilde{W}\| - \|\tilde{W}\|^2 \quad (17) \\ tr(\tilde{W}^T l\tilde{x}\sigma^T) &\leq \sigma_m\|\tilde{W}^T\|\,\|l\|\,\|\tilde{x}\| \quad (18) \end{aligned}$$

where $W_M = sup(W)$ and $\sigma_m = sup(\sigma)$. Note that, (17) is obtained using the following fact

$$tr(\tilde{W}^T l\tilde{x}\sigma^T) = \sigma^T\tilde{W}^T l\tilde{x} \quad (19)$$

Now by using (17) and (18), one can get

$$\begin{aligned} \dot{V} &\leq -\frac{1}{2}\lambda_{min}(Q)\|\tilde{x}\|^2 + \|\tilde{x}\|\|p\|(\|\tilde{W}\|\sigma_m + \bar{w}) \\ &+ \sigma_m\|\tilde{W}\|\|l\|\|\tilde{x}\| \\ &+ (W_M\|\tilde{W}\| - \|\tilde{W}\|^2)\|C\|\|\tilde{x}\| \quad (20) \end{aligned}$$

Furthermore by completing the squares $\|\tilde{W}\|$, we obtain the following condition for $\dot{V}$ to be negative definite.

$$\begin{aligned} \|\tilde{x}\| &\geq (2\|p\|\bar{w} + (\sigma_m\|p\| \\ &+ W_M\|C\| + \sigma_m\|l\|)^2/2)/\lambda_{min}(Q) \quad (21) \end{aligned}$$

Having the time derivative of V a negative definite function, the results of the theorem follow immediately.

Remark: In many cases, not all system states directly appears in the output of the system. Hence, some elements of C would be zero and this will slow down the learning process because of the structure of the backpropagation algorithm (see Equation (12)). It is suggested that for the purpose of the training only, output matrix C is modified as C_1 such that as if all states appear in the output of the system and directly contribute to the observation error. Therefore, (12) will be changed to

$$\dot{\hat{W}} = -\eta(\tilde{x}^T C^T C_1 A_c^{-1})^T(\sigma(\hat{x}))^T + \rho\|C\tilde{x}\|\hat{W} \quad (22)$$

and l is defined as:

$$l = \eta\rho^{-1}A_c^{-T}C_1^T C \quad (23)$$

The rest of the proof and the conditions remain unchanged.

3. MANIPULATOR MODEL

Manipulators with harmonic drive actuators or motors with long shafts tends to have inherent joint flexibilities. The most common way to model the joint flexibility is to consider a rotational spring between the input shaft (motor) and the output shaft (link) of the manipulator [13].

$$\begin{aligned} D_l(q_1)\ddot{q}_1 &+ C_1(q_1, \dot{q}_1) + g(q_1) + B_1\dot{q}_1 = \tau_s, \\ J\ddot{q}_2 &+ \tau_s + B_2\dot{q}_2 = \tau, \quad (24) \end{aligned}$$

where $q_1 \in \Re^n$ is the vector of links positions, $q_2 \in \Re^n$ is the vector of motor shaft positions, $g(q_1) \in \Re^n$ is the gravity loading force, $C_1(q_1, \dot{q}_1) \in \Re^n$ is the term corresponding to the centrifugal and Coriolis forces, $B_1 \in \Re^{n\times n}$ and $B_2 \in \Re^{n\times n}$ are the viscous damping matrices at the output and input shafts, $D_l(q_1) \in \Re^{n\times n}$ and $J \in \Re^{n\times n}$ are the robot and the actuator inertia matrices respectively, and τ is the input torque. The reaction torque τ_s from the rotational spring is often considered as

$$\tau_s = K(q_2 - q_1) + \beta(q_1, \dot{q}_1, q_2, \dot{q}_2) \quad (25)$$

where $K \in \Re^{n\times n}$ is the positive-definite stiffness matrix of the rotational spring attached between the input and the output shafts. In general, there is an unknown nonlinear force $\beta(q_1, \dot{q}_1, q_2, \dot{q}_2)$ which can be regarded as a combination of a nonlinear spring and the friction at the output shafts of the manipulator. The reaction torque τ_s cannot be modeled accurately and is assumed to be unknown for observer design and is included for simulation purposes only.

4. SIMULATION RESULTS

The performance of the proposed observer is investigated on a single-link and two-link flexible-joint manipulators.

4.1. A Single-link Flexible-Joint Manipulator

First, the neural network observer was simulated on a manipulator whose parameters are: $J = 1.16 kg.m^2$, $m = 1Kg$, $l = 1m$, $K = 100N/m$, where J is the motor inertia, m is the link mass, l is the link length, and K is the stiffness of the joint. A two-layer neural network was used with 5 neurons in the input layer, 4 neurons in the output layer. The controller and observer parameters are: $A = -100I$, $\eta = 1000$, $\rho = 0.01$, and

$$G = \begin{bmatrix} 3.0 & -0.5 \\ -5.0 & 2.0 \\ -0.5 & 2.0 \\ 0.5 & 1.0 \end{bmatrix}$$

The inputs to the network are $u, \hat{q}_1, \dot{\hat{q}}_1, \hat{q}_2$, and $\dot{\hat{q}}_2$. The input layer neurons have sigmoidal transfer functions and the output neurons use linear activation functions. The initial weights of the network are selected as small random numbers. Figure 2 shows the simulation results for this case. Figure 2–a shows the responses of q_2 and $\hat{q}_2$ and Figure 2–b shows those of q_4 and $\hat{q}_4$. Figure 2–c and 2–d show the responses of the system after the learning period. It is clear that the states of the observer follow those of the actual system. In the next simulation, a simulation was done to evaluate the ability of the neural network to work as an off-line training scheme. For this simulation, the weights obtained from the last simulation were used as initial weights and the network was used in recall mode. The simulation results in this case are shown in Figures 5-a and 5-b. Note that, the weights are not updated in this case. As can be seen, the neural network is still able to follow the states of the flexible-joint manipulator. These results demonstrate that the neural network has learned the system dynamics.

4.2. A Two-link Flexible-Joint Manipulator

Simulation results for a two–link planar manipulator are presented in this section. The dynamics of a two-link manipulator are far more complicated than those of a single-link manipulator. The manipulator consists of two flexible-joints with the following numerical data

$$J = diag\{1.16, 1.16\},\ m = diag\{1, 1\},\ l_1 = l_2 = 1m,$$
$$K = diag\{100, 100\},\ A = -10I,\ \eta = 100,\ \text{and}\ \rho = 0.01.$$

The Neural network has two layers, the first layer has 10 neurons with sigmoidal activation functions and the second layer has 8 neurons with linear transfer functions. The inputs of the network is $\bar{x}$. Figure 4 shows the result of the state estimation. As can be observed, despite the increased complexity in the manipulator model, the neural network has learned the manipulator dynamics, and all the states of the neural networks track the corresponding states of the system.

5. CONCLUSIONS

A stable neuro-observer for a general MIMO nonlinear model was considered in this paper. The weight updating mechanism is based on the modified backpropagation algorithm. The modification was based on the e-modification that guarantees the boundedness of the estimation error. The stability of the overall system was proved by using the Lyapunov direct method. The SPR condition on output error equation was also relaxed. Simulation results on a single-link and two-link flexible-joint manipulator was also presented to evaluate the performance of the proposed observer. The proposed scheme can be used as an online as well as offline observer. The extension of this approach to the case of nonlinearly parameterized neural networks is under investigation.

6. REFERENCES

[1] S. Nicosia, P. Tomei, and A. Tornambe, "An approximate observer for a class of nonlinear systems," *System and Control Letters*, vol. 12, pp. 43–51, 1989.

[2] S. Nicosia and A. Tornambe, "High-gain observer in the state and parameter estimiation of robots having elastic joints.," *System and Control Letters*, vol. 13, pp. 331–337, 1989.

[3] P.B Goldsmith, B.A. Francis and A.A. Goldenderg, "Stability of hybrid position/force control applied to manipulators with flexible joints," *International Journal of Robotics and Automation* ,Vol. 14,No., pp. 146–159, 1999.

[4] Y.H. Kim, F.L. Lewis and C.T. Abdallah, "Nonlinear observer design using dynamic recurrent neural networks," in *Proceedings of the 35th Conference on Decision and Control, Kobe, Japan*, pp. 949–954, December 1996.

[5] A.S. Poznyak, E.N. Sanchez, O. Palma and W. Yu, "Output trajectory tracking using dynamic neural networks," in *Proceedings of the 39th IEEE Conference on Decision and Control*, pp. 889–894, December 2000.

[6] J.A.R. Vargas and E.M. Hemerly, "Neural adaptive observer for general nonlinear systems," in *Proceedings of American Control Conference (ACC)*, pp. 708–712, 2000.

[7] J.A.R. Vargas and E.M. Hemerly, "Robust neural adaptive observer for MIMO nonlinear systems," in *Proceedings of IEEE Conference on Systems, Man, and Cybernetics, IEEE SMC' 99*, pp. 1084–1089, 1999.

[8] H.A. Talebi, R.V. Patel and M. Wong, "A neural network based observer for flexible-joint manipulators," in *Proceedings of the 15th IFAC World Congress on Automatic Control, Barcelona, Spain*, July 2002.

[9] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Proceedings of the Int. Joint Conf. on Neural Networks*, pp. 593–605, 1989.

[10] G. Cybenko, "Approximation by superposition of sigmoidal functions," *Mathematics of Control, Signals and Systems*,, pp. 303–314, October 1989.

[11] K.S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, pp. 4–27, March 1990.

[12] H.A. Talebi, R.V. Patel and H.Asmer, "Neural network based dynamic modeling of flexible-link manipulators with application to the ssrms," in *Journal of Robotic systems*, pp. 385–401, 17 2000.

[13] M. Spong and M. Vidyasagar, *Robot Dynamics and Control*. McGraw Hill, 1989.

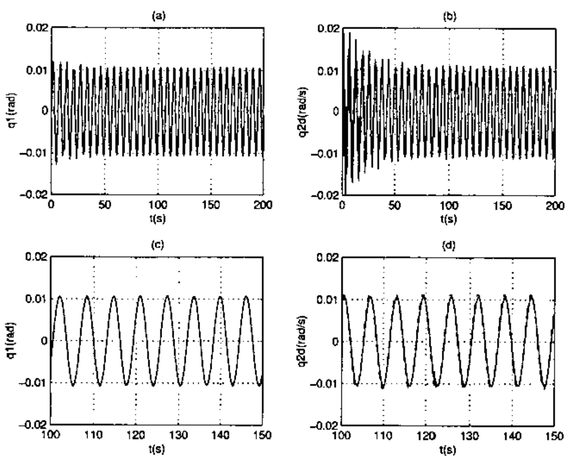

Figure 2: The responses of the single-link flexible-joint manipulator to $\sin(t)$ reference trajectory for LPNN: (a) link position, (b) link velocity (c) link position after learning period, (d) link velocity after learning period. The solid lines correspond to the actual states and the dashed lines correspond to the states of the observer.

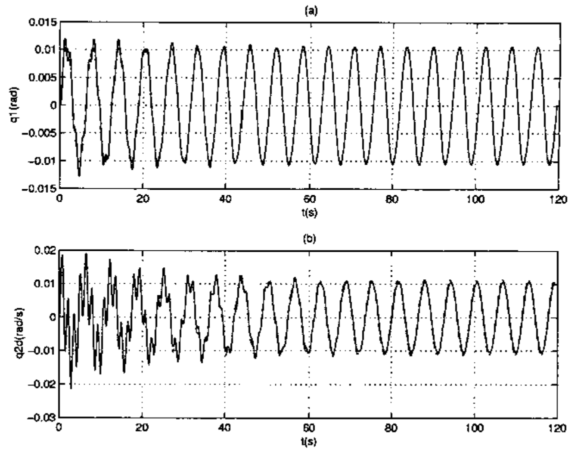

Figure 3: The responses of the single-link flexible-joint manipulator to $\sin(t)$ reference trajectory for LPNN after the learning stops: (a) link position, (b) link velocity. The solid lines correspond to the actual states and the dashed lines correspond to the states of the observer.

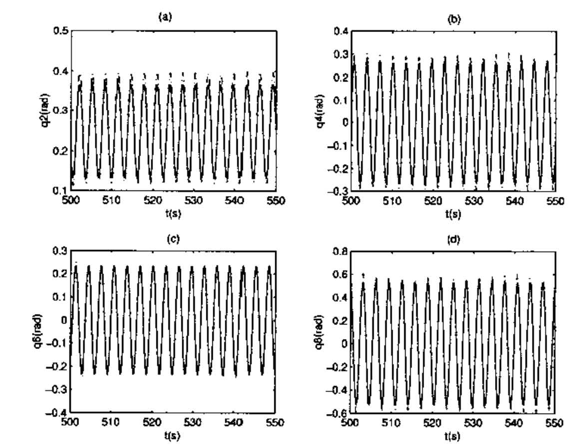

Figure 4: The responses of the two-link flexible-joint manipulator (after the learning period) to $sin(t)$ reference trajectory for LPNN: (a) The position of the first link , (b) the velocity of the first link (c) the position of the second link , (d) the velocity of the second link. The solid lines correspond to the actual states and the dashed lines correspond to the states of the observer.