TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 09/18 pp796-799
Volume 13, Number 6, December 2008

Anti-Jamming with Adaptive Arrays Utilizing Power Inversion Algorithm*

MENG Dawei (孟大伟), FENG Zhenming (冯振明), LU Mingquan (陆明泉)**

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

Abstract: The convergence rate of the power inversion (PI) algorithm is quite sensitive to the power of the interference with the used fixed parameters in the PI algorithm leading to degradation of its ability to handle interference. This paper presents a normalized PI algorithm that traces the stochastic characteristics of the interference. The algorithm adaptively adjusts the recursive step size to determine the constrained optimized parameters for the lowpass filter. Simulations show that the normalized PI algorithm achieves faster convergence and produces deeper nulls. The algorithm makes GPS receivers more robust in environments with large variations in the interference strength.

Key words: GPS; anti-jamming; antenna arrays; normalized power inversion

Introduction

GPS receivers need to operate efficiently at all times, even in the presence of interference and intentional jamming. In the last ten years, a growing number of studies have focused on antenna array methods, including programmable multibeam arrays, least mean square (LMS) error adaptive arrays, and applebaum arrays[1-6]. The strength of GPS signals reaching the receiver are often more than 30 dB below the thermal noise power. Adaptive arrays can reduce the effect of jamming signals so that the spread spectrum mechanism can extract the GPS signals[1]. The power inversion (PI) algorithm is similar to the LMS algorithm except that it does not require a reference signal nor does it require detailed information about the desired signal structure or desired signal arrival angle. Thus, the PI algorithm is ideal for anti-jamming processing of GPS signals. The PI algorithm seeks to minimize the array output power with the constraint that the antenna element weights do not go to zero. However, it cannot flexibly determine the lowpass filter parameters which impact the rate of convergence and the rejection capability. This paper presents an algorithm using Compton's approach to analyze GPS signals. The approach is normalized to improve the convergence speed of the algorithm.

1 Power Inversion Antenna Arrays

Compton[7] developed the PI algorithm in the late 1970's. The PI algorithm forms approximate nulls in the direction of the jamming based on the power ratio of the desired signal to the jammer, so that the strong signal is nulled in favor of the weaker signal if strong interference is detected. The antenna array in Fig. 1 can be formulated to perform power inversion using the processor shown in Fig. 2. This power inversion array is able to operate when neither the signal nor the arrival angle are known at the receiver. However, the power inversion array does not always provide as much signal improvement as adaptive array algorithms, which require additional information[8]. However, when the interference is strong, the power inversion array can still provide substantial improvement.

Received: 2007-04-23; revised: 2008-05-26

* Supported by the National High-Tech Research and Development (863) Program of China (No. 2006AA701108)

** To whom correspondence should be addressed.
E-mail: lumq@tsinghua.edu.cn; Tel: 86-10-62773632

Furthermore, even through the GPS signals generally have a low signal-to-thermal-noise ratio, the power inversion array does not null the GPS signals in the absence of interference. In the power inversion array, the number of interference signals to be cancelled is equal to the number of degrees of freedom where an M-element antenna array has $M-1$ degrees of freedom.

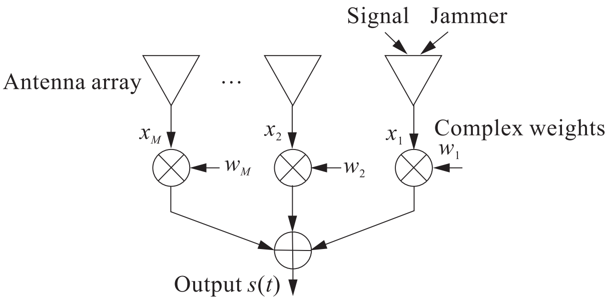


Fig. 1 *M*-element adaptive antenna array

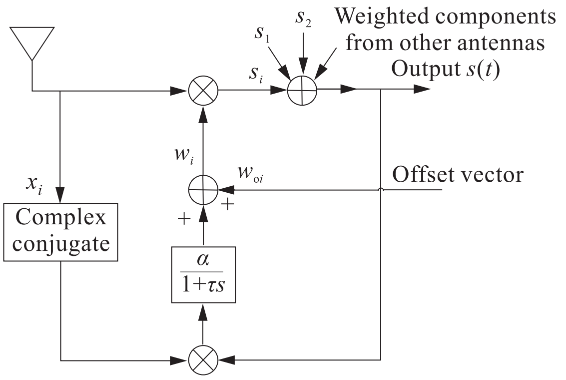


Fig. 2 Antenna array power inversion processor

The PI algorithm differs from the Applebaum array in that the integrator in the feedback loop is changed to a low-pass filter, with a simple offset vector, $\boldsymbol{w}_{\mathrm{o}}$ in Fig. 2, used as the weight vector. The power inversion array weights are designed to minimize the total output power under the constraint that the weights are not permitted to go to zero. The offset vector is chosen to provide the desired quiescent antenna pattern. For GPS signals, the desired antenna pattern is omni-directional,

$$\boldsymbol{w}_{\mathrm{o}} = [1,0,\ldots,0]^{\mathrm{T}} .$$

2 Normalized Power Inversion

The differential equation for the tap weights for the i-th element in Fig. 2 is as follows[7]:

$$\tau \frac{\mathrm{d}\, w_i}{\mathrm{d}\, t} + w_i = w_{\mathrm{o}i} - \alpha x_i^* s(t) \tag{1}$$

where α and τ are low pass filter (LPF) gain constants. Applying the backward difference $s = \dfrac{1-z^{-1}}{T}$ with $T=1$ to the control loop in Fig. 2 gives the update equation:

$$w_i'(k+1) = \frac{\tau}{1+\tau} w_i'(k) - \frac{\alpha}{1+\tau} x_i^*(k) s(k) \tag{2}$$

$$s(k) = \sum_{i=0}^{M-1} x_i(k) w_i(k) = \boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}(k) \tag{3}$$

where k is a time index, $\boldsymbol{x}$ is the array input signal vector, M is the number of antenna elements, and $\boldsymbol{w}$ is the adaptive weight vector which satisfies the following equation:

$$w_i(k) = w_i'(k) + w_{\mathrm{o}i} \tag{4}$$

In addition to the update equation, suitable values for the α and τ must be determined based on the speed of response requirements on the array and the allowable weight jitter. From Eq. (4),

$$w_i(k+1) - w_{\mathrm{o}i} = \frac{\tau}{1+\tau}(w_i(k) - w_{\mathrm{o}i}) - \frac{\alpha}{1+\tau} x_i^*(k) s(k) \tag{5}$$

For stability, assume that τ is sufficiently large[4] so $\dfrac{\tau}{1+\tau} \approx 1$. Then,

$$\delta w_i(k) = w_i(k+1) - w_i(k) = \frac{\alpha}{1+\tau} x_i^*(k) e(k) \tag{6}$$

The mean square error can be estimated by

$$\varepsilon^2(k) = \boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}(k)\boldsymbol{x}^{\mathrm{T}}(k)\boldsymbol{w}(k) \tag{7}$$

Let $\boldsymbol{w}(k+1)$ denote the updated weight vector at the (k+1)-th iteration. Then introduce the variable,

$$\begin{aligned}\tilde{\varepsilon}^2(k) = \boldsymbol{w}^{\mathrm{T}}(k+1)\boldsymbol{x}(k)\boldsymbol{x}^{\mathrm{T}}(k)\boldsymbol{w}(k+1) = \\ \varepsilon^2(k) + 2\delta\boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}(k)\boldsymbol{x}^{\mathrm{T}}(k)\boldsymbol{w}(k) + \\ \delta\boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}(k)\boldsymbol{x}^{\mathrm{T}}(k)\delta\boldsymbol{w}(k)\end{aligned} \tag{8}$$

The convergence speed can be improved by inducing the mean square error increment to as small as possible. The normalized PI algorithm can be derived using the principle of minimal disturbance[9].

$$\begin{aligned}\delta\varepsilon^2(k) = \tilde{\varepsilon}^2(k) - \varepsilon^2(k) = \\ 2\delta\boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}(k)\varepsilon(k) + \delta\boldsymbol{w}^{\mathrm{T}}(k)\boldsymbol{x}(k)\boldsymbol{x}^{\mathrm{T}}(k)\delta\boldsymbol{w}(k)\end{aligned} \tag{9}$$

Substituting Eq. (8) into Eq. (9) and expanding yields

$$\begin{aligned}\delta\varepsilon^2(k) = \frac{4\alpha}{1+\tau}\varepsilon^2(k)\boldsymbol{x}^*(k)\boldsymbol{x}(k) + \\ 4\left(\frac{\alpha}{1+\tau}\right)^2 \varepsilon^2(k)[\boldsymbol{x}^*(k)\boldsymbol{x}(k)]^2\end{aligned} \tag{10}$$

Taking a partial derivation of Eq. (10) with respect to $\alpha(k)$ gives

$$\frac{\partial\delta\varepsilon^2(k)}{\partial\alpha(k)} = 4\varepsilon^2(k)\boldsymbol{x}^{\mathrm{T}}(k)\boldsymbol{x}(k) + 8\left(\frac{\alpha}{1+\tau}\right)\varepsilon^2(k)[\boldsymbol{x}^{\mathrm{T}}(k)\boldsymbol{x}(k)]^2 \quad (11)$$

Setting Eq. (11) equal to 0 and solving for $\alpha(k)$ yields

$$\alpha(k) = -\frac{1}{2\boldsymbol{x}^{\mathrm{T}}(k)\boldsymbol{x}(k)} \quad (12)$$

One concern is that when the tap input vector is small numerical difficulties may arise because the square norm $\boldsymbol{x}^*(k)\boldsymbol{x}(k)$ includes division by a small number. To overcome this problem, Eq. (12) is modified as[10,11]

$$\alpha(k) = -\frac{1}{\gamma + 2\boldsymbol{x}^{\mathrm{T}}(k)\boldsymbol{x}(k)} \quad (13)$$

where $\gamma > 0$.

The stability of the normalized PI algorithm can be analyzed by rewriting the update equation as

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \frac{\alpha(k)}{1+\tau}\boldsymbol{x}^*(k)\boldsymbol{x}^{\mathrm{H}}(k)\boldsymbol{w}(k) = \boldsymbol{w}(k) - \frac{1}{2(1+\tau)\boldsymbol{x}^{\mathrm{T}}(k)\boldsymbol{x}(k)}\boldsymbol{x}^*(k)\boldsymbol{x}^{\mathrm{H}}(k)\boldsymbol{w}(k) \quad (14)$$

Taking the exception of both sides of Eq. (14) gives

$$E\{\boldsymbol{w}(k+1)\} = \left[I - \frac{1}{2(1+\tau)}\right]E\{\boldsymbol{w}(k)\} \quad (15)$$

Because $\tau > 0$, the matrix $\left[I - \frac{1}{2(1+\tau)}\right]$ is positive definite and its eigenvalues are less than 1; therefore, $\boldsymbol{w}(k)$ converges in the mean squares sense.

3 Numerical Simulations

Simulations were performed to evaluate the normalized PI algorithm using an 8-element uniform linear array with elements spaced a half wave length apart. All elements were assumed to have the same response characteristics. The desired signal is a baseband coarse/acquisition (C/A) code GPS signal. One sinusoidal jammer with a jammer power to signal power ratio, J/S, equal to 68 dB is located at $\theta = 38^\circ$. The thermal noise is 20 dB. All simulations were performed using two samples per C/A code chip. The simulations were performed using the update equations in Eq. (2) and Eq. (14) with $\tau = 200\,000$[6].

Simulation 1 shows that both algorithms can form an antenna pattern null at the jammer location. The null of the normalized PI algorithm is about 10 dB deeper than the null of PI algorithm. The antenna beam patterns are shown in Fig. 3.

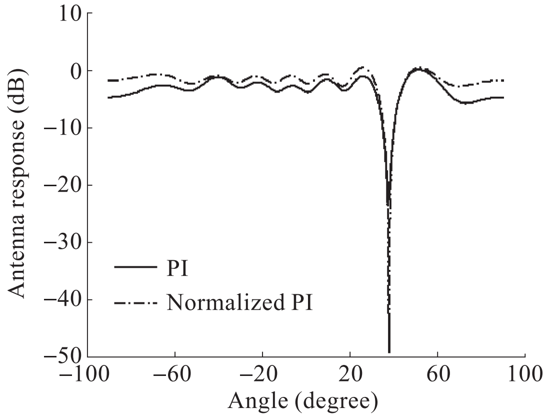
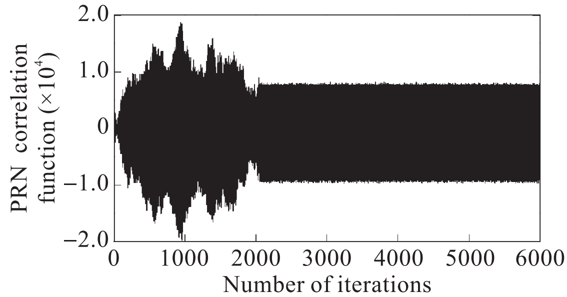
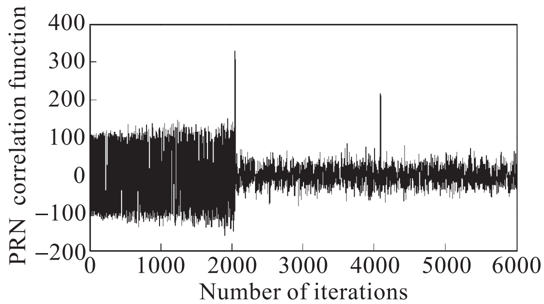


Fig. 3 Antenna beam patterns

The correlation function for the C/A code for the same scenario is shown in Fig. 4, where PRN represents the pseudo random noise. The quality of the correlation function is a function of the J/S ratios of the jammer and the spatial relationship of the jammer to the desired GPS signal. The correlation peak is quite clear with the null steering in the normalized PI algorithm, but almost cannot descend with the standard PI algorithm. Thus, the simulation indicates that the normalized PI algorithm more effectively filters the jammer to improve the signal-to-interference-and-noise ratio.



(a) Standard PI algorithm



(b) Normalized PI algorithm

Fig. 4 Spread spectrum signal correlation function

The convergence of the PI and normalized PI algorithm was also examined using the equations in Eq. (2) and Eq. (14) with $\tau = 200\,000$. The PI algorithm used $\alpha = 1$ and $\alpha = 100$. The results in Fig. 5 show that the weight vector of the normalized PI algorithm converged by about the 200 iteration, while the PI algorithm requires at least 1000 iterations to converge even with $\alpha = 100$. Thus, the normalized PI algorithm converges much faster.

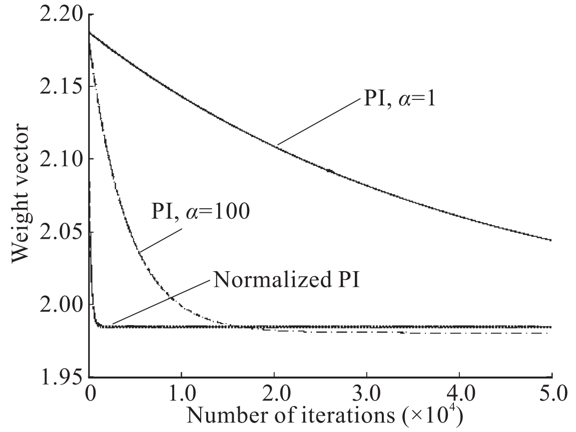


Fig. 5 PI and normalized PI learning curves

The effect of two continuous wave interference jammers located at 18° and 28° is shown in Fig. 6. The normalized PI algorithm accurately forms two mulls in the jamming directions. Thus, the new algorithm is able to counter multiple jammers.

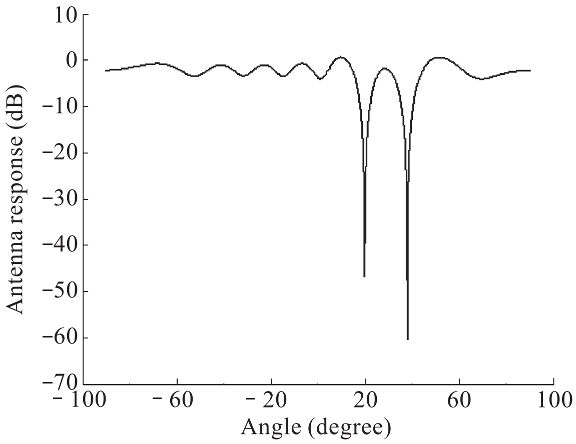


Fig. 6 Antenna beam patterns with two jammers

4 Conclusions

This paper presents a normalized PI algorithm for GPS receivers, which has excellent convergence. Since the interference environment can not be known a priori, the design of a power inversion antenna must choose α to be as small as any interference environment may require. Thus, the PI algorithm has very slow convergence when the interference power is strong. With the normalized PI algorithm, the array antenna adaptively adjusts the factor based on the input signal power to improve the convergence of the weight vector. Simulations show that the normalized PI algorithm has significantly faster convergence, which is significant in GPS applications.

References

[1] Spiker J J, Parkinson B W. Global Positioning System: Theory and Applications. Washington, DC, USA: American Institute of Aeronautics and Astronautics, 1996: 717-763.

[2] Gecan A, Zoltowski M. Power minimization techniques for GPS null steering antenna. In: ION GPS-95 Proceedings. Palm Springs, CA, USA, 1995: 861-868.

[3] Compton R T. Adaptive Antennas. Englewood Cliffs, NJ, USA: Prentice Hall, 1998.

[4] Applebaum A P. Adaptive arrays. *IEEE Trans. on Antennas and Propagation*, 1976, **24**(5): 585-598.

[5] Fante R L, Vaccaro J J. Wideband cancellation of interference in a GPS receiver array. *IEEE Trans. on Aerospace and Electronic Systems*, 2000, **36**(2): 549-564.

[6] Jade Y T, Liyeh L. Interference cancellation using power minimization and self-coherence properties of GPS signals. In: ION GNSS 17th International Technical Meeting of the Satellite Division. Long Beach, CA, USA: ION, 2004: 132-143.

[7] Compton R T. The power-inversion adaptive array: Concept and performance. *IEEE Trans. on Aerospace and Electronic Systems*, 1979, **15**(6): 803-814.

[8] Dirk T M Slock. On the convergence behavior of the LMS and the normalized LMS algorithms. *IEEE Trans. on Signal Processing*, 1993, **41**(9): 2811-2825.

[9] Manolakis D G. Statistical and Adaptive Signal Processing. Beijing: Tsinghua University Press, 2002.

[10] Haykin S. Adaptive Filter Theory, Fourth Edition. Beijing: Publishing House of Electronics Industry, 2002.

[11] Mohamed E A, Tan Zhangzhong. Adaptive antenna utilizing power inversion and linearly constrained minimum variances algorithm. *Chinese Journal of Aeronautics*, 2005, **18**(1): 153-160.