PAPER • OPEN ACCESS

Position and Velocity Estimation Using TOA and FOA Based on Lagrange Programming Neural **Network**

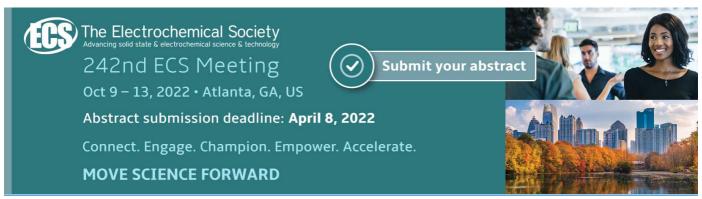
To cite this article: C Jia et al 2019 J. Phys.: Conf. Ser. 1284 012064

View the article online for updates and enhancements.

You may also like

- Analysis of displacement damage effects on bipolar transistors irradiated by spallation neutrons
 Yan Liu, , Wei Chen et al.
- Computational models to delineate 3D gaze-shift strategies in Parkinson's
- Palak Gupta, Sinem Beylergil, Jordan Murray et al.
- TURBULENT TRANSPORT IN A STRONGLY STRATIFIED FORCED SHEAR LAYER WITH THERMAL **DIFFUSION**

Pascale Garaud and Logithan Kulenthirarajah



Position and Velocity Estimation Using TOA and FOA Based on Lagrange Programming Neural Network

C Jia^{1,2,*}, J Yin^{1,2}, Z Yang^{1,2} and L Zhang^{1,2}

¹National Digital Switching System Engineering and Technology Research Center, Zhengzhou 450002, P.R. China

²Zhengzhou Information Science and Technology Institute, Zhengzhou, Henan 450002, P.R. China

*E-mail: Kingsway0221@163.com

Abstract. This paper addresses the problem of estimating the position and velocity of a moving source utilizing the time-of-arrival (TOA) and frequency-of-arrival (FOA) measurements. Since the concerned estimation problem is highly non-linear and non-convex, we propose to utilize a novel neural circuit, namely the Lagrange programming neural network (LPNN) framework, to solve this problem. LPNN equips the abilities of fast convergence and the robustness of resisting high noise level, and thus these two advantages have drawn much attention for it recently. Since LPNN is able to solve the optimization problem with constraint, we first reformulate the original non-linear and non-convex maximum likelihood (ML) problem by introducing additional variables and constraints, and thus a neural network is built up based on the LPNN framework. Subsequently, the convergence and stability of the proposed neural network is mathematically proved and then verified by the results of numerical experiments. Different from the conventional numerical algorithms, the analog neural network can be utilized to fulfil the task of real-time calculation, especially when there are limited computation resources in some applications. The simulation results demonstrate that the proposed LPNN model equips the basic properties of convergence and stability, and also show the superior localization accuracy of the proposed method than other numerical algorithms.

1. Introduction

In recent years, passive source localization has attracted many attentions for its wide applications in communications, radar and others [1-3]. Among the positioning parameters used so far, the time-of-arrival (TOA) measurements can be utilized to locate a stationary radiation source with high precision [1, 2]. When there is relative movement between the receiving sensors and the radiation source, the frequency of arrival (FOA), combined with the TOAs, can be incorporated to estimate the source position and velocity. Theoretically, these two types of parameters can be obtained simultaneously when the carrier frequency of the signal can be obtained, and the clock synchronization between receivers and the radiation source can be achieved [2]. Under these two assumptions, the hybrid of TOA and FOA measurements can improve the localization accuracy [2]. Due to the possibility of jointing and its corresponding advantages, TOA and FOA joint positioning has received intensive investigations in geolocation based on communication satellites, direct position determination and sensor network positioning [1, 2].

It is not easy to solve the hybrid of TOA and FOA localization problem due to its high nonlinearity

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

and non-convexity. Therefore, several types of methods have been utilized to tackle such nonlinear and non-convex problems [2-7]. The author in [3] proposes an iterative method based on Taylor-series to solve this problem. In general, the iterative methods are computationally expensive. To alleviate this shortcoming, algorithms with a closed form solution are developed based on the least squares criterion [4, 5]. However, when the measurement noise in the received signal is large, the second-order noise terms ignored in the two methods will significantly degrade the positioning performance of these algorithms. To obtain the optimal solution at a high noise level, the convex optimization technique is introduced to address this problem [6, 7]. But as shown in [7], the estimated result is sub-optimal if the relaxation applied to the original ML problem is not appropriate. More recently, a Monte Carlo importance sampling algorithm based on Pincus theorem is designed in [2] to generate an approximate global solution. However, this algorithm can only yield an optimal solution at moderate noise levels.

Unlike the numerical algorithms listed above, using neural circuit has attract many attentions since it can achieve real-time calculation through hardware implementation [8]. Among a series of neural networks proposed successively [8-10], the Lagrange programming neural network (LPNN) can solve a variety of optimization problems with constraints, and thus it has been applied in many fields [11-17], including source localization [13-17]. The authors in [13] and [17] first propose to use TOA (or TDOA) measurements based on LPNN framework for localization when only the measurement noise is considered. Then this method is employed to handle the outlier data [14] and a more sophisticated scenario in TOA positioning [15, 16]. These previous studies show the excellent performance of LPNN, but they do not consider the case where the radiation source and receivers are not stationary. Besides, the hybrid of TOA and FOA can be used to improve the positioning accuracy as mentioned before.

Therefore, based on the LPNN framework, we advocate to use TOA and FOA measurements to estimate the source position and velocity simultaneously. This is of great value for locating a moving source with high calculation speed, especially in applications that have limited computing resources, e.g., the satellite-based geolocation. Besides, note that the authors in [13, 14] simply ignore the inequality constraints in the process of reformulating the original ML problem. Instead, we advocate to introduce additional variables to convert inequality constraints into equalities as executed in [8, 16, 17], which can approximate the original maximum likelihood (ML) cost function more closely. Moreover, it is of great essential for a network of practical sense to equip the property of fast convergence, as well as the ability of maintaining its stable output when the neural network converges. Therefore, the convergence and stability of our model is analytically proved based on the analysis in [8]. These two properties of the proposed neural network, as well as its superior estimation performance, are verified by the simulation results.

The rest of the paper is organized as follows. First, the localization scenario and a TOA&FOA based model for moving source positioning are given in Section II. Section III constructs the proposed neural network and algorithm according to the LPNN framework, and the stability and computational complexity of the network is also analyzed. Then we conduct several simulation tests to examine the two basic properties and estimation performance of the proposed algorithm in Section IV. Finally, the conclusions are drawn in Section V.

2. Problem formulation

We consider the general problem of locating a moving source using TOA and FOA measurements in a sensor network. There are M sensors deployed in the D-dimensional (D = 2 or 3) space to determine the unknown positon u and velocity \dot{u} of a mobile source. The position and velocity of i^{th} , i = 1, 2, ..., M sensor is accurately known, which is denoted by s_i and \dot{s}_i . For simplicity, we assume line of sight (LOS) propagation. Thus the LOS TOA measurements with additive measurement noise e_i are given by

$$t_i = \frac{1}{c} \| \boldsymbol{u} - \boldsymbol{s}_i \| + e_i , \quad i = 1, 2, ..., M ,$$
 (1)

where $\|\cdot\|$ is Euclidean norm and c represents the signal propagation speed. Multiplying the both sides of (1) by constant c, the range of arrival (ROA) measurements can be obtained as

$$d_{i} = \|\mathbf{u} - \mathbf{s}_{i}\| + n_{i}, \quad i = 1, 2, ..., M$$
 (2)

where $n_i = c \cdot e_i$. Taking the time derivation of (2) yields the FOA measurements:

$$\dot{d}_{i} = \frac{(u - s_{i})^{T} (\dot{u} - \dot{s}_{i})}{\|u - s_{i}\|} + \dot{n}_{i}, \quad i = 1, 2, ..., M.$$
(3)

Without loss of generality, the measurement noises $\mathbf{n} = [n_1, n_2, ..., n_M]^T$ and $\dot{\mathbf{n}} = [\dot{n}_1, \dot{n}_2, ..., \dot{n}_M]^T$ are assumed to be uncorrelated. Moreover, these two types of noises follow the Gaussian distribution with zero means, whose covariance are given by $\mathbf{E}(\mathbf{n} \cdot \mathbf{n}^T) = \mathbf{Q}_{\alpha}$ and $\mathbf{E}(\dot{\mathbf{n}} \cdot \dot{\mathbf{n}}^T) = \mathbf{Q}_{\beta}$, respectively. Therefore, based on the above assumptions, the ML estimation of the source position and velocity can be directly computed as

$$\{\hat{\boldsymbol{u}}, \hat{\boldsymbol{u}}\} = \arg \min_{\boldsymbol{u}, \hat{\boldsymbol{u}}} \frac{1}{2} (\boldsymbol{d} - \boldsymbol{d}^{o})^{\mathrm{T}} \boldsymbol{Q}_{\alpha}^{-1} (\boldsymbol{d} - \boldsymbol{d}^{o}) + \frac{1}{2} (\dot{\boldsymbol{d}} - \dot{\boldsymbol{d}}^{o})^{\mathrm{T}} \boldsymbol{Q}_{\beta}^{-1} (\dot{\boldsymbol{d}} - \dot{\boldsymbol{d}}^{o})$$
(4)

where $\mathbf{d} = [d_1, d_2, ..., d_M]^T$ and $\dot{\mathbf{d}} = [\dot{d}_1, \dot{d}_2, ..., \dot{d}_M]^T$. Besides, $\mathbf{d}^o = [d_1^o, d_2^o, ..., d_M^o]^T$, where $d_i^o = \|\mathbf{u} - \mathbf{s}_i\|$, and $\dot{\mathbf{d}}^o = [\dot{d}_1^o, \dot{d}_2^o, ..., \dot{d}_M^o]^T$, where $\dot{d}_i^o = (\mathbf{u} - \mathbf{s}_1)^T (\dot{\mathbf{u}} - \dot{\mathbf{s}}_1) / d_i^o$.

The problem of interest is to find the optimal estimates of u and \dot{u} , which minimizes the ML cost function in (4) by making use of the available noisy measurements of TOA and FOA.

3. The proposed method

According to [8], the LPNN is designed to solve constrained optimization problems. This method defines two kinds of neurons, that is, the variable neurons and the Lagrange neurons. Moreover, these two types of neurons represent the optimization variables and the Lagrangian multipliers, respectively. Based on the original constrained optimization problem and the neurons defined, a Lagrangian function is firstly established. Then the differential dynamic equations of neurons are calculated from the Lagrangian function, and a self-feedback neural network model can be constructed based on these dynamic equations. During the working processing of the network, these equations will govern the network state until it reaches an equilibrium point. More detailed introduction of LPNN can be found in [8], and here we illustrate this method through the construction of the proposed neural network.

3.1. LPNN Model for TOA and FOA Based Localization.

When the measurement noise at different sensors are independent from each other, the measurement noise covariance matrices $\mathbf{Q}_{\alpha} = diag(\sigma_1^2, \sigma_2^2, ..., \sigma_M^2)$ and $\mathbf{Q}_{\beta} = diag(\dot{\sigma}_1^2, \dot{\sigma}_2^2, ..., \dot{\sigma}_M^2)$. Here $diag\{\cdot\}$ is the diagonal operator. Therefore, the cost function in (4) can be equivalently written as

$$\{\hat{\boldsymbol{u}}, \hat{\boldsymbol{u}}\} = \arg \min_{\boldsymbol{u}, \hat{\boldsymbol{u}}} \sum_{i=1}^{M} \frac{1}{2\sigma_{i}^{2}} (d_{i} - \|\boldsymbol{u} - \boldsymbol{s}_{i}\|)^{2} + \sum_{i=1}^{M} \frac{1}{2\dot{\sigma}_{i}^{2}} (\dot{d}_{i} - \frac{(\boldsymbol{u} - \boldsymbol{s}_{i})^{T} (\dot{\boldsymbol{u}} - \dot{\boldsymbol{s}}_{i})}{\|\boldsymbol{u} - \boldsymbol{s}_{i}\|})^{2},$$
 (5)

In order to apply the LPNN framework, the above unconstrained problem need to be reformulated as a constrained one, and thus we introduce the dummy variable $g_i = \|\mathbf{u} - \mathbf{s}_i\|$ to the original problem in (5). Therefore, it is straightforward to obtain the following constrained optimization problem:

$$\min_{\boldsymbol{u}, \dot{\boldsymbol{u}}} \sum_{i=1}^{M} \left\{ \frac{1}{2\sigma_{i}^{2}} (d_{i} - g_{i})^{2} + \frac{1}{2\dot{\sigma}_{i}^{2}} (\dot{d}_{i} - \frac{(\boldsymbol{u} - \boldsymbol{s}_{i})^{T} (\dot{\boldsymbol{u}} - \dot{\boldsymbol{s}}_{i})}{g_{i}})^{2} \right\},$$
s.t. $g_{i}^{2} = \|\boldsymbol{u} - \boldsymbol{s}_{i}\|^{2}$,
$$g_{i} > 0, \quad i = 1, 2, ..., M.$$
(6)

Remark 1: It is worthy to point out that in order to guarantee the validity of the objective function, g_i should not equal to 0, thus the original constraint $g_i \ge 0$ is further required to be $g_i > 0$. Hence,

we have $g_i = ||\mathbf{u} - \mathbf{s}_i|| \neq 0$, which means that the location of source should not be the same as one of the sensor positions. In practice, this condition can be met in many localization scenarios.

As described in [8], the inequality constraints in (6) can be transferred into equality constraints by introducing additional variables y_i , i=1,2,...,M. That is, to ensure that the non-negative property of g_i , we advocate to let $g_i = \alpha_i \cdot y_i^2$, α_i is a constant and usually $\alpha_i \ge 1$. Moreover, we define $\mathbf{g} = [g_1, g_2, ..., g_M]^T$, and $\mathbf{y} = [y_1, y_2, ..., y_M]^T$ is the additional variable vector. $\mathbf{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_M]^T$ and $\mathbf{\mu} = [\mu_1, \mu_2, ..., \mu_M]^T$ are the Lagrange multipliers. Hence, the problem in (6) can be further recast as

$$\min_{\boldsymbol{u}, \dot{\boldsymbol{u}}, g} \sum_{i=1}^{M} \left\{ \frac{1}{2\sigma_{i}^{2}} (d_{i} - g_{i})^{2} + \frac{1}{2\dot{\sigma}_{i}^{2}} (\dot{d}_{i} - \frac{(\boldsymbol{u} - \boldsymbol{s}_{i})^{T} (\dot{\boldsymbol{u}} - \dot{\boldsymbol{s}}_{i})}{g_{i}})^{2} \right\},$$
s.t.
$$g_{i}^{2} = \|\boldsymbol{u} - \boldsymbol{s}_{i}\|^{2},$$

$$g_{i} = \alpha_{i} \cdot y_{i}^{2}, \quad \alpha_{i} \geq 1, \quad i = 1, 2, ..., M.$$
(7)

Based on (7), a Lagrange function can be directly computed as

$$\mathcal{L}(\boldsymbol{u}, \dot{\boldsymbol{u}}, \boldsymbol{g}, \boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{i=1}^{M} \left\{ \frac{1}{2\sigma_{i}^{2}} (d_{i} - g_{i})^{2} + \frac{1}{2\dot{\sigma}_{i}^{2}} (\dot{d}_{i} - \frac{(\boldsymbol{u} - s_{i})^{T} (\dot{\boldsymbol{u}} - \dot{s}_{i})}{g_{i}})^{2} \right\} + \sum_{i=1}^{M} \left[\lambda_{i} (g_{i}^{2} - \|\boldsymbol{u} - s_{i}\|^{2}) + \mu_{i} (g_{i} - \alpha_{i} \cdot y_{i}^{2}) \right]$$
(8)

However, our previous simulation experiments show that the LPNN model directly established by (8) is not satisfactory in terms of convergence speed and stability, as well as the final equilibrium output. Such kind of problem also appears in [12,14]. Therefore, we add penalty parameter C_0 and augmented term $\mathcal{A}(u,g,y) = (C_0/2)\{\sum_{i=1}^{M}[(g_i^2 - ||u-s_i||^2)^2 + (g_i - \alpha_i \cdot y_i^2)^2]\}$ to the objective function in (7) for the sake of improving the convexity of it and further enhancing the stability of the LPNN model. Details can be found in [8].

Based on (7) and the augmented term, the LPNN framework can be applied since both the objective function and constraints are twice differentiable, and thus the augmented Lagrangian function is calculated as

$$\mathcal{L}_{a}(u,\dot{u},g,y,\lambda,\mu) = \mathcal{L}(u,\dot{u},g,y,\lambda,\mu) + \mathcal{A}(u,g,y). \tag{9}$$

According to the Lagrangian function established (9) and the LPNN framework described in [8], the dynamics of variable neurons can be directly computed as

$$\frac{d\mathbf{u}}{dt} = -\frac{\partial \mathcal{L}_a}{\partial \mathbf{u}} = \sum_{i=1}^{M} \frac{1}{\dot{\sigma}_i^2 \cdot g_i} [\dot{d}_i - \frac{(\mathbf{u} - \mathbf{s}_i)^{\mathrm{T}} (\dot{\mathbf{u}} - \dot{\mathbf{s}}_i)}{g_i}] (\dot{\mathbf{u}} - \dot{\mathbf{s}}_i) + \sum_{i=1}^{M} 2[C_0(g_i^2 - \|\mathbf{u} - \mathbf{s}_i\|^2) + \lambda_i] (\mathbf{u} - \mathbf{s}_i), \quad (10)$$

$$\frac{d\dot{\boldsymbol{u}}}{dt} = -\frac{\partial \mathcal{L}_a}{\partial \dot{\boldsymbol{u}}} = \sum_{i=1}^{M} \frac{1}{\dot{\sigma}_i^2 \cdot g_i} [\dot{d}_i - \frac{(\boldsymbol{u} - \boldsymbol{s}_i)^{\mathrm{T}} (\dot{\boldsymbol{u}} - \dot{\boldsymbol{s}}_i)}{g_i}] (\boldsymbol{u} - \boldsymbol{s}_i) , \qquad (11)$$

$$\frac{dg_{i}}{dt} = -\frac{\partial \mathcal{L}_{a}}{\partial g_{i}} = -\frac{1}{\dot{\sigma}_{i}^{2} g_{i}^{2}} \left[\dot{d}_{i} - \frac{\left(\boldsymbol{u} - \boldsymbol{s}_{i} \right)^{\mathrm{T}} \left(\dot{\boldsymbol{u}} - \dot{\boldsymbol{s}}_{i} \right)}{g_{i}} \right] \left[\left(\boldsymbol{u} - \boldsymbol{s}_{i} \right)^{\mathrm{T}} \left(\dot{\boldsymbol{u}} - \dot{\boldsymbol{s}}_{i} \right) \right]
- C_{0} \left[2g_{i} \left(g_{i}^{2} - \left\| \boldsymbol{u} - \boldsymbol{s}_{i} \right\|^{2} \right) + \left(g_{i} - \alpha_{i} \cdot y_{i}^{2} \right) \right] - \frac{1}{\sigma_{i}^{2}} \left(d_{i} - g_{i} \right) - 2\lambda_{i} \cdot g_{i} - \mu_{i}, \tag{12}$$

$$\frac{dy_i}{dt} = -\frac{\partial \mathcal{L}_a}{\partial y_i} = 2C_0(g_i - \alpha_i \cdot y_i^2) \cdot \alpha_i \cdot y_i + 2\mu_i \cdot \alpha_i \cdot y_i.$$
(13)

The dynamics of Lagrangian neurons can be obtained in the same way, and they are given by

$$\frac{d\lambda_i}{dt} = \frac{\partial \mathcal{L}_a}{\partial \lambda_i} = g_i^2 - \|\mathbf{u} - \mathbf{s}_i\|^2, \tag{14}$$

$$\frac{d\mu_i}{dt} = \frac{\partial \mathcal{L}_a}{\partial \mu_i} = g_i - \alpha_i \cdot y_i^2.$$
 (15)

By utilizing the LPNN framework, we can build up a neural network based on the above differential equations defined in (10)-(15). According to [10,13,14], the block diagram in Figure 1 shows the main structure of the neural network established. It can be noticed that the proposed neural

model is a self-feedback network which consists of 2+4M function units and integrators. Besides, there are 2D+2M variable neurons to hold parameters u,\dot{u},g and y. Moreover, 4M Lagrangian neurons are employed to hold λ and μ . According to the analysis and circuit implementation approach given in [10], the working process of the network can be summarized as follows:

- **Step 1:** Initialize the neurons $u, \dot{u}, g, y, \lambda, \mu$. Input the TOA and FOA measurements and set the integration time t = 0
- **Step 2:** The differential result of each neuron is first calculated and then transmitted to the integrator. The integrators will output the updated network states.
 - Step 3: Feedback the updated network states as inputs.
 - Step 4: Repeat steps 2 and 3 until the network converges to an equilibrium point.
 - Step 5: Take the steady output states of the network as the final estimation results.

The hardware realization of LPNN model is illustrated in [10,18]. For simplicity, as applied in [13-17], we advocate simulating the searching process of the designed neural network by employing the MATLAB ode solver to solve the differential equations (10)-(15). This is mainly because the purpose of our paper is to establish a TOA and FOA based neural network model for source localization, and further verify its stability and estimation performance. Therefore, the convergence and stability required for an asymptotical stable network are analyzed in the coming subsection.

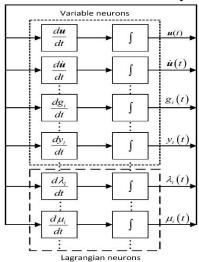


Figure 1. The structure of the proposed LPNN model.

3.2. Network convergence and stability analysis

To explain how the network can converge to the optimal solution, the saddle point property is employed to make an illustration in [8]. It is necessary to point out that our proposed LPNN model also satisfies the general analysis made in [8] since the problem considered meets the fundamental requirements of applying LPNN framework. Hence the convergence of network can be guaranteed. Then we prove that the designed neural network is asymptotically stable when the network converges to the minimum point (or equilibrium point), which means that the network can maintain the stable output state when it settles down at an equilibrium point from an arbitrary initial state within the attraction domain [8].

It is proved in [8] that when the gradient vectors of constraints in (7) are linearly independent and the Hessian matrix of the Lagrangian function in (8) is positive definite, then the equilibrium point $(\boldsymbol{\theta}^{*T}, \boldsymbol{\lambda}^{*T}, \boldsymbol{\mu}^{*T})^T$ is asymptotically stable, where $\boldsymbol{\theta}^* = (\boldsymbol{u}^{*T}, \dot{\boldsymbol{u}}^{*T}, \boldsymbol{g}^{*T}, \boldsymbol{y}^{*T})^T$. Here the property of linear independence is first examined. The equality constraints in (7) can be rewritten in the form:

$$h_i(\boldsymbol{\theta}) = g_i^2 - \|\boldsymbol{u} - \boldsymbol{s}_i\|^2,$$

$$h_{i+M}(\boldsymbol{\theta}) = g_i - \alpha_i \cdot y_i^2.$$
(16)

The gradient vectors of above functions at the equilibrium point are given by

$$\frac{\partial h_{i}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{*}} = [-2(\boldsymbol{u}^{*}-\boldsymbol{s}_{i}^{*})^{\mathrm{T}},\boldsymbol{\theta}_{1\times2},\boldsymbol{\theta}_{1\times(i-1)},2\boldsymbol{g}_{i}^{*},\boldsymbol{\theta}_{1\times(M-i)},\boldsymbol{\theta}_{1\times M}]^{\mathrm{T}},
\frac{\partial h_{i+M}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{*}} = [\boldsymbol{\theta}_{1\times4},\boldsymbol{\theta}_{1\times(i-1)},1,\boldsymbol{\theta}_{1\times(M-i)},\boldsymbol{\theta}_{1\times(i-1)},-2\boldsymbol{\alpha}_{i}\cdot\boldsymbol{y}_{i}^{*},\boldsymbol{\theta}_{1\times(M-i)}]^{\mathrm{T}}.$$
(17)

It is easy to check that if $g_i = \| \boldsymbol{u} - \boldsymbol{s}_i \| \neq 0$, that is to say $\boldsymbol{u} - \boldsymbol{s}_i \neq 0$, the gradient vectors $\partial h_1(\boldsymbol{\theta}^*)/\partial \boldsymbol{\theta}^*$, $\partial h_2(\boldsymbol{\theta}^*)/\partial \boldsymbol{\theta}^*$, $\partial h_M(\boldsymbol{\theta}^*)/\partial \boldsymbol{\theta}^*$, $\partial h_{M+1}(\boldsymbol{\theta}^*)/\partial \boldsymbol{\theta}^*$, $\partial h_{M+2}(\boldsymbol{\theta}^*)/\partial \boldsymbol{\theta}^*$, ..., $\partial h_{2M}(\boldsymbol{\theta}^*)/\partial \boldsymbol{\theta}^*$ are linearly independent. As stated in Remark 1, the sufficient condition that the source position is different from the sensor locations can be met in many localization scenarios. Thus, the linear independent property is established.

Then we show that the Hessian matrix of the proposed Lagrangian function is positive definite when the penalty parameter C_0 is taken sufficient large. According to (7)-(9), the augmented Lagrangian function can be re-expressed as

$$\mathcal{L}_{a}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\mu}) + \frac{C_{0}}{2} \|\boldsymbol{h}(\boldsymbol{\theta})\|^{2}.$$
(18)

where $h(\boldsymbol{\theta}) = [h_1(\boldsymbol{\theta}), h_2(\boldsymbol{\theta}), ..., h_M(\boldsymbol{\theta}), h_{1+M}(\boldsymbol{\theta}), h_{2+M}(\boldsymbol{\theta}), ..., h_{2M}(\boldsymbol{\theta})]^T$. Therefore, at the equilibrium point, the Hessian matrix of (18) can be obtained as

$$\nabla_{\theta\theta}^{2} \mathcal{L}_{a}(\theta^{*}, \lambda^{*}, \mu^{*}) = \nabla_{\theta\theta}^{2} \mathcal{L}(\theta^{*}, \lambda^{*}, \mu^{*}) + C_{0} \cdot \nabla h(\theta^{*}) \nabla h(\theta^{*})^{\mathrm{T}},$$
(19)

where $\nabla^2_{\theta\theta} \mathcal{L}(\theta^*, \lambda^*, \mu^*)$ is the Hessian matrix of (8). From (19), we can easily observe that the local convexity of $\mathcal{L}(\theta, \lambda, \mu)$ around the equilibrium point can be remedied when the constant penalty parameter C_0 is set sufficiently large. Hence we have $\nabla^2_{\theta\theta} \mathcal{L}_a(\theta^*, \lambda^*, \mu^*) > 0$. Combine the above two proofs, we can reach the asymptotical stability of the neural network designed.

3.3. Complexity analysis

In this subsection, the computational complexity of the designed algorithm is analyzed. For comparison, we also include the analysis for other three classic numerical algorithms, such as Taylor-series (TS) method [3], weighted least square (WLS) algorithm extended from the algorithm compared in [5] and Monte Carlo importance sampling (MCIS) [2].

As put in [11,12,17], the main computational complexity of using neural network lies in the calculation of the time derivatives defined by the dynamic equations in (10)-(15). Typically, considering that there are M sensors in a K dimensional scenario, the complexity of calculating \mathbf{Q}_{α}^{-1} to obtain $1/\sigma_i^2$ is utmost $O(M^2)$. Similarly, the complexity of calculating \mathbf{Q}_{β}^{-1} to obtain $1/\dot{\sigma}_i^2$ is $O(M^2)$. When obtaining the reciprocal of noise covariance, the complexity of computing the differential equations in (10)-(15) is roughly $MK^3 + MK^2 + MK + 20M$, and thus the computational complexity of our model is approximately $O(2M^2) + MK^3 + MK^2 + MK + 20M$ in a single iteration, as given in Table 1.

The main computational complexities of the other three classical numerical algorithms in a single iteration are also provided in Table 1. We can clearly see from Table 1 that the computational complexity of the TS method and the WLS algorithm is comparable, and the MCIS algorithm is computationally high due to the number of sampling points employed in this method. In contrast, the algorithm designed in this paper has a relatively low computational complexity compared with the other three algorithms. This is mainly due to the difference in methods, and the use of neural network has some certain computational advantages.

Table 1. Comparison of computational complexity.

Algorithms	Computational complexity			
LPNN	$O(2M^2) + MK^3 + MK^2 + MK + 20M$			
WLS	$O(2M^2)+O(K^3)+M^2+4M^2(K+2)+3M(K+2)^2+5MK$			

MCIS
$$O(2M^{2})+O(8K^{3})+8K^{2}M+8KM^{2}+4KM+4K^{2}N+4KN+2M,$$
 where N is the number of sampling points.
TS method
$$O(2M^{2})+O(8K^{3})+8M^{2}K+8MK^{2}+9MK$$

4. Simulation results

In this section, we first use noise-free and noisy TOA and FOA measurements to examine the convergence and stability of the designed network, and then we perform several simulations to test the localization performance of the network. In the simulation, we use Matlab ode solver to simulate the network dynamic process, which is governed by a group of differential equations defined in (10)-(15). The initial states of the variables are randomly generated between 0 and 10. Besides, the penalty parameter C_0 is set as 5, which is sufficient to guarantee the local convexity. The positive constants α_i , i = 1, 2, ..., M, are set as 100 to accelerate the convergence speed [8].

4.1 Verification of convergence and stability

4

-300

In this subsection, we use four sensors deployed in the 2-dimension plane to locate a moving source at the position u = (600, -400) m with velocity $\dot{u} = (30, -20)$ m/s. The positions and velocities of the four sensors are listed in Table 2. It is worthy to point out that our method can be directly generalized to a 3-dimension scenario, though a 2-dimension scenario is considered here for simplicity.

Sensor no.	Sensor position (m)		Sensor velocity (m/s)	
	x	у	v_x	v_y
1	-300	200	10	-10
2	300	-200	20	10
3	300	200	-10	20

-200

15

-15

Table 2. Positions and velocities of four sensors.

When using the noise-free TOA and FOA measurements, the transient behavior of each neuron that represents the coordinates of source position and velocity after 50 independent runs is shown in Figure 2. It can be observed that after a short transient period from a random initial state, the neuron output state can eventually settle down at the true coordinates of source position and velocity. Besides, the output states can remain stable as the time elapses. The estimation results of using noise-free measurements exhibit that the proposed LPNN model is intrinsically to converge and remain stable.

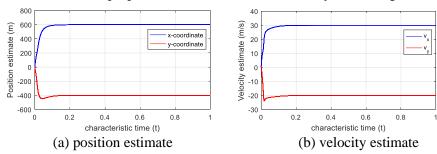


Figure 2. Transient behaviors of using noise-free measurements.

Then we examine the proposed network with noisy TOA and FOA measurements since we can only obtain the contaminated positioning parameters in the non-ideal scenarios. For the sake of simplicity, assume that the measured noise power of different sensors is the same, namely $\sigma_1^2 = \sigma_2^2 = ... = \sigma_M^2 = \sigma^2$ and $\dot{\sigma}_1^2 = \dot{\sigma}_2^2 = ... = \dot{\sigma}_M^2 = \dot{\sigma}^2$. Besides, we let $\dot{\sigma}^2 = 0.1\sigma^2$ as claimed in [2,3]. Here we set covariance σ^2 as 1 m^2 and 10 m^2 respectively to achieve two different measurement noise levels. The

DMCIT2019 IOP Publishing

IOP Conf. Series: Journal of Physics: Conf. Series 1284 (2019) 012064 doi:10.1088/1742-6596/1284/1/012064

transient behaviors of 50 independent runs for the neurons that of interest under two noise level are shown in Figure 3 and Figure 4. From the figures, we find that our neural network can converge and settle down after a short period of oscillation under both conditions. Moreover, it takes fewer time to converge when measurement noise is small, and the output estimates of source position and velocity are close to true values even when the measurement noise is large. In conclusion, our model equips the abilities of convergence and stability through the numerical experiment results.

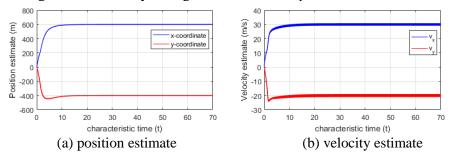


Figure 3. Transient behaviors when $\sigma^2 = 1 \text{ m}^2$.

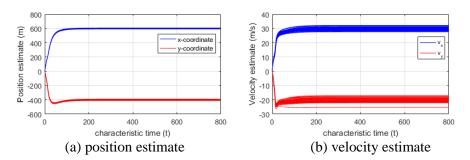


Figure 4. Transient behaviors when $\sigma^2 = 10 \text{ m}^2$.

4.2 Estimation performance

In this subsection, several simulations are conducted to evaluate the location and velocity estimation performance of the designed neural model. Meanwhile, the proposed method is compared with three classic numerical algorithms, namely the Taylor-series method (TS), weighted least square algorithm (WLS) and Monte Carlo importance sampling (MCIS) algorithm developed in [2]. Besides, the Cramér-Rao lower bound (CRLB) is also included as a benchmark.

We consider a 2-dimension scenario and four sensors are employed as listed in Table 1. The noise standard deviation σ is varied from 1 m to 17 m to achieve different noise level, and the localization performance is compared in terms of root mean square error (RMSE) through 1000 independent runs.

In the first test, the position of the moving source is (600, -50) m with velocity $\dot{u} = (30, -15)$ m/s. Figure 5 shows the RMSEs versus the noise standard deviation σ . It can be clearly seen that the method designed in this paper can reach the CRLB within a large noise variation range. However, the other three algorithms perform much worse than our method. As the measurement noise gradually grows large, the WLS algorithm suffers from the threshold effect, and thus its RMSE deviates from the CRLB immediately. The iterative Taylor series algorithm equips a higher threshold value than WLS, but the second-order term of noise ignored in this algorithm will cause a significant error when the measurement noise is large. Besides, the performance of the MCIS algorithm deteriorates seriously once the importance weight employed fail to modify the estimation in the importance sampling (IS) method at the high noise level.

In the second test, the true position and velocity of the moving source are randomly chosen from the 600×600 m and 60×60 square regions, respectively. Figure 6 shows the RMSEs versus the

DMCIT2019 IOP Publishing

IOP Conf. Series: Journal of Physics: Conf. Series 1284 (2019) 012064 doi:10.1088/1742-6596/1284/1/012064

measurement noise. We can directly observe that the designed algorithm still outperforms the other three algorithms and can obtain the CRLB even when σ is large.

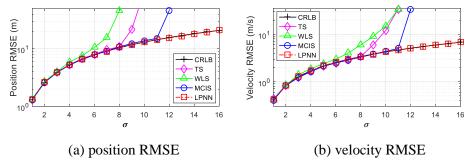


Figure 5. RMSEs versus σ with u = (600, -50) m and $\dot{u} = (30, -15)$ m/s.

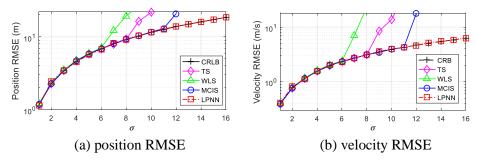


Figure 6. RMSEs versus σ when the source position and velocity are randomly generated.

In the third test, since the non-ideal sensor geometry will greatly affect the positioning performance, we place the four sensors in a line to test the robustness of the proposed neural network. Here the moving source position is (600, -50) m and $\dot{u} = (-30, 15)$ m/s, and the velocities of four sensors are the same as in Table 1, but the positions are changed to (300, 300) m, (100, 100) m, (-100, -100) m and (-300, -300) m. The RMSEs versus the measurement noise are shown in Figure 7 except for the RMSE of WLS algorithm. Because in this case, the WLS method has failed to estimate the source location and velocity, thus the estimation result of WLS is not given. Figure 7 exhibits the excellent performance of the proposed LPNN model even the other two types of algorithms deviate from CRLB consecutively under the non-ideal sensor localization geometry.

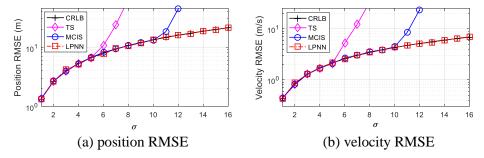


Figure 7. RMSEs versus σ when the four sensors are placed in a line.

5. Conclusions

In this paper, we build up a neural network for mobile source localization based on LPNN framework, which uses TOA and FOA measurements. Moreover, the proposed analog neural model is proved to be asymptotically stable, and the stability of the network is also verified through numerical experiments. The simulation results demonstrate that our LPNN model has excellent estimation performance

DMCIT2019 IOP Publishing

IOP Conf. Series: Journal of Physics: Conf. Series 1284 (2019) 012064 doi:10.1088/1742-6596/1284/1/012064

compared with the existing numerical algorithms and it can reach the CRLB even at the high noise level.

Acknowledgements

The authors acknowledge support from National Natural Science Foundation of China (Grant No. 61201381, No. 61401513 and No. 61772548), China Postdoctoral Science Foundation (Grant No. 2016M592989).

References

- [1] Romero L A and Mason J 2015 Geolocation using TOA, FOA and altitude information at singular geometries *IEEE Trans. Aerosp. Electron. Sys.* **51** 1069–1078
- [2] Liu R, Wang Y, Yin J, Wang D and Wu Y 2017 Passive source localization using importance sampling based on TOA and FOA measurements *Frontiers of Information Technology & Electronic Engineering* **18** 1167–1179
- [3] Foy W H 1976 Position-location solutions by Taylor-series estimation *IEEE Trans. Aerosp. Electron. Syst.* **12** 187–194
- [4] Ho K C, Lu X and Kovavisaruch L 2007 Source localization using TDOA and FDOA measurements in the presence of receiver location errors: Analysis and solution *IEEE Trans. Signal Process.* **55** 684–696
- [5] Wang Y, Huang J, Yang L and Xue Y 2015 TOA-based joint synchronization and source localization with random errors in sensor positions and sensor clock biases *Ad Hoc Networks* 27 99–111
- [6] Zou Y and Wan Q 2106 Asynchrous Time-of-arrival-based source localization with sensor position uncertainties *IEEE Commun. Lett.* **20** 1860–1863
- [7] Wang Y and Wu Y 2017 An efficient semidefinite relaxation algorithm for moving source localization using TDOA and FDOA measurements *IEEE Commun. Lett.* **21** 80–83
- [8] Zhang S and Constantinides A G 1992 Lagrange programming neural networks *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.* **39** 441–452
- [9] Bouzerdoum A and Pattison T R 1993 Neural network for quadratic optimization with bound constraints *IEEE Trans. Neural Netw.* **4** 293–304
- [10] Costantini G, Perfetti R and Todisco M 2008 Quasi-lagrangian neural network for convex quadratic optimization *IEEE Trans. Neural Netw.* **19** 1804–1809
- [11] Liang J, So H C, Leung C S, Li J and Farina A 2015 Waveform design with unit modulus and spectra l shape constraints via Lagrange programming neural network *IEEE J. Sel. Topics Signal Process.* **9** 1377–1386
- [12] Liang J, Leung C S and So H C 2016 Lagrange programming neural network approach for target localization in distributed MIMO radar *IEEE Trans. Signal Process.* **64** 1574–1585
- [13] Leung C S, Sum J, So H C, Constantinides A G and Chan F K W 2014 Lagrange programming neural network approach for time-of-arrival based source localization *Neural Comput. Appl.* **12** 109–116
- [14] Wang H, Feng R, Leung A C S and Tsang K F 2018 Lagrange programming neural network approaches for robust time-of-arrival localization *Cogn. Comput.* **10** 23–34
- [15] Jia C, Yin J, Wang D and Zhang L 2018 Lagrange programming neural network for TOA-based localization with clock asynchronization and sensor location uncertainties *Sensors* **18** 2293
- [16] Jia C, Wang D, Yin J, Chen X and Zhang L 2019 Joint multiple sources localization using TOA measurements based on Lagrange programming neural network *IEEE Access* **7** 3247–3263
- [17] Han Z, Leung C S, So H C and Constantinides A G 218 Augmented Lagrange programming neural network for localization using time-difference-of-arrival measurements *IEEE Trans. Neural. Netw. Learn. Syst.* **29** 3879–3884