University of New Mexico
UNM Digital Repository

Electrical & Computer Engineering Faculty Publications

Engineering Publications

11-9-2003

Machine learning based CDMA power control

Chaouki T. Abdallah

Judd A. Rohwer

Christos G. Christodoulou

Follow this and additional works at: https://digitalrepository.unm.edu/ece_fsp

Machine Learning based CDMA Power Control

Judd A. Rohwer
Sandia National Laboratories
Telemetry Technology Development
P.O. Box 5800 MS-0986
Albuquerque, NM 87185-0986
Email: jarohwe@sandia.gov

Chaouki T. Abdallah and Christos G. Christodoulou
Department of Electrical and Computer Engineering
MSC01 1100
1 University of New Mexico
Albuquerque, NM 87131-0001
Email: chaouki,christos@eece.unm.edu

Abstract—This paper presents binary and multiclass machine learning techniques for CDMA power control. The power control commands are based on estimates of the signal and noise subspace eigenvalues and the signal subspace dimension. Results of two different sets of machine learning algorithms are presented. Binary machine learning algorithms generate fixed-step power control (FSPC) commands based on estimated eigenvalues and SIRs. A fixed-set of power control commands are generated with multiclass machine learning algorithms. The results show the limitations of a fixed-set power control system, but also show that a fixed-set system achieves comparable performance to high complexity closed-loop power control systems.

I. INTRODUCTION

Power control is a critical component in CDMA cellular systems. Power control combats the "near-far" effect by adjusting the transmit power of each mobile. This technique reduces the multiple access interference and if the system capacity is within the set limits, the desired signal-to-interference ratios (SIR) are achieved at all base stations.

Results of both binary and multiclass machine learning algorithms for CDMA power control are presented. The machine learning power control algorithms classify sets of signal and noise subspace eigenvalues, from the received signal sample covariance matrix, into SIR sets. The SIR based classes are related to fixed-sets of the CDMA power control commands.

The basic machine learning techniques include estimation of the signal subspace dimension, required for separating the signal and noise subspace eigenvalues. The advanced machine learning technique does not require the signal subspace dimension, only the complete set of eigenvalues. The machine learning training and testing methodologies differ for each, but the two techniques generate accurate CDMA power control commands.

II. MACHINE LEARNING TECHNIQUES

Machine learning research has largely been devoted to the binary and multiclass problems of data mining, text categorization, and pattern classification. Machine learning algorithms have already impacted analysis and design of communication

systems. Neural Networks have been applied to numerous problems, ranging from adaptive antenna arrays [1], multiuser receiver design [2], interference suppression [3], and power prediction [4]. New advanced learning techniques, such as support vector machines (SVM) have been applied, in the binary case, to receiver design and channel equalization [5]. Boosting algorithms [6] have been applied to standard classification problems, such as text and image classification, but have yet to be applied to specific communication problems.

Pattern classification algorithms apply classification rules to generate binary and multiclass labels to the input data. In the binary case, a classification function is estimated using input/output training pairs with unknown probability distribution, $P(\mathbf{x}, \mathbf{y})$, $\mathbf{x}$ is a vector of observations and $\mathbf{y}$ is the corresponding vector of machine learning labels.

Estimating the classification function is the process of minimizing the expected risk, defined as

$$R[f] = \int L\left(f\left(\mathbf{x}\right), \mathbf{y}\right) dP(\mathbf{x}, \mathbf{y}), \qquad (1)$$

where L is the loss function. For detailed information review the Vapnik-Chervonenkis (VC) theory and structural risk minimization (SRM) [7].

A. SVMs - Background

SVMs generate a classification function that separates data classes, with the largest margin, using a hyperplane. The data points near the optimal hyperplane are the "support vectors". SVMs are a nonparametric machine learning algorithm with the capability of controlling the capacity through the support vectors.

The general process of SVM algorithms is based on a projection of the input space to a higher dimensional feature space, F, via a nonlinear mapping,

$$\Gamma \quad : \quad \mathbb{R}^N \rightarrow F \qquad (2)$$

$$x \quad \mapsto \quad \Gamma\left(x\right). \qquad (3)$$

The input data $x_1, \ldots, x_N \in \mathbb{R}^N$ is mapped into a new feature space F which could have a much higher dimensionality. The data in the new feature space is then applied to the desired machine learning algorithm. Machine learning theory shows that the dimension of the feature space is not as important as the complexity of the classification functions. In the input

space separating the input/output pairs may require a nonlinear separating function, but in a higher dimension feature space the input/output pairs may be separated with a linear hyperplane.

1) Kernel Functions: Kernel functions are used to compute the scalar dot products of the input/output pairs in the feature space, F. This allows a decision rule to be applied to the inner product of training and test points in the feature space.

$$\Gamma(x) \cdot \Gamma(x_n) = k(x, x_n) \tag{4}$$

Four popular kernel functions are the linear kernel, polynomial kernel, radial basis function (RBF), and multilayer perceptrons (MLP). The performance of each kernel function varies with the characteristics of the input data. Refer to [8] for more information on feature spaces and kernel methods.

B. Binary Classification

In binary classification systems the machine learning algorithm generates the output labels with a hyperplane separation. The input sequence and a set of training labels are represented as $\{x_n, y_n\}_{n=1}^{N}$, $y_n = \{-1, +1\}$. If the two classes are linearly separable in the input space then the hyperplane is defined as $w^T x + b = 0$, w is a weight vector perpendicular to the separating hyperplane, b is a bias that shifts the hyperplane parallel to itself. If the input space is projected into a higher dimensional feature space then the hyperplane becomes $w^T \Gamma(x) + b = 0$.

C. Multiclass Classification

For the multiclass problem the machine learning algorithm produces estimates with multiple hyperplane separations. The set of input vectors and training labels is defined as $\{x_n, y_n^c\}_{n=1, c=1}^{n=N, c=C}$, $x_n \in \mathbb{R}^N$, $n = 1, \ldots, N$, $y_n \in \{1, \ldots, C\}$, n is the index of the training pattern and C is the number of classes.

There exist a number of approaches to the multiclass classification problem. Two primary techniques are one-vs-one and one-vs-rest. One-vs-one applies a binary machine learning algorithm to selected pairs of classes. For C distinct classes there are $\frac{C(C-1)}{2}$ hyperplanes that separate the classes. The one-vs-rest machine learning technique generates C hyperplanes that separate each distinct class from the ensemble of the rest. The Decision Directed Acyclic Graph (DDAG) and DAGSVM are specific techniques for one-vs-one multiclass classification.

D. DDAG and DAGSVM

Platt, et.al., [9] introduced the DDAG, a VC analysis of the margins, and the development of the DAGSVM algorithm. The two techniques are based on $\frac{C(C-1)}{2}$ classifiers for a C class problem. The DDAG algorithm includes $\frac{C(C-1)}{2}$ nodes, each associated with a binary classifier and it's respective hyperplane. The test error of the DDAG depends on the number of classes, C, and the margins between the data points and the hyperplanes, not on the dimension of the input space. In [9] it is proved that maximizing the margins at each node of the DDAG will minimize the generalization error, independent
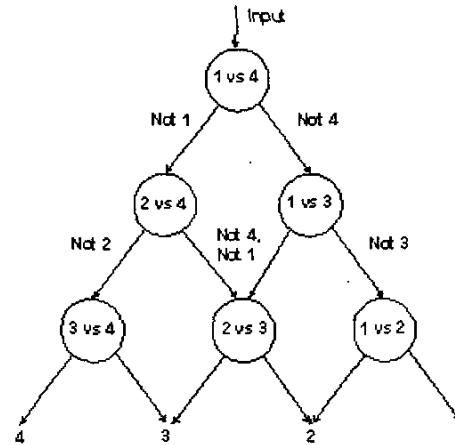

Fig. 1. DDAG for four classes

of the dimension of the input space. Refer to Figure 1 for a diagram of a four class DDAG with $\frac{4(4-1)}{2} = 6$ nodes.

The DAGSVM includes an efficient one-vs-one SVM implementation that allows for faster training than the standard one-vs-one algorithm and the one-vs-rest approach. The DAGSVM algorithm is based on the DDAG architecture with each node containing a binary SVM classifier of the ith and j^{th} classes. The training time of each DAGSVM node is equivalent to a binary SVM. The performance benefit of the DAGSVM is realized when the ith classifier is selected at the ith/j^{th} node and the jth class is eliminated. Thus any other class pairs containing the jth class are removed from the remaining SVM operations. The jth class is not a candidate for the output label. An analysis of the training times for one-vs-rest, one-vs-one and the DAGSVM are presented in [9]. In this paper one-vs-one multiclass classification is based on the binary Least Squares SVMs (LS-SVMs) [14].

III. FIXED-STEP POWER CONTROL BASED ON BINARY MACHINE LEARNING

The IS-95 and cdma2000/3G systems have an 800 bps up/down power control command rate; the single bit power control command is sent to the mobile station every 1.25 milliseconds [11]. This design limits the options with regards to power control systems, but the design constraints reduce the power control problem to generating the fixed-step power control command.

Binary classification for power control is based on a machine learning technique to produce a fixed-step command based on signal and noise subspace eigenvalues and the estimate of the signal subspace dimension. The optimal separating hyperplane in generated in the feature space, which separates the two power control classes. The label, $y_i \in [-1, 1]$, is

208

TABLE I
TRAINING AND TEST ERRORS OF BINARY MACHINE LEARNING
ALGORITHMS

	Training Error		Test Error	
	Class 0	Class 1	Class 0	Class 1
SVM	0.0012	0.0060	0.0016	0.0077
LS-SVM	0.0010	0.0000	0.0073	0.0000
AdaBoost	0.0000	0.0000	0.0032	0.0050
LocBoost	0.0071	0.0150	0.0151	0.0233

generated from the signal and interference power estimates,

$$
y_i = \begin{cases} 1 \text{ if } \frac{\hat{\phi}_s^2}{\hat{\phi}_{I+N}^2} < \gamma^* \\ -1 \text{ if } \frac{\hat{\phi}_s^2}{\hat{\phi}_{I+N}^2} \geq \gamma^* \end{cases} \quad \gamma^* \text{ is the desired } SIR. \quad (5)
$$

The signal power is defined as $\hat{\phi}_s^2$ and the interference and noise power is defined as $\hat{\phi}_{I+N}^2$. Refer to [10] for details on estimating the SIR given the signal and noise subspace eigenvalues and the signal subspace dimension. Refer to [12] for a LS-SVM based algorithm for estimating the signal subspace dimension.

Simulated results of four machine learning techniques for estimating the optimal hyperplane separation of a binary classification system are presented below. The four algorithms tested with simulated signal and noise power estimates are: 1) SVM with RBF kernel and perceptron solver [13], 2) LS-SVM [14], 3) AdaBoost,[6], and 4) LocBoost [15]. The training and test data consists of 4000 data points representing signal and interference power estimates. The four simulations include a 25% holdout processing; the system is trained with 1000 data points and binary machine learning labels are applied to 3000 test data points. The SVM, AdaBoost, and LocBoost simulations are based on MATLAB m files from the Classification Toolbox [16]. The LS-SVM simulation is based on MATLAB functions from the LS-SVMlab toolbox. Table I lists the training and test errors associated with each algorithm. The performance of the support vector machine algorithms equals that of the boosting algorithms; the LS-SVM technique has the best performance in terms of classification errors. Figure 2 displays the hyperplane separation of the LS-SVM machine learning algorithm. The vertical axis is the signal power estimate and the horizontal axis is the interference power estimate.

Simulation results of a binary FSPC system are shown in Figure 3. The top window includes the received SIR for 60 samples and the bottom window includes the FSPC commands. The transition between the two commands includes 20 sample points that must be rounded to the nearest power control label.

IV. FIXED-SET POWER CONTROL BASED ON MULTICLASS MACHINE LEARNING

A power control system with a fixed-set of power control commands is a compromise between the fixed-step power
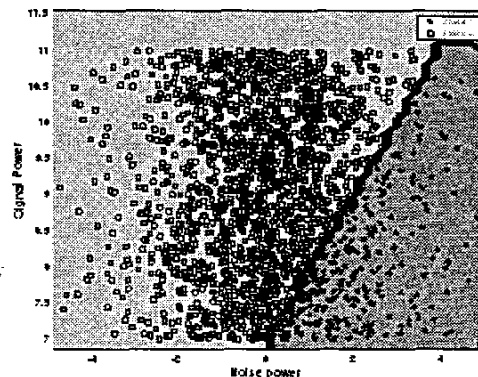

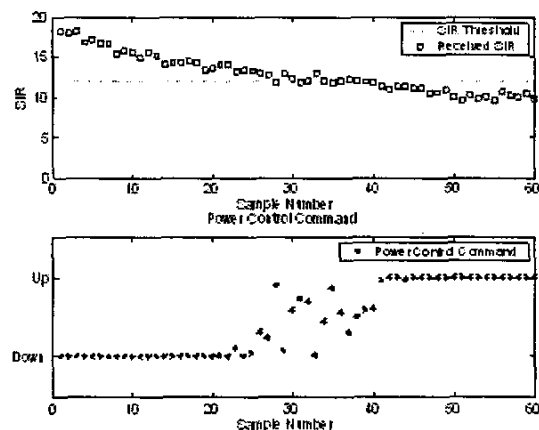Fig. 2. LS-SVM and hyperplane separation of SIR estimates.

Fig. 3. Simulation results of the binary LS-SVM power control system.

control command and a continuous power control system, such as the state-space linear quadratic power control (LQPC) [17] or the constrained second-order power control algorithm (CSOPC) [18]. This machine learning approach to power control is a system that relies only on training data and the receiver outputs. A basic SIR based power control system would not require a state-space power control design, the SIR estimate would be compared to the desired SIR and then the appropriate power control command, from a fixed-set, would be sent to the mobile station.

The fixed-set of power control commands is generated with a multiclass machine learning system. The multiclass system is based on the binary label system, $y_i \in \mathbb{k}$, where $\mathbb{k}$ is a set of real numbers that represent an appropriate range of expected SIR values, for example $\mathbb{k} \in [3, 5, 7, 9, 11]$. Each class represents a range of received SIRs, which is translated into a power control command. The size of the power control command, PC, is directly related to the size of the one-vs-one multiclass DDAG structure.

$$PC = (\gamma^* - y_i), \quad y_i \in \mathbf{k} \qquad (6)$$

Refer to [12] for details of the LS-SVM algorithm for CDMA power control based on eigenvalue estimates.

The IS-95 and cdma2000/3G systems have an 800 bps up/down power control command rate. The power control systems are limited to a single power control bit sent to the mobile station every 1.25 milliseconds [11]. This constraint forces the cellular system to FSPC. The fixed-set power control system requires additional power control bits. Two power control bits are required for the three and four class fixed-set designs. The five class system could be implemented with three power control bits. These bit requirements could be achieved with the use of auxiliary channels defined in [19].

A. Simulation Results

The training and testing vectors for the LS-SVM DDAG algorithm for fixed-set power control are based on simulated CDMA data. The training vectors include uniformly distributed noise powers with a set of mean values equal to one of the five DDAG classes. The simulated data includes a range of transmit powers, time varying channel conditions, an eight element antenna array, and eigenvalue estimates generated with the PASTd subspace tracking algorithm [20].

1) DDAG Training and LS-SVM Testing: The DDAG training is primarily affected by the span of the signal and interference powers in the training vectors. The data in the training vectors must span the entire region around the SIR thresholds, but the data should not overflow into adjacent SIR sets. Figure 4 displays the training data plotted with the set of SIR thresholds. The noise power spans the region of $8dB$ to $24dB$. Likewise the signal power spans the region of $10dB$ to $35dB$. The training data must cover all signal and noise powers that could be detected at the receiver.
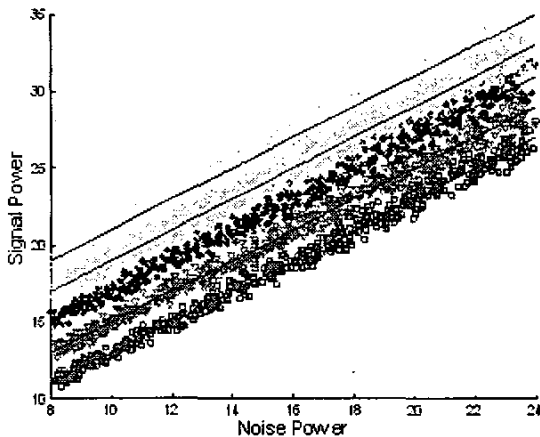

Fig. 4. SIR data points plotted as signal power versus noise power. The SIR power control thresholds are displayed for reference.

Figure 5 presents a graphic representation of the received SIR, the machine learning estimates, and the SIR thresholds.

As shown in Figure 5 the received SIRs range from $3.5dB$ to $11dB$, with a large majority of SIRs around the $5dB$ and $7dB$ classes.
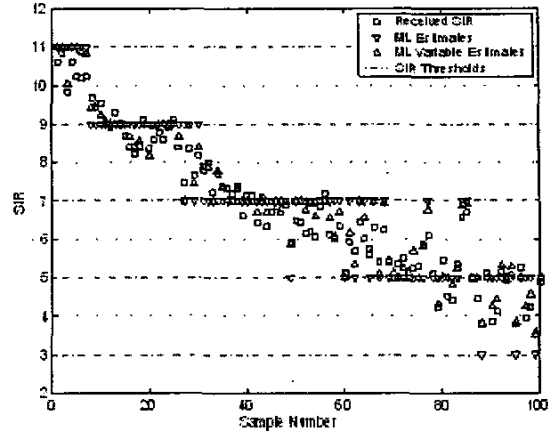

Fig. 5. Received SIR data points plotted as a function of samples. The plot includes the ML estimates and SIR thresholds.

V. SIMULATIONS OF POWER CONTROL ALGORITHMS

Two standard methods for characterizing the performance capabilities of power control algorithms is the convergence rate and the mobile capacity [17]. A performance indicator for both methods is the probability of outage, the probability that the mobile's received SIR is below the desired threshold. The rate of convergence is defined as the number of power control iterations required for the system's probability of outage to converge to a steady state value. The capacity of a mobile cellular system is the number of mobile users that can be supported while achieving the required Quality of Service (QoS).

The simulation system includes randomly generated link gains for the number of mobiles simultaneously entering the system, $P_{i,\max} = 1W$, $\gamma^* = 6dB$, bit rate $= 9.6KHz$, $BW = 1.2288MHz$, and $n_i = 10^{-12}$. Refer to [17] for a complete overview of the simulation environment.

Figure 6 is a comparison of the LQPC and the CSOPC algorithms. The data in the top window shows that LQPC supports 20 mobiles with zero outages while CSOPC supports 17 mobiles. Simulated data in the bottom window show that the LQPC requires three iterations before it generates the optimal power assignments for eighteen mobiles entering a stable system.

The top window of Figure 7 includes the probability of outage versus the number of mobiles in the cellular system for FSPC and three fixed-set power control systems. For fixed-set power control the capacity increases with the size of the power control set. The five class fixed-set system supports twenty mobiles with zero probability of outage. The bottom window of Figure 7 plots the probability of outage versus the number of iterations. The FSPC system converges
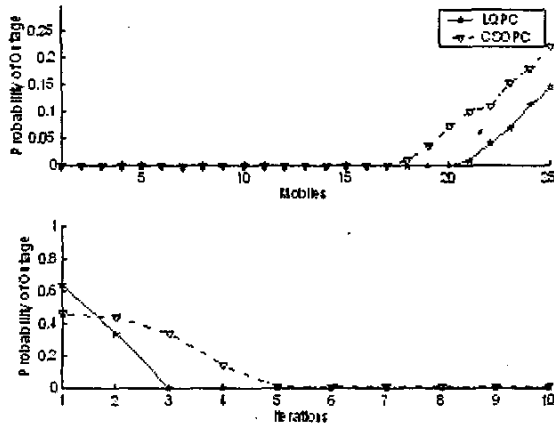
Fig. 6. Comparison of LQPC and CSOPC, in terms of probability of outage versus mobile capacity and iterations.
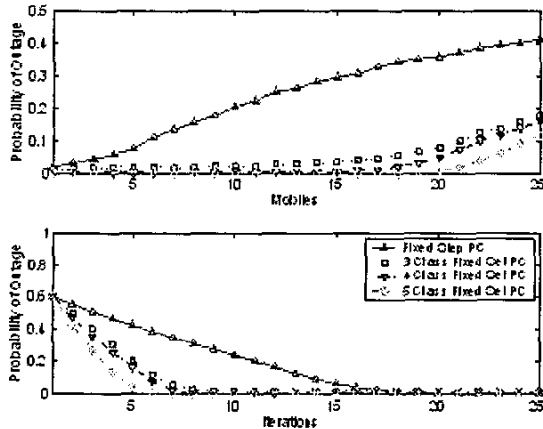

Fig. 7. Comparison of probability of outage versus mobile capacity and iterations for FSPC, and three, four, and five class fixed-set power control.

after eighteen power control iterations. The three, four, and five class fixed-set systems converge after eleven, eight, and seven iterations, respectively. The mobile capacity for each system may increase with a greater number of power control iterations, but as the number of iterations increases so does the required computational time.

VI. CONCLUSION

In this paper we present two system solutions for machine learning based power control. Both binary and multiclass machine learning algorithms are developed to solve this power control classification problem. Knowledge of the received SIR, signal subspace dimension, or BER/FER are not required for an accurate and fast power control system. The machine learning power control algorithms classify the set of eigenvalues into the received SIR set, which then determines the power control command.

REFERENCES

[1] Ahmed H. El Zooghby, Christos G. Christodoulou, and Michael Georgiopoulos, "A Neural Network-Based Smart Antenna For Multiple Source Tracking", IEEE Transactions On Antennas and Propagation, vol. 48, no. 5, pp. 768-776, May 2000.

[2] David C. Chen and Bing J. Sheu, "A Compact Neural-Network-Based CDMA Receiver", IEEE Transactions On Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, no. 3, pp. 384-387, March 1998.

[3] Kaushik Das and Salvatore D. Morgera, "Adaptive Interference Cancellation for DS-CDMA Systems Using Neural Network Techniques", IEEE Journal On Selected Areas In Communications, vol. 16, no. 9, pp.1774-1784, December 1998.

[4] Xiao Ming Gao, Xiao Zhi Gao, Jarno M. A. Tanskanen, and Seppo J. Ovaska, "Power Prediction in Mobile Communication Systems Using an Optimal Neural-Network Structure", IEEE Transactions On Neural Networks, vol. 8, no., 6, pp. 1446-1455, November 1997.

[5] Daniel J. Sebald and James A. Bucklew, "Support Vector Machine Techniques for Nonlinear Equalization", IEEE Transactions On Signal Processing, vol. 48, no. 11, pp. 3217-3226, November 2000.

[6] Michael Collins, Robert E. Schapire, and Yoram Singer, "Logistic Regression, AdaBoost and Bregman Distances", Machine Learning, vol. 48, no.1/2/3, pp. 253-285, 2002.

[7] Vladimir N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.

[8] Nello Cristianini and John Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, New York, 2000.

[9] John C. Platt, Nello Christianini, and John Shawe-Taylor, "Large Margin DAGs for Multiclass Classification", in Advances in Neural Information Processing Systems, vol. 12, pp. 547-553, Cambridge, MA, MIT Press, 2000.

[10] Deppa Ramakrishna, Naraya B. Mandayam, and Roy D. Yates, "Subspace-Based SIR Estimation for CDMA Cellular Systems", IEEE Transactions On Vehicular Technology, vol. 49, no. 5, pp. 1732-1742, May 2000

[11] Joseph C. Liberti, Jr. and Theodore S. Rappaport, Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications, Prentice Hall, Upper Saddle River, NJ, 1999.

[12] Judd A. Rohwer, "Machine Learning Methods for CDMA Power Control and Direction of Arrival Estimation", Ph.D. Dissertation, University of New Mexico, Albuquerque, New Mexico, May 2003.

[13] Johan A.K. Suykens, "Support Vector Machines: A Nonlinear Modelling and Control Perspective", European Journal of Control, vol. 7, pp. 311-327, 2001.

[14] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, Least Squares Support Vector Machines World Scientific, New Jersey, 2002..

[15] Ron Meir, Ran El-Yaniv, and Shai Ben-David, "Localized Boosting", Proceedings of the 13th Annual Conference on Computational Learning Theory, pp. 190-199, 2000.

[16] David G. Stork and Elad Yom-Tov, "User's Guide: Classification Toolbox", http://tiger.technion.ac.il/~eladyt/classification/index.htm

[17] Aly El-Osery and Chaouki Abdallah, "Distributed Power Control in CDMA Cellular Systems", IEEE Antennas and Propagation Magazine, Vol. 42, no. 4, pp. 152-159, August 2000.

[18] Riku Jantti and Seong-Lyun Kim, "Second-Order Power Control with Asymptotically Fast Convergence", IEEE Journal On Selected Areas In Communications, vol. 18, no. 3, pp.447-457, March 2000.

[19] Vijay K. Garg, IS-95 CDMA and cdma2000, Prentice Hall, Upper Saddle River, NJ, 2000.

[20] Bin Yang, "Projection Approximation Subspace Tracking", IEEE Transactions on Signal Processing, vol. 43, no. 1, pp. 95-107, January 1995.