A Tool for the Analysis of Market Power Potential in Electric Energy Markets

by

Alex Borden

A thesis submitted in fulfillment of the

requirements for the degree of

Master of Science

(Electrical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2010

Abstract

Market power is the ability of a market participant to alter the market price of a good or service without losing customers to competition. The participant's price is profitably maintained above a competitive level for a significant period of time. In ideal competitive markets, one would need to lower prices to garner more market share. In uncompetitive markets, participants who know their product is absolutely needed or not substitutable, can profitably raise prices to high levels [i]. In electric energy markets, there are many contributing factors determining competitiveness including market share, market concentration, elasticity of demand, excess capacity, contractual arrangements, price establishment processes and ease of market entry [ii].

The market participants in electric energy markets are the generators supplying electricity. Here the suppliers possessing market power potential can increase revenue without affecting the revenue and electric dispatch of other generators. When a centrally-dispatched electricity market is competitive, economic operation of the grid is transparent and efficient. However when a supplier possess an advantage over others, the benefits of a competitive electricity market may be jeopardized [ix].

Market power problems are complicated in electric markets due to specific properties of electricity and transmission grids [iii]. High concentrations of intensive power use can constrain transmission systems, and the ability of some loads to be served by generators located at remote distances would be limited. The areas where insufficient transmission capacity cannot reliably supply 100% of the electric load, without relying on generation physically located nearby, are termed "load pockets". Substitutability for the commodity is not readily available in these load pockets. Other sources of inefficiency are limited generation capacity of suppliers and a lack of sufficient energy storage. Generators that can capitalize on transmission deficiencies can

increase price while maintaining their amount of energy sold. The constraints on the transmission grid allow one or more suppliers to exploit locational advantages to their benefit at the expense of the end user.

There exist many tools widely used for identifying market power potential. Most rely on direct application of concentration measures, such as Residual Supplier Indices and the Hirschmann-Herfindahl Index, which show a single supplier or small number of suppliers dominating the market. These concentration measures are valuable but often fail to capture the true degree of competition [ix]. Concentration measures do not consider price-responsiveness or elasticity of demand, and a detailed representation of the electric grid may not be included. In electric energy markets various network operating limits often present market participants an uncompetitive advantage in small segments of the network, referred to as “local market power.” Concentration measures may fail to identify these market participants exhibiting local geographic advantage when assessing the network as an aggregate [iv].

Distribution factors related to sensitivity of constrained lines and supplier dispatch are used in a PJM three pivotal supplier test that is said to be more accurate than concentration measures [v]. Here each constraint is examined individually to determine whether three suppliers are jointly pivotal in relieving the constraint. A similar approach will be taken in the algorithm of this paper, but will differ in that the simultaneous combined effect of all line constraints will be considered instead of evaluating one at a time. This new work is significant because the presence of multiple line constraints could further provoke market power potential, and these locational advantages may be less evident in the analysis of a lone constraint.

The goal of this paper is to outline an algorithmic routine that exploits sensitivity analyses of coupled economic and physical models. Load demand is assumed to be inelastic, and only the effect of offer price on the market is studied. Limits are placed on transmission lines known to be easily congested, and a linear programming optimal flow is run. Matrices of dispatch/offer price and revenue/offer price sensitivities are then calculated for further evaluation which will highlight suppliers able to manipulate the market to their advantage.

This analysis does not necessarily conclude certain generators are exercising market power or to what extent this potential may be exercised. It serves as a screening tool so that dominant suppliers flagged can be investigated for the possibility of market power potential. The impact of this work will improve the efficiency and reliability of the electric power grid.

Acknowledgements

This work in this paper would not have been made possible without the generous support of the Consortium for Electric Reliability Technology Solutions (CERTS). I personally found much purpose and meaning in this research. It has helped me comprehend the nature of generation markets and the pursuit of fairness between suppliers, competitors and customers. I also give thanks to the Power System Engineering Research Center (PSERC). Being involved with PSERC and hearing the seminars has helped me understand necessary areas of research and future challenges. I am extremely grateful for the invaluable experiences provided by CERTS and PSERC through which young engineers are given the opportunity to make the most of themselves.

I would also like to thank my advisor Professor Bernard Lesieutre. His vast understanding and knowledge in the area of Power Systems has been a crucial resource in my education as a Master's student. His accomplishments are inspiring, and his cheerful attitude is also a quality to be admired. My sincerest appreciation is given for the rewarding experience he has presented myself with under his supervision.

The faculty professors of the UW college of engineering are not to go without mention. I am very thankful to be under the tutelage of such esteemed and intelligent individuals. I feel fortunate to learn under their guidance.

Last but not least I would like to express my gratitude to the WiSPERC and WEMPEC students. They are truly some of the brightest young students I've ever met and are another valuable source of support and friendship.

Table of Contents

List of Tables

List of Figures

1. Introduction

The goal of this paper is to use models capturing market and network behavior to establish an efficient and accurate algorithm for identifying small numbers of suppliers with the ability to change prices without affecting dispatch. The algorithm should be robust and quick enough to be used in real-time market scenarios. To begin, some market economics being dealt with will be examined. An experiment involving market power exploitation of load pockets will also be presented.

1.1 Market Economics

In electric energy markets, increasing revenues may not imply increasing profits. Consider a basic example. When offer prices $\left[\frac{\$}{MW}\right]$ are lowered, an increased market share $[MW]$ should be witnessed as lower bids for energy are purchased first in market auctions. Lowering price to increase market share could increase revenue [$], but the cost to generate more electricity$\left[\frac{\$}{MW}\right]$ will also increase.

$$profit[\$] = revenue[\$] - cost\ of\ generation\left[\frac{MW}{\$}\right] \times\ dispatch[\$]$$

The difference between profit and revenue has been acknowledged, but for the remainder of this work revenue will be used as a proxy for profit. Information of increased cost of generation and other expenses synonymous with increasing dispatch are not needed when studying market power. Aside from knowledge of transmission grid topology, only data of offer prices resulting in zero change in dispatch is needed, hence the study of revenue and not profit is used.

1.2 Experiment Market Studies

If a small set of suppliers can simultaneously raise prices and increase revenues, they have some amount of market power. Consider a six supplier example extracted from a Real Time Market Power Monitoring PSERC presentation by Dr. Bernard Lesieutre [vi] [vii]:

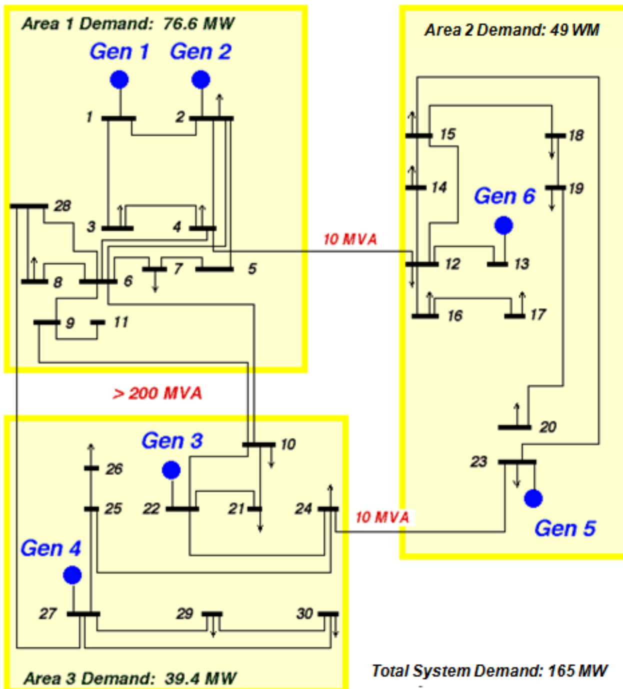


Figure 1: Six Supplier Network

The total system load is 165 MW and a capacity of 60 MW is placed on each generator. The lines connecting the left and right network are each limited to a max flow of 10MVA and inhibit power flow. The constraints produce a load pocket in the right network where generators 5 and 6 are. The above can be simplified to the following diagram:

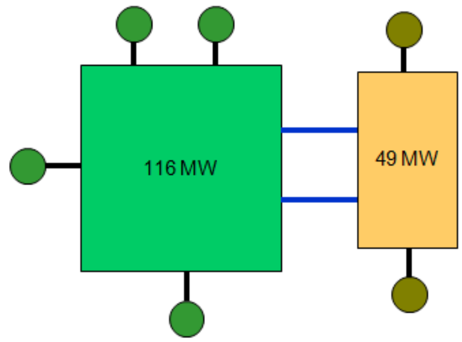


Figure 2: Six Supplier Network Simplified

The right load demands 49 MWs and can draw a maximum of 20 MWs from the left network. Therefore the load on the right must purchase at least 29 MWs from generators 5 and 6. Generators 5 and 6 can offer market prices higher than offered by Generators 1-4 due to locational advantages, shown in the following Figure 3: Example Base Case Offers. The three blocks of energy represent each generator's 60 MW capacity and would be sold in a uniform price auction.

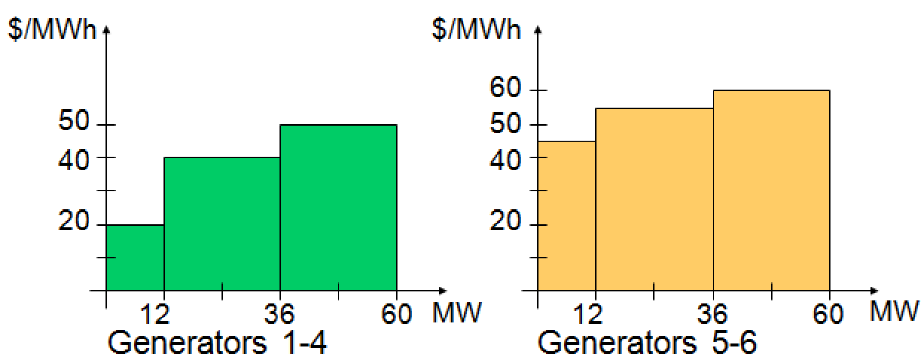


Figure 3: Example Base Case Offers

A base case solution for the full nonlinear AC optimal power flow is shown in Table 1.

	G1	G2	G3	G4	G5	G6
Dispatch (MW)	31.7	36.0	34.0	36.0	17.6	12.0
Price ($/MWh)	40.0	40.1	40.0	40.1	55.0	54.3

Table 1: Base Case Dispatch and Nodal Prices

The matrix of revenue/offer price sensitivities for this base case solution is shown in Table 2.

$$\Delta r = change\ in\ revenue$$

$$\Delta y = change\ in\ offer\ price$$

$$\Delta g = change\ in\ dispatch$$

$$\Delta \lambda = nodal\ price$$

Each entry of Table 2 is defined by

$$\frac{dr_i}{dy_j} = \lambda_i^* \frac{dg_i}{dy_j} + g_i^* \frac{d\lambda_i}{dy_j}$$

around the operating point (g^*, λ^*) representing the proportion in revenue change for the i^{th} generator (row i in the matrix) due to a change in nodal price at the j^{th} generator (column j of the matrix).

$$\begin{bmatrix} \Delta r_1 \\ \Delta r_2 \\ \Delta r_3 \\ \Delta r_4 \\ \Delta r_4 \end{bmatrix} = \begin{bmatrix} -3298 & 3231 & 31 & 65 & 52 & -49 \\ 3219 & -3695 & 244 & 263 & 315 & -310 \\ 31 & 244 & -544 & 308 & -234 & 229 \\ 65 & 263 & 307 & -597 & -127 & 125 \\ 38 & 230 & -170 & -93 & -160 & 173 \\ -36 & -229 & 169 & 92 & 175 & -159 \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta y_4 \\ \Delta y_4 \end{bmatrix}$$

Table 2: Six Supplier Revenue/Price Matrix

The negative entries on the diagonal indicate no generator acting alone can simultaneously increase offer price and revenue. Also, the sum of each row is positive, revealing if all suppliers equally raise prices, all would experience increased revenue. Closer examination reveals a load pocket for generators 5 and 6.

$$\begin{bmatrix} \Delta r_1 \\ \Delta r_2 \\ \Delta r_3 \\ \Delta r_4 \\ \Delta r_4 \end{bmatrix} = \left[\begin{array}{cccc|cc} -3298 & 3231 & 31 & 65 & 52 & -49 \\ 3219 & -3695 & 244 & 263 & 315 & -310 \\ 31 & 244 & -544 & 308 & -234 & 229 \\ 65 & 263 & 307 & -597 & -127 & 125 \\ \hline 38 & 230 & -170 & -93 & -160 & 173 \\ -36 & -229 & 169 & 92 & 175 & -159 \end{array}\right] \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta y_4 \\ \Delta y_4 \end{bmatrix}$$

Table 3: Six Supplier Load Pocket Identified

If generators 5 and 6 raise prices simultaneously their own revenues increase. They are the only pair of generators with this ability. Furthermore, if the two generators proportionally raise their prices near unity, revenue of the remaining four generators remains unchanged. This suggests generators 5 and 6 share potential market power.

An experiment was conducted at Cornell University with students acting as the six generators. Each generator attempted to maximize revenue without communication or collusion between participants. A seventy-five round uniform price auction was performed with each student specifying a price for each of the three fixed blocks of energy representing the capacity of their generator. An $80/MWh price offer cap was imposed, and results suggest that had the experiment continued past seventy-five rounds, the load pocket would have approached the price offer cap.

	G1	G2	G3	G4	G5	G6
Dispatch (MW)	37.9	34.9	30.1	34.9	14.9	14.6
Price ($/MWh)	48.5	48.7	48.5	48.6	**72.0**	**70.0**

Table 4: Experimental Dispatch and Nodal Prices Results

Generators 5 and 6 in the load pocket were able to exploit their joint market power to their mutual advantage. It is typical in economic experiments for suppliers with joint market power potential to discover this ability without direct collusion.

2. Sensitivity Analysis

Traditional concentration measures fail to account for some engineering constraints [ix], so the approach of this work relies on sophisticated procedures developing sensitivity data of economic and physical models. Analysis is simplified by assuming quadratic offer function for the suppliers, and the change in dispatch to the change in the cost function's quadratic coefficient term is calculated. Qualitatively the change in dispatch to changes in cost is preserved which allows simple DC power flow analysis with limits imposed. If a supplier can successfully raise prices without changing dispatch, their profits will necessarily increase [viii]. This lack of competitiveness clearly indicates the potential for market power. Revenue, offer price and dispatch sensitivities will be observed in order to monitor locational advantages and market power.

2.1 Dispatch/Offer Price Observations

A simple DC power flow analysis is employed to model injected active power in a lossless network, and a dispatch/offer price ($\Delta g/\Delta y$) sensitivity matrix is generated to gain insight into some network spectral properties.

$$m_{ij} = \frac{\partial g_i}{\partial y_j}$$

$$\Delta g = M\Delta y$$

If m is the number of generators in the network, than M is an $m \times m$ matrix, and Δg and Δy are $m \times 1$ vectors. The matrix M reflects certain substitutability properties of the system. Its diagonal entries are typically negative, so one supplier cannot unilaterally increase both supply and price as competitors can substitute supply [ix]. More importantly M is singular, meaning a price perturbation vector Δy exists creating zero change in dispatch $\Delta g = 0$.

$$0 = M\Delta y \tag{1}$$

Here Δy is in the null space of M, $Nul(M)$. A price perturbation vector Δy that will always create zero change in dispatch is a column of all ones representing equal incremental cost.

This signifies the ability of all suppliers equally and simultaneously raising prices and thus not affecting dispatch. This vector of all equal elements arises when no line limits are active and losses are neglected as in a DC power flow representation.

A less obvious property of M is that the dimension of its null space increases with the number of transmission constraints. As line flow limits become active the null space grows in dimension. In general the basis of $Nul(M)$ in an electric network with $(n-1)$ line constraints will have n linearly independent vectors [x].

$$Nul(M) = B = [\Delta y_1 \; \Delta y_2 \dots \Delta y_n]$$

Any linear combination of $\Delta y_1, \Delta y_2, \dots, \Delta y_n$ will also satisfy (1), and hence there are an infinite number of price perturbation vectors in the null space of M [xi].

2.2 Dispatch/Offer Price Utilization

A nineteen generator bus example of the dispatch/offer price null space matrix B is shown below in Table 5. The rows represent generation buses, and each independent column represents a scenario of offer price perturbations about market clearing price that results in no change of electric dispatch. In this example a DC optimal power flow, with assumed quadratic offer function for the suppliers, was run on an IEEE sample 118-bus system [xii]. Once the binding constraints are known the $m \; x \; n$ matrix B can be easily computed from a linear programming tableau. Three line constraints were imposed in the example of Table 5. Hence there are four columns or four scenarios of price perturbation that do not change dispatch. Three of the columns are from the three line constraints and the fourth for the scenario of all generators equally incrementing prices.

$$B = \begin{bmatrix} -0.9915 & 0.3210 & 0.3388 & 1.0000 \\ 0.0171 & -0.4538 & 0.3332 & 1.0000 \\ 0.0171 & 0.0626 & 0.2686 & 1.0000 \\ 0.0171 & 0.0777 & 0.2916 & 1.0000 \\ 0.0171 & 0.0207 & 0.2862 & 1.0000 \\ 0.0171 & 0.0576 & -0.1247 & 1.0000 \\ 0.0171 & 0.0586 & -0.1587 & 1.0000 \\ 0.0171 & 0.0596 & -0.1960 & 1.0000 \\ 0.0171 & 0.0604 & -0.2280 & 1.0000 \\ 0.0171 & 0.0608 & -0.2404 & 1.0000 \\ 0.0171 & 0.0615 & -0.2666 & 1.0000 \\ 0.0171 & 0.0603 & -0.2221 & 1.0000 \\ 0.0171 & 0.00592 & -0.2009 & 1.0000 \\ 0.0171 & 0.0598 & -0.2183 & 1.0000 \\ 0.0171 & 0.0596 & -0.2140 & 1.0000 \\ 0.0171 & 0.0597 & -0.2148 & 1.0000 \\ 0.0171 & 0.0597 & -0.2160 & 1.0000 \\ 0.0171 & 0.0597 & -0.2160 & 1.0000 \\ 0.0171 & 0.0597 & -0.2160 & 1.0000 \end{bmatrix}$$

$$= [B_1 \quad B_2 \quad B_3 \quad B_4] \quad (19\, rows\ x\ 4\, columns)$$

Table 5: Dispatch/Offer Price Sensitivity Matrix

The goal is to find a linear combination of these four columns that will generate a new price perturbation vector. Ideally for the purpose of identifying market power, a price perturbation vector having few large entries and many zero or near zero entries is desired. This ideal scenario would highlight a few market participants raising prices, with the other generator prices remaining unchanged and all dispatch levels maintained constant. The few dominant generators highlighted would appear to have the potential ability to increase prices, maintain dispatch and print money with no negative feedback.

$$[B_1 \quad B_2 \quad B_3 \quad B_4] * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$= B_1 * x_1 + B_2 * x_2 + B_3 * x_3 + B_4 * x_4$$

$$= \Delta y = \begin{bmatrix} few\ large\ entries \\ - \\ many\ zero\ entries \end{bmatrix} \quad (19\ rows\ x\ 1\ column)$$

3. Generating Price Perturbation Vectors

The possible combination values of x (x_1, x_2, x_3 and x_4 for the $19\ x\ 4$ example of Table 5) are infinite, thus infinite possible linear combinations of B and infinite price perturbation vectors Δy. Somewhere in these infinite combinations, a vector x that generates a quality price perturbation vector Δy is desired.

3.1 Moore-Penrose Pseudo Inverse

If Δy is already known or can be approximated, then the Moore-Penrose pseudo-inverse can be used to find the least squares x values [xiii].

$$Bx = \Delta y$$

$$x_{optimal} = B^{\dagger} \Delta y_{initial\ \ condition}$$

$$Bx_{optimal} = \Delta y^{*}$$

$$minimizes\{\|\Delta y^{*} - \Delta y_{initial\ \ condition}\|_2\}$$

The vector $x_{optimal}$ minimizes the Euclidean two-norm. The problem here is $\Delta y_{initial\ \ condition}$ needs to be specified or approximated before solving. Recall the basis B referred to in Table 5. It appears that a linear combination of columns 1 and 4 could generate a new vector close to $[1\ 0\ 0\ \ldots 0]^T$.

$$x_{optimal} = B^{\dagger}[1\ 0\ 0\ \ldots 0]^T = \begin{bmatrix} -0.9915 \\ -0.0000 \\ 0.0000 \\ 0.0169 \end{bmatrix}$$

$$= \begin{bmatrix} 1.0000 \\ -0.0000 \\ -0.0000 \\ -0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ -0.0000 \\ -0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ -0.0000 \\ -0.0000 \\ 0.0000 \\ 0.0000 \\ -0.0000 \end{bmatrix}$$

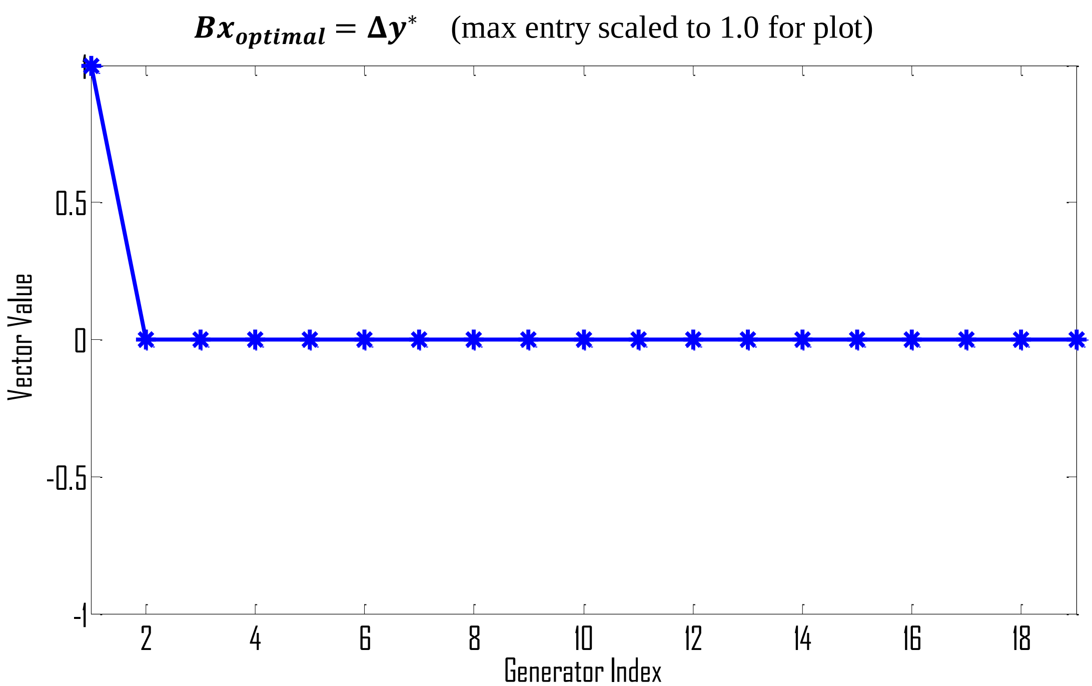


Figure 4: Moore-Penrose Success

$$||\Delta y^* - \Delta y_{initial\ \ condition}\ ||_2 = 3.4203 \times 10^{-15}$$

The new price perturbation vector Δy^* is basically equivalent to the initial condition unit vector $\Delta y_{initial\ \ condition}$ and is an ideal price perturbation vector desired to be identified. The above demonstrates the quickness and ease of successfully using the Moore-Penrose pseudo inverse.

Now consider a less suitable example however. Take the initial condition to be another unit vector with 1.0 placed at the 11th row.

$$x_{optimal}\ \ = B^{\dagger}[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T = \begin{bmatrix} -0.0720 \\ -0.0892 \\ -0.2344 \\ 0.0366 \end{bmatrix}$$

$$= \begin{bmatrix} -0.0000 \\ -0.0240 \\ -0.3584 \\ -0.4513 \\ -0.3628 \\ 0.6440 \\ 0.7290 \\ 0.8229 \\ 0.9031 \\ 0.9342 \\ 1.0000 \\ 0.8883 \\ 0.8356 \\ 0.8791 \\ 0.8685 \\ 0.8704 \\ 0.8734 \\ 0.8734 \\ 0.8734 \end{bmatrix}$$

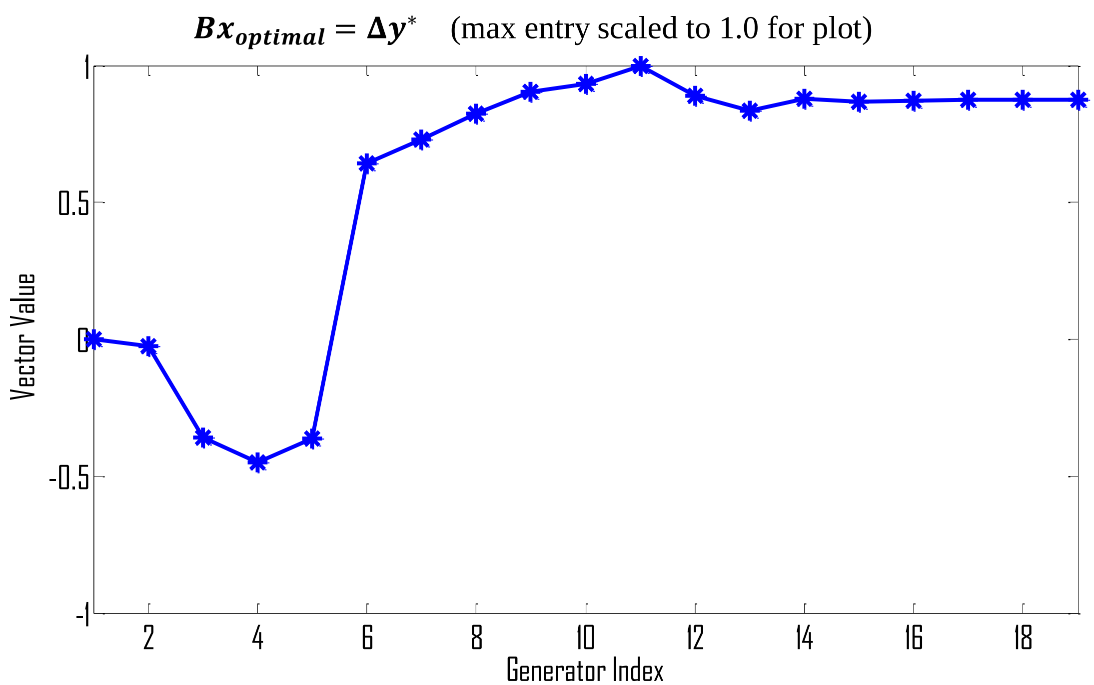


Figure 5: Moore-Penrose Failure

$$\|\Delta y^* - \Delta y_{initial\ \ condition}\|_2 = 0.9527$$

This new price perturbation vector does not resemble $\Delta y_{initial\ \ condition}$ whatsoever as can be seen with a larger two-norm of 0.9527. More importantly though, the vector can be refined. How the following refined vector was generated is not important for now, just the fact that its achievable is what's of interest.

$$
= \begin{bmatrix} -0.0000 \\ -0.0000 \\ 0.0433 \\ -0.0000 \\ 0.0175 \\ 0.7470 \\ 0.8075 \\ 0.8742 \\ 0.9311 \\ 0.9352 \\ 1.0000 \\ 0.9207 \\ 0.8829 \\ 0.9139 \\ 0.9063 \\ 0.9077 \\ 0.9099 \\ 0.9099 \\ 0.9099 \end{bmatrix}
$$

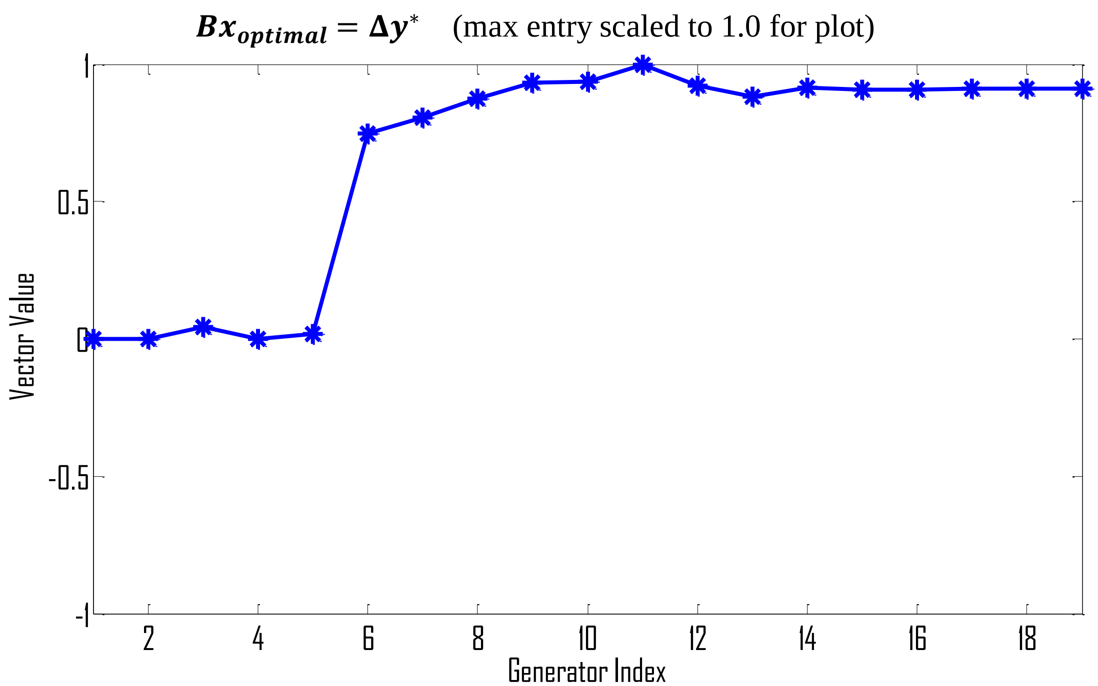


Figure 6: Refined Vector

This price perturbation vector is better than that shown in Figure 5. To generate this preferred vector for results, the perfect initial condition $\Delta y_{initial\ condition}$ would be needed. An infinite number of initial condition vectors $\Delta y_{initial\ condition}$ are available, making it difficult to find the right one. The Moore-Penrose pseudo-inverse is a quick and easy refinement tool for exposing market power potential if a good initial guess at a price perturbation vector is known. Typically there will be better available approaches however.

3.2 Eigen-analysis

Recall that market power is the ability of one or more market participants to raise the price of their service without losing customers to competition. Of course all market participants could simultaneously raise prices, and customers would have no better option of provider than those of whom they original purchased from. This scenario is unlikely however as in practice there are far too many suppliers who would need to jointly act in overt coordination. Therefore it is of primary concern to identify one or small groups of generators that can raise prices and profit while the many other generator prices remain unchanged or very slightly changed. This can be shown to be an eigenvalue analysis problem.

$$Bx = \begin{bmatrix} B_+ \\ - \\ B_- \end{bmatrix} x = \begin{bmatrix} B_+x \\ - \\ B_-x \end{bmatrix} \begin{matrix} \leftarrow large\ entries, few\ rows \\ \\ \leftarrow small\ entries, many\ rows \end{matrix}$$
$$(m\ rows\ x\ 1\ column)$$

$$\lambda_{max} = \max\{x^T(B_+^T B_+ - B_-^T B_-)x\} \quad (2)$$

Preferably the number of generators in group B_+ is few $\left(\#rows(B_+) \ll \#rows(B_-)\right)$ implying a small group of generators can exercise market power if the many generators making up group B_- do not alter prices. The proof of the above equation (2) is simple. It is desired the absolute value entries of $|B_+x| \gg |B_-x|$. Taking $(B_+x)^T(B_+x)$ yields the scalar sum of the squares of B_+x, and the same can be done for B_-x. Therefore the eigenvector x that maximizes the difference between $(B_+x)^T(B_+x)$ and $(B_-x)^T(B_-x)$ is needed in the search for price perturbation vectors having few large entries and many small entries.

$$\max\{(B_+x)^T(B_+x) - (B_-x)^T(B_-x)\}$$
$$= \max\{x^T B_+^T B_+ x - x^T B_-^T B_- x\}$$
$$= \max\{x^T(B_+^T B_+ - B_-^T B_-)x\}$$
$$= x_{max}^T (B_+^T B_+ - B_-^T B_-)x_{max}$$

Equation (2) equals the maximum eigenvalue of the square matrix $B_+^T B_+ - B_-^T B_-$. Consider the standard eigenvalue equation.

$$Ax = \lambda x$$

$$\rightarrow x^T A x = \lambda$$

In this case A equals $B_+^T B_+ - B_-^T B_-$ and choosing its maximum eigenvalue and corresponding eigenvector maximally optimizes the separation between the large entries in $B_+ x$ and small entries in $B_- x$. Once the optimal eigenvector is identified, a price perturbation vector in the column space of B can be generated.

$$Bx_{optimal} = \Delta y_{new}$$

The last refinement made to this new vector Δy_{new} is that it is scaled so its maximum absolute value row equals one.

3.3 Orthonormality

It is best if the given basis $B = \begin{bmatrix} B_+ \\ - \\ B_- \end{bmatrix}$ is orthonormal for eigen-analysis. An orthornormal basis can be thought of as a rotation or a unitary transformation to a new coordinate system. The basis B dealt with has dimensions $m \times n$ with $m > n$. Thus the orthonormal basis B has orthornormal columns, but not rows. The two-norm of each column in B must equal 1.0, and $B^T B = I$ the identity matrix. An orthonormal basis provides the best rigid-body transformation to a simpler coordinate system. It can be computed using singular value decomposition (SVD). Here the matrix B can be expressed as $B = U\Sigma V^*$ with Σ being a square matrix having $B's$ singular values on its diagonal. If r is the rank of B, the first r left singular vectors or columns of $U\{u_1, \ldots, u_r\}$ generate an orthonormal basis of B [xi].

Consider performing one computation of eigen-analysis on the non-orthonormal basis B of Table 5: Dispatch/Offer Price Sensitivity Matrix.

$$B_+ = [-0.9915 \quad 0.3210 \quad 0.3388 \quad 1] \qquad B_- = \begin{bmatrix} 0.0171 & -0.4538 & 0.3332 & 1 \\ 0.0171 & 0.0626 & 0.2686 & 1 \\ 0.0171 & 0.0777 & 0.2916 & 1 \\ 0.0171 & 0.0207 & 0.2862 & 1 \\ 0.0171 & 0.0576 & -0.1247 & 1 \\ 0.0171 & 0.0586 & -0.1587 & 1 \\ 0.0171 & 0.0596 & -0.1960 & 1 \\ 0.0171 & 0.0604 & -0.2280 & 1 \\ 0.0171 & 0.0608 & -0.2404 & 1 \\ 0.0171 & 0.0615 & -0.2666 & 1 \\ 0.0171 & 0.0603 & -0.2221 & 1 \\ 0.0171 & 0.00592 & -0.2009 & 1 \\ 0.0171 & 0.0598 & -0.2183 & 1 \\ 0.0171 & 0.0596 & -0.2140 & 1 \\ 0.0171 & 0.0597 & -0.2148 & 1 \\ 0.0171 & 0.0597 & -0.2160 & 1 \\ 0.0171 & 0.0597 & -0.2160 & 1 \\ 0.0171 & 0.0597 & -0.2160 & 1 \end{bmatrix}$$

$$eig(B_+^T B_+ - B_-^T B_-) \rightarrow x_{\max\_non\_orthonormal}$$

Now take basis B of Table 5: Dispatch/Offer Price Sensitivity Matrix to be orthonormal, $B = orth(B)$, and again perform eigen-analysis with B_+ similarly being the first row and B_- the remaining rows.

$$eig(B_+^T B_+ - B_-^T B_-) \rightarrow x_{\max\_orthonormal}$$

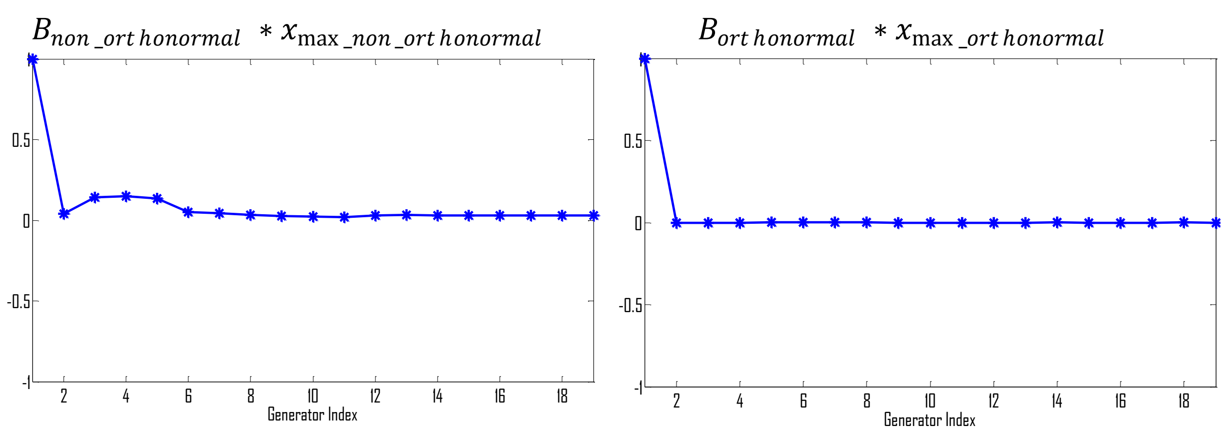


Figure 7: Non-Orthonormal vs. Orthonormal

It can be seen that the orthonormal basis produces a sharper, more unit-like vector. Upon orthonormal rotation of a basis, unit vectors or least-squared vectors which are desired in market power analysis are successfully isolated.

Consider the IEEE 118-bus case [xii], with nineteen generators considered online. Constraints were placed on the three lines shown in bold red in the below Figure 8.

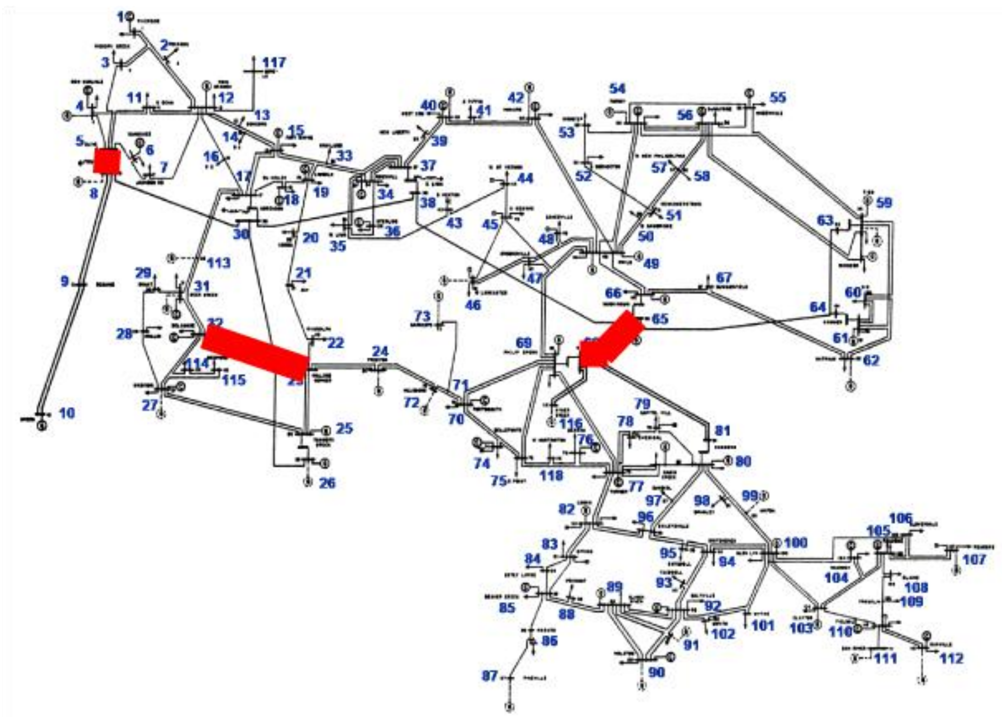


Figure 8: 118-Bus Test System w/ Line Congestion

The following is the orthonormal basis calculated from the DC optimal power flow:

$$B_{orth} = B = \begin{bmatrix} 0.2438 & 0.1080 & 0.5039 & -0.3428 \\ 0.2455 & 0.1195 & -0.8490 & -0.0640 \\ 0.2377 & 0.0599 & 0.0415 & 0.2302 \\ 0.2398 & 0.0764 & 0.0671 & 0.0907 \\ 0.2427 & 0.0903 & -0.0804 & -0.8494 \\ 0.2430 & 0.1033 & 0.0314 & 0.0732 \\ 0.2466 & 0.1345 & 0.0359 & 0.0899 \\ 0.2507 & 0.1696 & 0.0404 & 0.1067 \\ 0.2544 & 0.2013 & 0.0445 & 0.1220 \\ 0.2562 & 0.2166 & 0.0464 & 0.1293 \\ 0.2602 & 0.2517 & 0.0509 & 0.1462 \\ 0.2532 & 0.1908 & 0.0431 & 0.1169 \\ 0.1984 & -0.2774 & 0.0068 & 0.0460 \\ 0.1925 & -0.3284 & 0.0030 & 0.0144 \\ 0.1931 & -0.3229 & 0.0035 & 0.0215 \\ 0.1930 & -0.3240 & 0.0034 & 0.0201 \\ 0.1928 & -0.3254 & 0.0033 & 0.0183 \\ 0.1928 & -0.3254 & 0.0033 & 0.0183 \\ 0.1928 & -0.3254 & 0.0033 & 0.0183 \end{bmatrix}$$

Table 6: Orthonormal Basis

Table 6 will now be referred to as example basis B, for an orthonormal basis is needed when using eigen-analysis.

3.4 Market Power Monitoring Tool Challenges

There are a few significant challenges encountered when developing a practical algorithm for identifying market power potential.

The first challenge is ensuring a reasonable execution time of the algorithm. For instance when using the Moore-Penrose pseudo inverse to generate price perturbation vectors, one must specify what initial condition vectors $\Delta y_{initial\ condition}$ to try. The number of potential vector candidates is infinite. The more initial conditions tried, the more resulting price perturbation vectors generated and odds increase of finding a meaningful vector. When using eigen-analysis, one must first select which rows to be placed in B_+ with the remaining rows in B_-. This approach at least has a limited amount of combinations. If there are m rows in B than there are 2^m combinations of B_+/B_- $\left(\binom{0}{m} + \binom{1}{m} + \binom{2}{m} + \cdots + \binom{m}{m} = 2^m\right)$. Though this number of computations is limited, it doubles with each additional row or generator, and can become unreasonably large very quickly. It will be shown later that clustering methods can be used for speed. A negligible or slight accuracy sacrifice is made in exchange for vast improvements in execution time.

A second challenge encountered when developing the algorithm is avoiding redundant price perturbation results. Typically the resulting vectors aren't exactly identical but very slightly different as they are all linear combinations of one another. Vectors that are a slight perturbation of another are considered redundant and should not be outputted. Only a small core set of these vectors is desired, so a filtering mechanism must be established to weed out similar or redundant results.

A third challenge encountered is determining what is considered to be a successful new price perturbation vector $Bx = \Delta y$. The use of vector norms is suggested as the best measure of quality and will be discussed further on.

4. Clustering

As considered earlier, one challenge encountered when developing an efficient algorithm for determining market power is computational cost and execution time. Eigen-analysis will be used to calculate new vectors highlighting market power. If the computational power and time was available, the eigenvalue equation (2) could be executed for every row combination of B_+/B_- to calculate all generator collusions and observe all resulting price perturbation vectors of interest. Fundamentally this approach is combinatorial, and with the nineteen generator basis of Table 6, there would be 524,288 computations ($2^{19} = 524{,}288$). This number of computations usually takes more than an hour depending on the computer being used. In practice there will be even more than 19 generators to be considered, thus some method of generalization needs to be resorted to in order to avoid unreasonable computation time.

It is typical in power systems for generator dispatch/offer price sensitivities to respond similarly to neighboring generators due to grid infrastructure and distance to line congestions. Therefore generators exhibiting similar behavior can be clustered together allowing for fewer computations of eigen-analysis. Combinatorial brute force will be used to check many combinations of B_{+}/B_{-}, but instead of switching one row at a time in and out of B_{+}/B_{-}, an entire cluster of similar generators can be switched. For example, the nineteen row basis B of Table 6 will be clustered into 5 clusters. After clustering, the eigenvalue equation (2) will be executed 2^5 times for all combinations of the five clusters. Here $2^5 = 32 \ll 2^{19} = 524{,}288$. This lower number of computations is now reasonable, and by clustering similar generators no significant results highlighting market power are lost.

4.1 Quality Threshold Clustering

The Quality Threshold (QT) algorithm is a clustering method where the number of resulting clusters is not specified, but the maximum diameter of entries in a cluster is specified *a priori*. The algorithm ensures a quality guarantee in that all entries of a particular cluster do not exceed the maximum diameter threshold. Any measure of distance can be used for measurement, but only the Euclidean distance will be evaluated for now.

The QT algorithm generates candidate clusters by encapsulating all points within the specified maximum diameter. The candidate cluster encapsulating the most points is selected as the first permanent cluster. All points from that cluster are removed from availability in the next iteration, and the process repeats itself until all points belong to a cluster [xiv]. The QT algorithm gives the same results for a given basis every time it is performed.

4.2 K-Means Clustering

The K-means clustering algorithm requires less computational power than the Quality Threshold algorithm. It requires a desired number of clusters k to be specified *a priori*, as opposed to a specified distance threshold being specified.

The k random points are selected heuristically and become the initial cluster centroids. Each point in the basis is then assigned to its closest centroid. Next the centroids are reformulated by taking the mean location of the points in the centroid. Each point in the basis is again assigned to the nearest new centroid, and the centroid means are recalculated. This process continues until the points stop changing clusters, and the total sum of distance to centroid mean is minimized.

An issue to be mindful of with the K-means clustering algorithm is that the random initial centroids affect the outcome of the final centroids. Stated another way, different final clusters can be expected in multiple runs of the K-means algorithm as local minimums rather than global minimums are converged to. To get around this issue the sum of the distances of each point to the mean of their centroid is taken. The K-means algorithm is replicated multiple times, and the run with the least sum of the distances is taken to be the best represented set of clusters [xv].

It is less than desirable to require the user of this market power algorithm to have to specify any input parameters such as number of clusters. It is possible a type of measure could be developed that identifies the best number of clusters k to eliminate the need for user input. The K-means algorithm will be chosen as the primary clustering tool for the algorithm. Like the QT algorithm, any measure of distance can be used, but for now only the Euclidean distance will be chosen for use. Applying the K-means clustering algorithm to the nineteen row orthonormal basis B of Table 6 specifying five clusters gives the following results:

$$\begin{array}{cc}
Clusters & \\
\begin{array}{c} 2 \\ 5 \\ 4 \\ 4 \\ 1 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \end{array} &
\begin{bmatrix}
0.2438 & 0.1080 & 0.5039 & -0.3428 \\
0.2455 & 0.1195 & -0.8490 & -0.0640 \\
0.2377 & 0.0599 & 0.0415 & 0.2302 \\
0.2398 & 0.0764 & 0.0671 & 0.0907 \\
0.2427 & 0.0903 & -0.0804 & -0.8494 \\
0.2430 & 0.1033 & 0.0314 & 0.0732 \\
0.2466 & 0.1345 & 0.0359 & 0.0899 \\
0.2507 & 0.1696 & 0.0404 & 0.1067 \\
0.2544 & 0.2013 & 0.0445 & 0.1220 \\
0.2562 & 0.2166 & 0.0464 & 0.1293 \\
0.2602 & 0.2517 & 0.0509 & 0.1462 \\
0.2532 & 0.1908 & 0.0431 & 0.1169 \\
0.1984 & -0.2774 & 0.0068 & 0.0460 \\
0.1925 & -0.3284 & 0.0030 & 0.0144 \\
0.1931 & -0.3229 & 0.0035 & 0.0215 \\
0.1930 & -0.3240 & 0.0034 & 0.0201 \\
0.1928 & -0.3254 & 0.0033 & 0.0183 \\
0.1928 & -0.3254 & 0.0033 & 0.0183 \\
0.1928 & -0.3254 & 0.0033 & 0.0183
\end{bmatrix}
\end{array}$$

Table 7: Orthonormal Basis Clustered

5. Filtering Redundant Results

The second challenge mentioned for developing the market power algorithm, was that an approach was needed for filtering numerous result vectors down to a smaller easily observable selection. If considering the nineteen row basis B of Table 6 with five clusters, $2^5 = 32$ new price perturbation vectors will be generated. Of these 32 vectors, only four would be needed to span the column space of basis B having rank four. At the bare minimum no fewer than the rank number of linear independent price perturbation vectors will be computed for output, however more vectors are usually preferred when examining for market power.

Clustering the price perturbation vectors is an effective way to eliminate redundant results. This second round of clustering is not to be confused with the first round of clustering described in Section 4. Clustering. Row clustering was done then to prevent from executing the eigenvalue equation (2) too many times thus saving time. This second round of clustering is performed on the resulting columns or price perturbation vectors to remove redundancy. There are many options available for performing the column clustering which will be further explored.

5.1 K-Means Revisited

One method of clustering the price perturbation vectors could be done by revisiting the K-means algorithm. Each of the k clusters specified would correspond to a possible price increment scenario by the market participants. As usual with the K-means clustering algorithm the number of desired clusters k must be specified *a priori*, and as suggested earlier a type of measure could be developed to identify the best number of output clusters k to avoid a second user input.

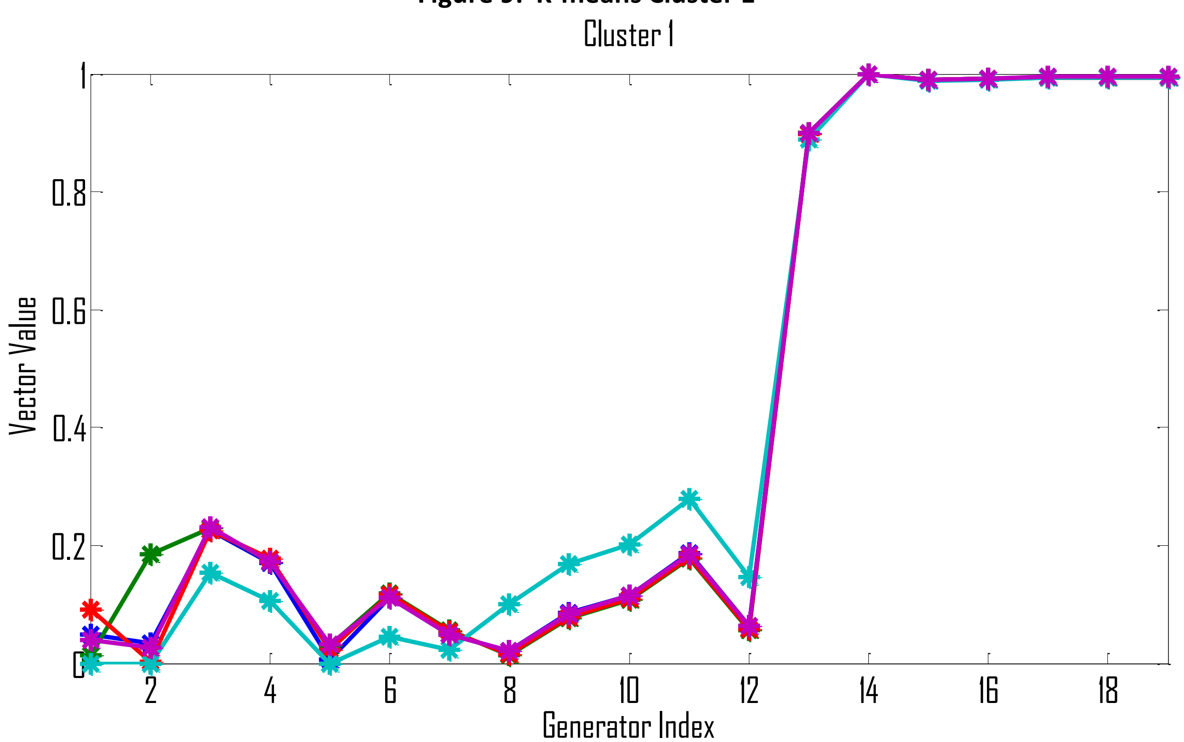
5.2 K-Means Column Clustering Example

Using five row clusters yields 32 resulting price perturbation vectors from the orthonormal basis of Table 6. It can be observed that many of the vectors are similar to each other, and there are more results than cared to be analyzed. The vectors hence needed to be clustered. In one particular run of the K-means algorithm, specifying six clusters for output yielded the following results. The plots are of the absolute valued vectors. Eventually in Section 6.4 Output Price Perturbation Vectors, the best price perturbation vector from each of these six clusters is chosen. Below each of the following figures is the linked figures of price perturbation vectors eventually chosen for output and vice versa.

Cluster 1				
0.0497	-0.0134	0.0907	0	0.0391
0.0338	0.1852	0.0029	0	0.0274
0.2258	0.2285	0.2262	0.154	0.2303
0.1707	0.1716	0.1771	0.1061	0.1714
-0.0061	0.0305	0.0236	0	-0.0326
0.111	0.1186	0.1166	0.0449	0.1109
0.0488	0.0564	0.0545	-0.0231	0.049
-0.0215	-0.0137	-0.0156	-0.0997	-0.0209
-0.0851	-0.0773	-0.0791	-0.1691	-0.0842
-0.1158	-0.1079	-0.1097	-0.2025	-0.1147
-0.1861	-0.1781	-0.1798	-0.2791	-0.1847
-0.064	-0.0562	-0.058	-0.146	-0.0632
0.8988	0.8993	0.8987	0.8892	0.8995
1	1	1	1	1
0.9897	0.9897	0.9896	0.9884	0.9899
0.9918	0.9918	0.9917	0.9907	0.9919
0.9944	0.9944	0.9943	0.9937	0.9945
0.9944	0.9944	0.9943	0.9937	0.9945
0.9944	0.9944	0.9943	0.9937	0.9945

Table 8: K-means Cluster 1

Figure 9: K-means Cluster 1

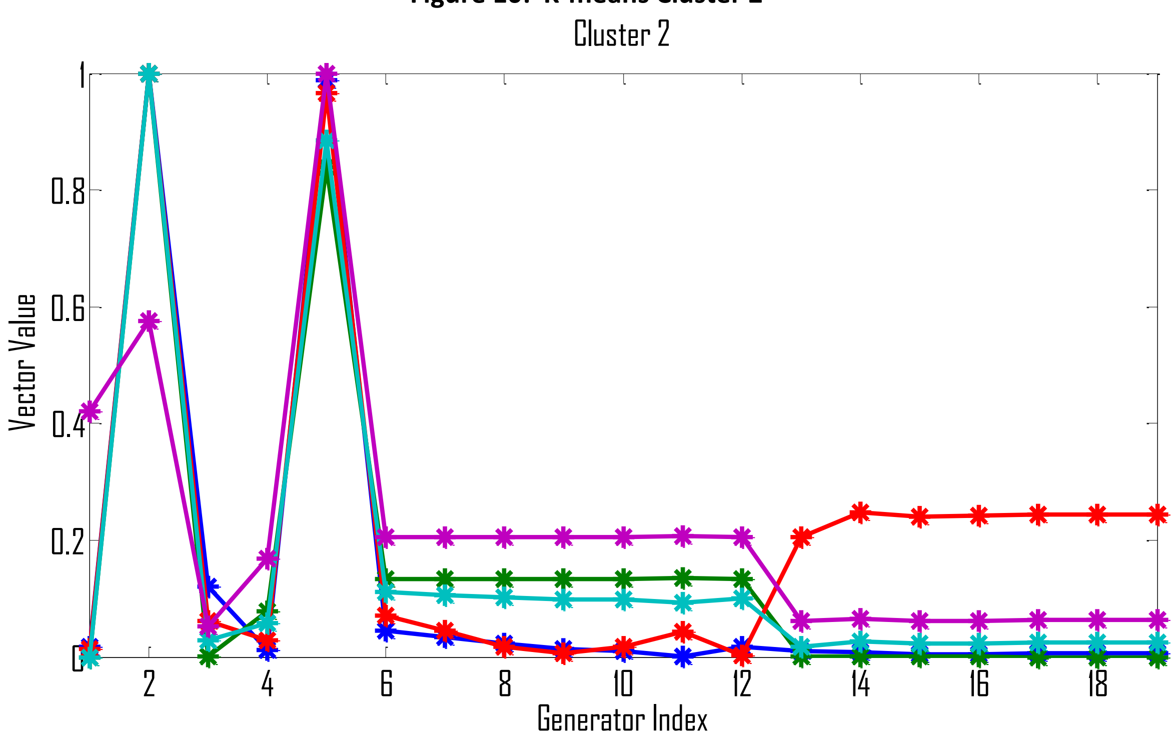


(See Output Selected Figure 20)

Cluster 2				
-0.0173	0	-0.0129	0	0.4206
1	1	1	1	0.5762
-0.1199	0.0006	-0.0626	-0.028	0.0519
-0.0121	0.0784	0.029	0.0577	0.1684
0.9888	0.8386	0.9655	0.8849	1
0.0447	0.1335	0.0708	0.1119	0.2054
0.0336	0.1326	0.0451	0.1068	0.2045
0.0234	0.1334	0.0184	0.1029	0.2053
0.0141	0.1341	-0.0058	0.0994	0.2059
0.0097	0.1344	-0.0174	0.0977	0.2062
-0.0005	0.1352	-0.0441	0.0938	0.207
0.0172	0.1338	0.0023	0.1006	0.2057
-0.0096	0.0003	0.2052	0.0179	0.0608
0.0092	0.0015	0.2476	0.0264	0.0649
0.0037	-0.0012	0.2398	0.0227	0.0613
0.0048	-0.0006	0.2414	0.0234	0.062
0.0062	0	0.2433	0.0243	0.0629
0.0062	0	0.2433	0.0243	0.0629
0.0062	0	0.2433	0.0243	0.0629

Table 9: K-means Cluster 2

Figure 10: K-means Cluster 2

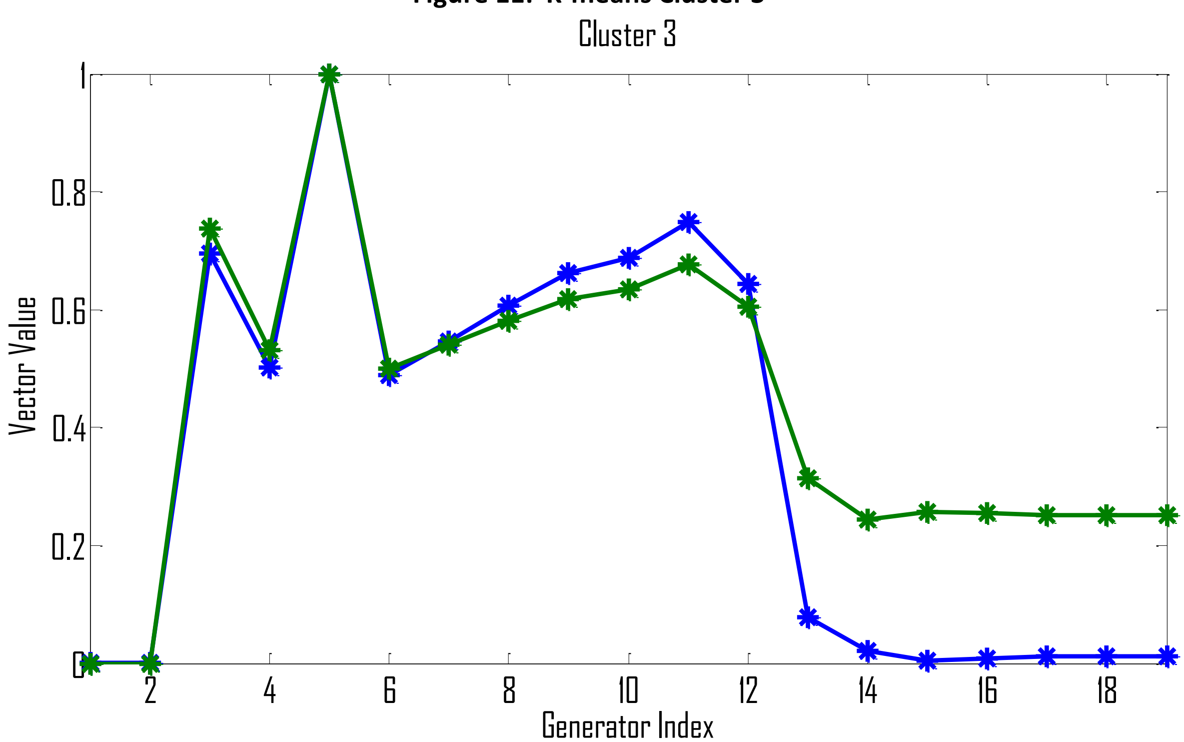


(See Output Selected Figure 21)

Cluster 3	
0.0001	0
0.0003	0
-0.6958	-0.7368
-0.5022	-0.5306
1	1
-0.4886	-0.5008
-0.5457	-0.5399
-0.6066	-0.5806
-0.6618	-0.6175
-0.6884	-0.6352
-0.7494	-0.6759
-0.6435	-0.6052
-0.0778	-0.3132
0.0211	-0.2435
0.0047	-0.2569
0.008	-0.2542
0.0121	-0.2508
0.0121	-0.2508
0.0121	-0.2508

Table 10: K-means Cluster 3

Figure 11: K-means Cluster 3

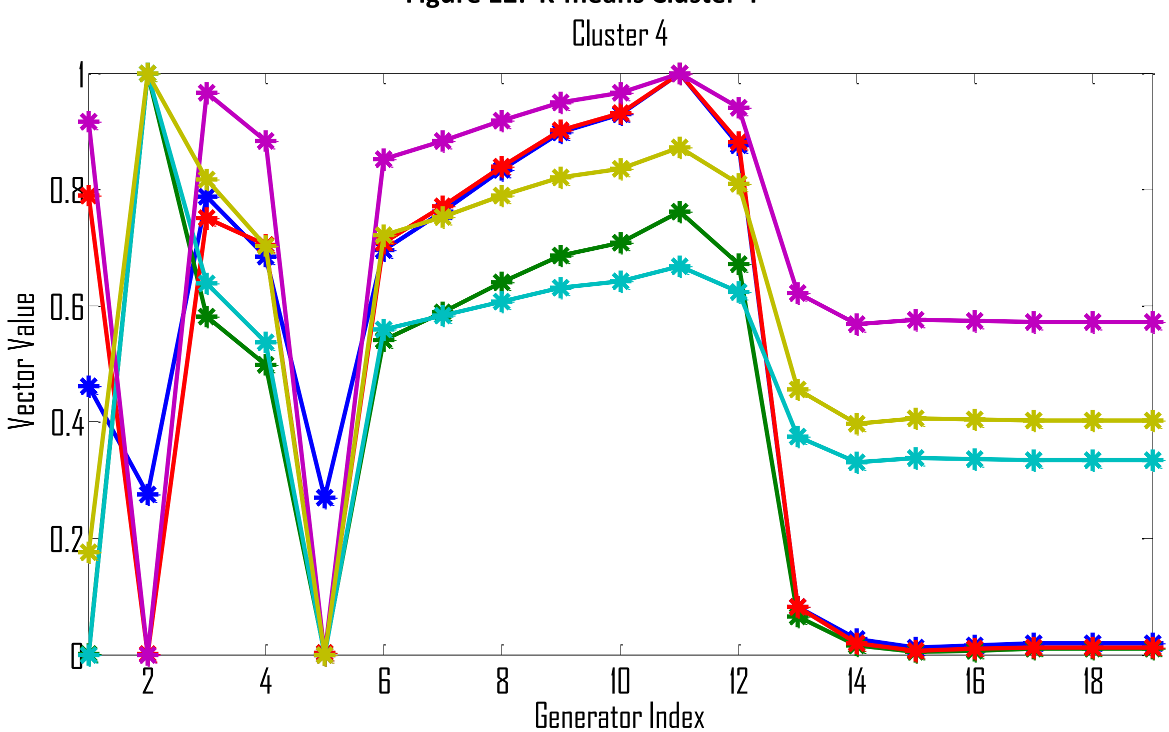


(See Output Selected Figure 22)

Cluster 4					
0.4609	-0.0014	0.7887	0	0.9167	0.1761
0.2762	1	-0.0008	1	0	1
0.7879	0.5815	0.7509	0.6378	0.9661	0.8168
0.685	0.4975	0.7039	0.5367	0.8826	0.7017
-0.2691	0.0021	0.0021	0	0	0
0.6961	0.5412	0.7081	0.5591	0.8513	0.7207
0.7614	0.5879	0.7706	0.5824	0.8839	0.7532
0.8328	0.6396	0.8392	0.6077	0.9186	0.7885
0.8975	0.6864	0.9014	0.6307	0.9501	0.8205
0.9286	0.7089	0.9313	0.6417	0.9653	0.8359
1	0.7606	1	0.667	1	0.8711
0.876	0.6708	0.8807	0.6231	0.9397	0.8099
0.0826	0.0652	0.0824	0.3744	0.6214	0.4556
-0.0273	-0.0162	-0.0201	0.3311	0.5674	0.3972
-0.0119	-0.0051	-0.0067	0.3382	0.5762	0.4065
-0.0149	-0.0073	-0.0093	0.3368	0.5745	0.4046
-0.0189	-0.0101	-0.0128	0.335	0.5722	0.4023
-0.0189	-0.0101	-0.0128	0.335	0.5722	0.4023
-0.0189	-0.0101	-0.0128	0.335	0.5722	0.4023

Table 11: K-means Cluster 4

Figure 12: K-means Cluster 4

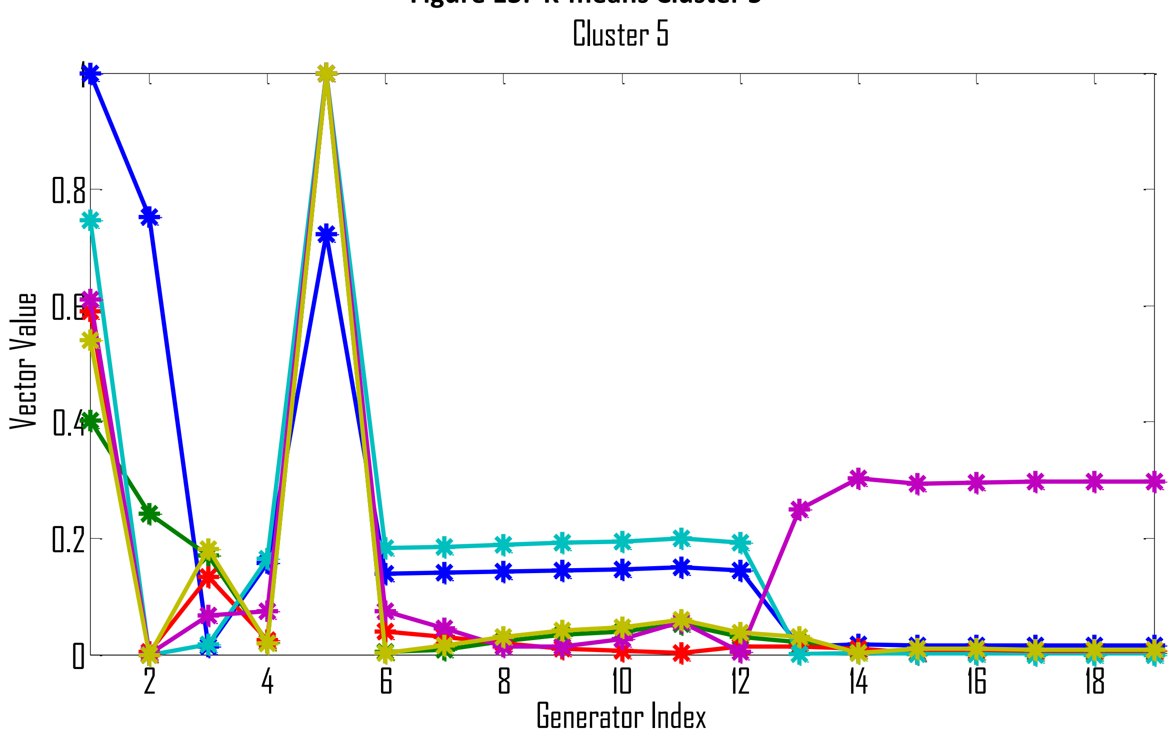


(See Output Selected Figure 23)

Cluster 5					
1	0.4019	0.5904	0.7467	0.6112	0.5405
-0.7528	0.2428	-0.0037	0	-0.0008	0
0.0146	-0.1707	-0.1341	0.0167	-0.0674	-0.1818
0.1569	-0.0218	0.024	0.1653	0.0752	-0.0202
0.7222	1	1	1	1	1
0.1382	0.0045	0.0404	0.1829	0.0738	-0.0031
0.14	-0.0091	0.0298	0.1852	0.0451	-0.0165
0.1428	-0.0223	0.0199	0.1893	0.0146	-0.0295
0.1454	-0.0343	0.0109	0.1931	-0.0129	-0.0412
0.1467	-0.0401	0.0066	0.1949	-0.0262	-0.0469
0.1496	-0.0533	-0.0034	0.1991	-0.0567	-0.0599
0.1446	-0.0303	0.0139	0.1918	-0.0038	-0.0373
0.0137	-0.0208	-0.0132	0.0004	0.2501	-0.0309
0.0183	0.0058	0.0095	0.002	0.3024	-0.0036
0.0149	-0.001	0.0031	-0.0016	0.293	-0.0107
0.0156	0.0003	0.0044	-0.0009	0.2948	-0.0093
0.0164	0.0021	0.006	0	0.2972	-0.0075
0.0164	0.0021	0.006	0	0.2972	-0.0075
0.0164	0.0021	0.006	0	0.2972	-0.0075

Table 12: K-means Cluster 5

Figure 13: K-means Cluster 5

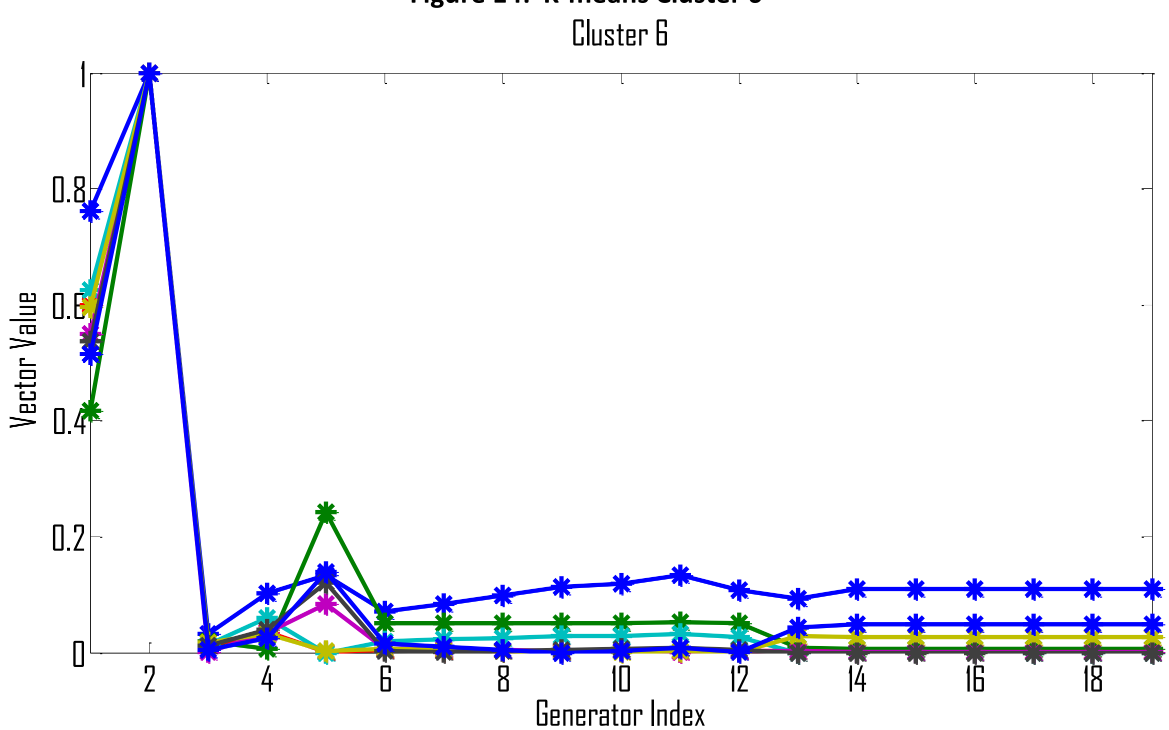


(See Output Selected Figure 24)

Cluster 6							
-0.7615	-0.4167	-0.5987	-0.6261	-0.55	-0.5962	-0.5365	-0.5148
1	1	1	1	1	1	1	1
-0.0325	0.0194	0.0117	-0.0134	0.0005	0.0179	-0.0114	0.0042
-0.1022	0.0065	-0.0364	-0.0601	-0.0345	-0.0315	-0.0403	-0.0248
-0.134	0.2414	0.0017	0	0.0846	0.0023	0.12	0.1382
-0.0712	0.0508	0.004	-0.0198	0.0073	0.0072	0.0021	0.0149
-0.0845	0.0505	0.0033	-0.0226	0.0057	0.0048	-0.0004	0.0098
-0.0992	0.0509	0.0028	-0.0253	0.0044	0.0024	-0.0026	0.0047
-0.1125	0.0512	0.0024	-0.0277	0.0032	0.0003	-0.0046	0
-0.119	0.0514	0.0022	-0.0289	0.0027	-0.0007	-0.0056	-0.0022
-0.1337	0.0518	0.0017	-0.0316	0.0014	-0.0031	-0.0078	-0.0074
-0.1081	0.0511	0.0026	-0.0269	0.0036	0.001	-0.004	0.0016
0.0934	0.0086	0.0032	0	0.0021	0.0277	0	0.0441
0.1097	0.0057	-0.0003	-0.0002	0.0005	0.0269	0	0.0484
0.1087	0.0056	0.0006	0.0002	0.0008	0.0275	0	0.0479
0.1089	0.0056	0.0004	0.0001	0.0007	0.0274	0	0.048
0.1091	0.0056	0.0002	0	0.0007	0.0272	0	0.0482
0.1091	0.0056	0.0002	0	0.0007	0.0272	0	0.0482
0.1091	0.0056	0.0002	0	0.0007	0.0272	0	0.0482

Table 13: K-means Cluster 6

Figure 14: K-means Cluster 6



(See Output Selected Figure 25)

5.3 K-Means Clustering Issues

Now the 32 price perturbation vectors have been clustered into six clusters. There is an issue that arises when using the K-means algorithm for clustering. As mentioned before the algorithm uses random initial centroid placements, and thus final cluster combinations could differ from one run to the next. In the first round of clustering this issue was solved by replicating the K-means algorithm numerous times and likely evaluating the optimal minimum sum of points to centroid mean during one of the runs.

Consider the 19×4 basis B of Table 6 as an example. First its rows were clustered using the K-means algorithm. The centroids were 4-dimensional with a dimension for each column. From there 32 new price perturbation vectors were developed and the 19×32 result set was to be clustered. These centroids are now 19-dimensional with a dimension for each row. Taking random initial centroids and getting the same clusters from one run to the next is less likely when the centroids are higher dimensional.

The K-means algorithm will converge to many local minimum, and occasionally the actual global minimum when centroids are in higher dimensions. In real markets, there will be much more than nineteen generators in consideration and the search for the optimum K-means clusters becomes even more improbable. The consequence of this is not dire however. What could be witnessed is when the market power algorithm is run, it will produce a certain selection of clustered price perturbation vectors. When it is ran a second time, one or a few of the price perturbation vectors could have switched to a new cluster.

5.4 K-means Clustering Options

There are multiple distance measuring methods available to use in Matlab's K-means algorithm. Distances between two coordinates (x_1, y_1) and (x_2, y_2) can be measured using "sqEuclidean" $\left(\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}\right)$ or "cityblock" $(|x_1 - x_2| + |y_1 - y_2|)$. Angles between vectors u and v can be measured with "cosine" $cos\theta = \frac{u*v^T}{\sqrt{u^T u}\sqrt{v^T v}}$ or similarly "correlation" $\frac{(u-\bar{u})*(v-\bar{v})^T}{\sqrt{(u-\bar{u})^T(u-\bar{u})}\sqrt{(v-\bar{v})^T(v-\bar{v})}}$ [xv].

The cosine angle between vectors could be a useful distance measurement when clustering price perturbation columns for output, however it is not appropriate for use when clustering rows of the orthonormal basis B. This will be demonstrated with the following simple seven row orthonormal basis.

$$\begin{bmatrix} 0.499 & -0.5061 \\ 0.501 & -0.4938 \\ 0.499 & 0.506 \\ 0.501 & 0.4938 \\ 0.005 & 0.0051 \\ 0.005 & -0.0051 \\ -0.005 & -0.0051 \end{bmatrix} =$$

$$\begin{bmatrix} 0.7107\angle -45.4061° \\ 0.7035\angle -44.5875° \\ 0.7107\angle 45.4061° \\ 0.7035\angle 44.5855° \\ 0.0071\angle 45.4061° \\ 0.0071\angle -45.4061° \\ 0.0071\angle -134.5959° \end{bmatrix}$$

Table 14: Clustering Demonstration

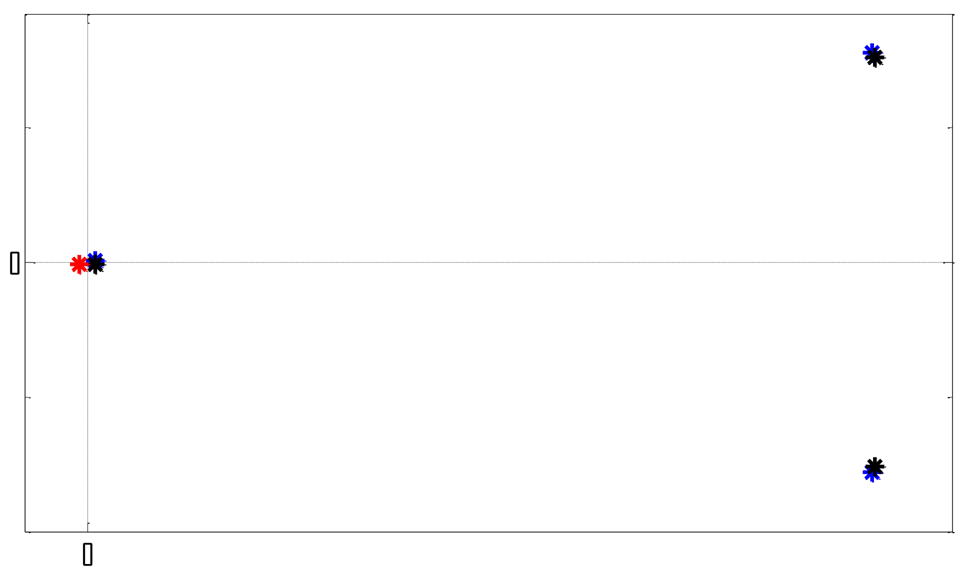


Figure 15: Clustering Demonstration

It is clear there are three clusters, and there are three points with small magnitudes clustered around the origin. The following is the cluster results gotten when clustering using

Euclidean distance versus using cosine angles between vectors.

Point		Euclidean Clusters	Cosine Clusters
1	$0.7107\angle -45.4061°$	1	1
2	$0.7035\angle -44.5875°$	1	1
3	$0.7107\angle 45.4061°$	2	2
4	$0.7035\angle 44.5855°$	2	2
5	$0.0071\angle 45.4061°$	3	2
6	$0.0071\angle -45.4061°$	3	1
7	$0.0071\angle -134.5959°$	3	3

Table 15: Euclidean and Cosine Clusters

With the cosine clustering, the very last point becomes its own cluster. That is not desirable however because it's magnitude is so small at 0.0071. It is preferable that the last three rows with small magnitudes be clustered together. No matter how hard it is tried to make row 7 become a large and thus interesting price perturbation, it will always remain small in magnitude. This is demonstrated using eigen-analysis and setting row 7 to be B_+.

$$\begin{bmatrix} B_- \\ - \\ B_+ \end{bmatrix} = \left[\begin{array}{cc} 0.499 & -0.5061 \\ 0.501 & -0.4938 \\ 0.499 & 0.506 \\ 0.501 & 0.4938 \\ 0.005 & 0.0051 \\ 0.005 & -0.0051 \\ \hline -0.005 & -0.0051 \end{array}\right]$$

$$\max\{x^T(B_+^T B_+ - B_-^T B_-)x\} \to x_{max} = \begin{bmatrix} 0.0583 \\ -0.9983 \end{bmatrix}$$

$$\begin{bmatrix} 0.499 & -0.5061 \\ 0.501 & -0.4938 \\ 0.499 & 0.506 \\ 0.501 & 0.4938 \\ 0.005 & 0.0051 \\ 0.005 & -0.0051 \\ -0.005 & -0.0051 \end{bmatrix} x_{max} = \begin{bmatrix} 0.5343 \\ 0.5222 \\ -0.476 \\ -0.4637 \\ -0.0022 \\ 0.0054 \\ 0.0048 \end{bmatrix} \sim \begin{bmatrix} 1 \\ 0.9772 \\ -0.8909 \\ -0.8678 \\ -0.0041 \\ 0.0101 \\ 0.009 \end{bmatrix} \leftarrow \text{Row 7 still small in magnitude}$$

In general when there are many rows that are small in magnitude, they should be clustered together. Therefore the Euclidean distance between points will be used for measurement when clustering the rows of B. This is not a problem however during our second round of clustering price perturbation vectors. Each column is scaled to have a maximum entry of 1.0, and thus there really are no columns drastically smaller in magnitude than others. The problem illustrated in Figure 15 is thus not applicable when clustering price perturbation vectors for output. The next issue at hand is determining if cosine angle clustering actually outperforms Euclidean distance clustering. This hints at the need for some measure of quality of results which will be discussed next.

6. Assessing Quality of Price Perturbation Results

The last challenge in writing this market power algorithm is the need to develop a means of measuring the quality of results. This will assist in multiple ways like determining which K-means distance measurement suits desires best. It will primarily be used to select the best performing price perturbation vector from its cluster set, thus filtering results and giving a more easily observable output to the user. Earlier in Section 4.2 K-Means Clustering, 32 price perturbation vectors were clustered into 6 clusters. The best vector from each cluster should be selected, and the 6 vectors should be outputted as final results. Consider the following two price perturbation vectors for example.

|vec 1|	|vec 2|
0.1098	0.0001
0.0512	0.0003
0.5613	0.6958
0.3787	0.5022
1.0000	1.0000
0.3623	0.4886
0.4082	0.5457
0.4569	0.6066
0.5011	0.6618
0.5224	0.6884
0.5712	0.7494
0.4865	0.6435
0.0628	0.0778
0.0177	0.0211
0.0037	0.0047
0.0064	0.0080
0.0100	0.0121
0.0100	0.0121
0.0100	0.0121

Table 16: Vector Quality Assessment

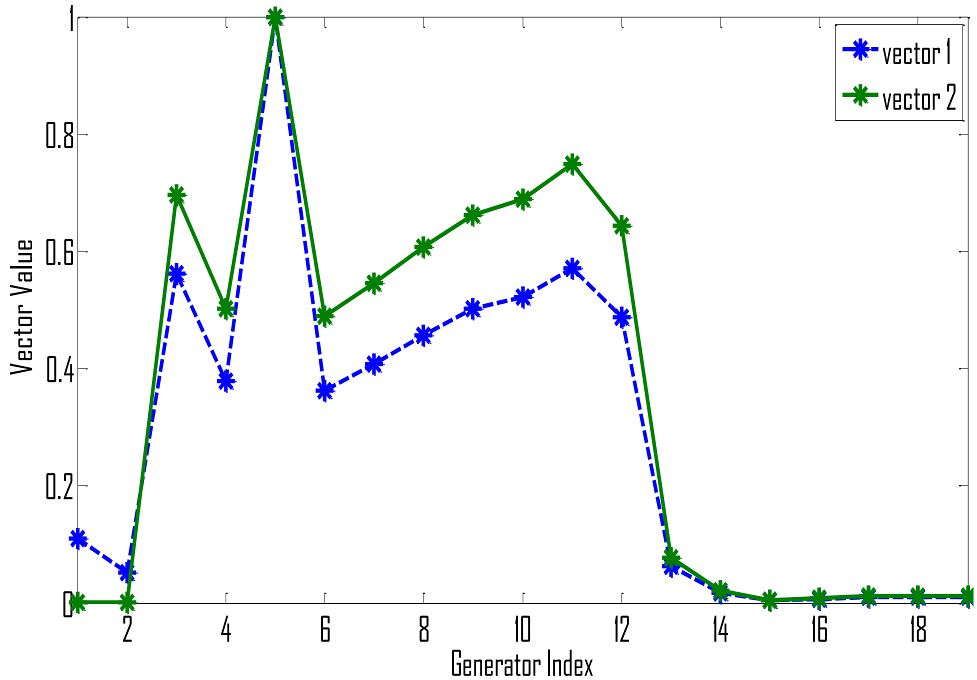


Figure 16: Vector Quality Assessment

It may appear vague as to which vector would be preferred over another. The price perturbation scenario of the solid green line shows generator 5 maximally raising its price and generators 3, 4 and 6 – 12 moderately raising prices without affecting dispatch. The dashed blue line similarly shows generator 5 maximally raising its price, but the prices of generators 3, 4 and 6 – 12 are decreased from about \$0.65 to \$0.50. Generators 1 and 2 also now have prices up around \$0.08 as well. The question is which price perturbation scenario is more interesting? It is almost a matter of preference, but a consistent method of calculation should be implemented.

6.1 One-Norm Quality Assessment

The first measure of success suggested is using the one-norm, the sum of the absolute valued vector entries. A small one-norm would then imply success so as many of the price increments as possible are driven to zero, $minimize\{\|\Delta y\|_1\}$. The one-norms of the above two vectors are 5.5302 for vector 1 and 6.7305 for vector 2. This implies the dashed blue vector 1 would be chosen as the better of the two and outputted for final results, but this may not actually be the price perturbation vector wanted for selection.

6.2 Eigenvalue Quality Assessment

The purpose of the market power monitoring tool is to maximize large entries and minimize small entries accomplished using eigen-analysis. It appears the solid green vector 2 fits this criterion better.

$$Bx = \begin{bmatrix} B_+ \\ - \\ B_- \end{bmatrix} x = \begin{bmatrix} B_+x \\ - \\ B_-x \end{bmatrix} \begin{matrix} \leftarrow large\ entries \\ \\ \leftarrow small\ entries \end{matrix}$$
$$(m\ rows\ x\ 1\ column)$$

$$\lambda_{max} = \max\{x^T(B_+^T B_+ - B_-^T B_-)x\}$$

Intuitively it may make sense to simply select the price perturbation vector having being generated by the eigenvector with largest eigenvalue. This implies the greatest separation between B_+x and B_-x as desired, but there is a caveat in that switching any given row or more rows out of B_- and into B_+ will automatically increase or not change the eigenvalue λ. $(B_+x)^T(B_+x) - (B_-x)^T(B_-x)$ is to be maximized.

$$Bx = \begin{bmatrix} B_+ \\ - \\ B_- \end{bmatrix} x = \begin{bmatrix} B_+x \\ - \\ B_-x \end{bmatrix} = \begin{bmatrix} b_{+,1}x \\ b_{+,2}x \\ \vdots \\ b_{+,r}x \\ \hline b_{-,r+1}x \\ b_{-,r+2}x \\ \vdots \\ b_{-,m}x \end{bmatrix} = \Delta y = \begin{bmatrix} \Delta y_{+,1} \\ \Delta y_{+,2} \\ \vdots \\ \Delta y_{+,r} \\ \hline \Delta y_{-,r+1} \\ \Delta y_{-,r+2} \\ \vdots \\ \Delta y_{-,m} \end{bmatrix} \quad (m \times 1)$$

$$\lambda = (B_+x)^T(B_+x) - (B_-x)^T(B_-x)$$
$$= \left(\Delta y_{+,1}^2 + \Delta y_{+,2}^2 + \cdots + \Delta y_{+,r}^2\right) - \left(\Delta y_{-,r+1}^2 + \Delta y_{-,r+2}^2 + \cdots + \Delta y_{-,m}^2\right)$$

All $\Delta y_{\pm,i}^2 \geq 0$, so one can see moving any row from B_- to B_+ will always increase or not change the eigenvalue λ. Furthermore the eigenvalue λ will max out at 1.0 when $\#rows(B_-) \leq n-1$ or equivalently $\#rows(B_+) \geq m-n+1$.

Consider a basis B with $m > n$, as typically there will be more generators than line constraints plus one. Assuming that the n columns are linearly independent the basis will have rank $r = n$.

$$B = \begin{bmatrix} b_{1,1} & \dots & b_{1,n} \\ \vdots & & \vdots \\ b_{m,1} & \dots & b_{m,n} \end{bmatrix} \quad (m \times n)$$

$$m > n, \ \ rank\ r = n$$

Nullity is defined by $Bx_1 = \cdots = Bx_q = 0$ where q equals the nullity υ. The rank-nullity theorem shows $rank\ r + nullity\ \nu = \min(m, n)$ [xi]. In this example, $n + nullity\ \nu = $ n, which implies the nullity $\upsilon = 0$, or that there is no null space for B. However there is a guarantee that at least $n-1$ of the rows of Bx can be nullified.

$$Bx = \begin{bmatrix} B_- \\ - \\ B_+ \end{bmatrix} x = \left[\begin{array}{ccc} b_{1,1} & \dots & b_{1,n} \\ \vdots & & \vdots \\ b_{n-1,1} & \dots & b_{n-1,n} \\ \hline b_{n,1} & \dots & b_{n,n} \\ \vdots & & \vdots \\ b_{m,1} & \dots & b_{m,n} \end{array}\right] x = \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline [b_{n,1} \ \dots \ b_{n,n}]x \\ \vdots \\ [b_{m,1} \ \dots \ b_{m,n}]x \end{array}\right]$$

For $n-1$ number of rows of Bx to exactly equal zero, the number of rows in B_- must equal $n-1$ or equivalently the number of rows in B_+ must equal $m-n+1$. If all rows of $B_-x = 0$ then the eigenvalue $\lambda = x^T(B_+^TB_+ - B_-^TB_-)x = 1.0$. Therefore the caveat of using eigenvalues as a measure of vector quality is that an eigenvalue $\lambda = 1.0$ will always be achievable by moving $m - n + 1$ rows or more into B_+. The price perturbation vectors generated by eigenvectors with the largest eigenvalues would always be selected for output, but it can be shown that these vectors are often not the best output choice of their clusters.

6.3 Midpoint Quality Assessment

It seems that minimizing the one-norm of price perturbation vectors does not necessarily select the best vector, and using eigenvalues biases some vectors over others. It is preferred to either have entries that are large tending toward 1.0 or very low tending toward 0. This would imply that entries floating halfway around 0.5 are unwanted. Therefore a measure that penalizes entries near 0.5 and rewards values far from 0.5 might select the results preferred. Here it is proposed to do so by subtracting the absolute value of a price perturbation vector by 0.5, computing the one-norm, and selecting the vector with this maximum one-norm for output results.

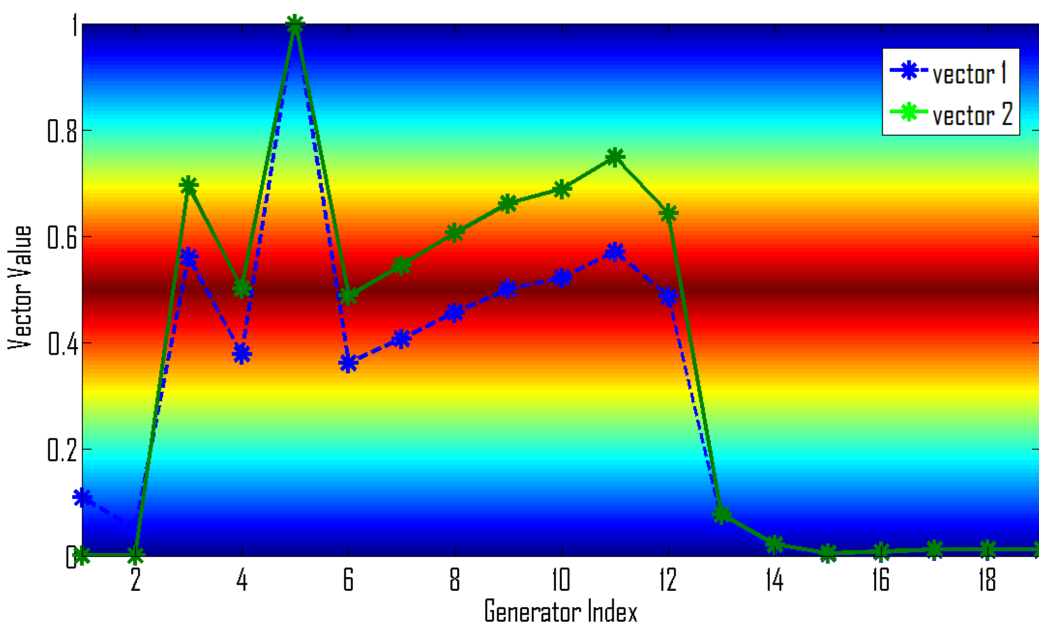


Figure 17: Midpoint Quality Assessment

The above figure graphically shows where it is desired for rows of the absolute value price perturbation vectors to fall, with the reference line 0.5 in red being the least desired. The dashed blue vector 1 appears to have more points floating around 0.5 than the solid green vector 2, and is thus less desired for identifying market power price increment scenarios. The procedure used for output selection is demonstrated below.

	$\left\| \|\textbf{vec 1}\| - 0.5 \right\|$		$\left\| \|\textbf{vec 2}\| - 0.5 \right\|$
	0.3902		0.4999
	0.4488		0.4997
	0.0613		0.1958
	0.1213		0.0022
	0.5000		0.5000
	0.1377		0.0114
	0.0918		0.0457
	0.0431		0.1066
	0.0011		0.1618
	0.0224		0.1884
	0.0712		0.2494
	0.0135		0.1435
	0.4372		0.4222
	0.4823		0.4789
	0.4963		0.4953
	0.4936		0.4920
	0.4900		0.4879
	0.4900		0.4879
	0.4900		0.4879
$norm\left(\left\| \|\textbf{vec 1}\| - 0.5 \right\|\right)_1$	5.2818	$norm\left(\left\| \|\textbf{vec 2}\| - 0.5 \right\|\right)_1$	5.9565

Table 17: Midpoint Quality Assessment

Vector 2 has the greater one-norm, and thus has more absolute value points away from 0.5. Using this measure of vector quality, the solid green vector 2 would be selected for output. One problem does arise however which will be demonstrated with a basic example.

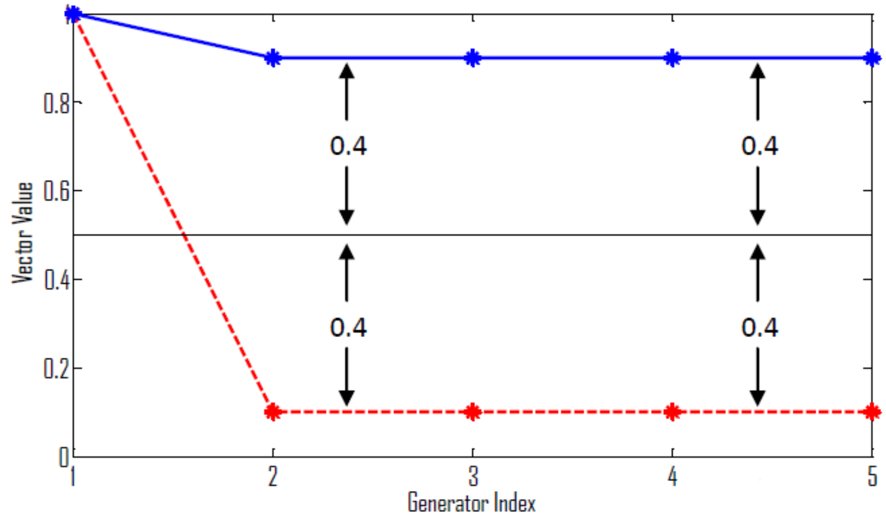


Figure 18: 0.5 Reference Line

Both of the vectors are an equal distance from the reference line at 0.5. If the solid blue vector were slightly further from or the dashed red vector slightly closer to 0.5, then the solid blue vector would be chosen for output. In general a vector with many small entries and few large entries is wanted, like the dashed red vector, and not the other way around. To avoid selection of vectors with many large entries over vectors with many small entries the reference line will be slightly shifted above 0.5. Doing so will favor vectors having many small entries.

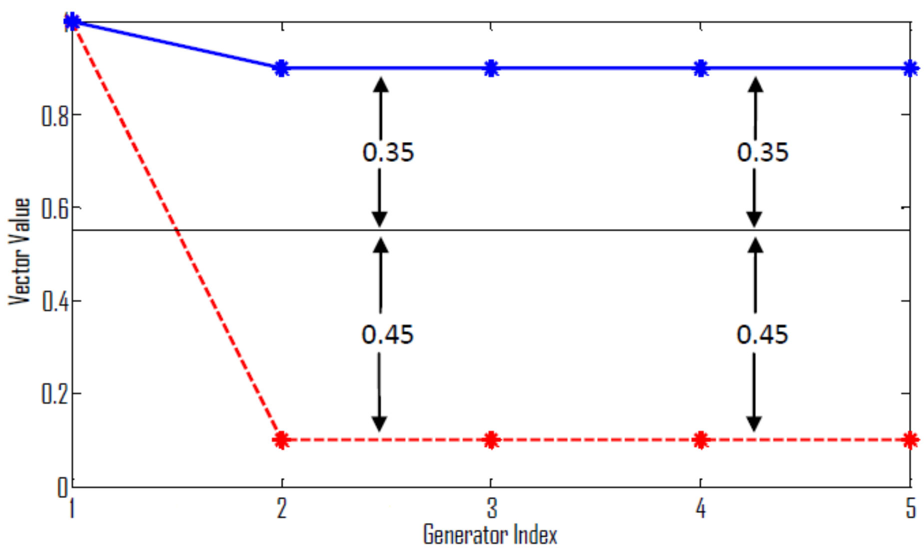


Figure 19: 0.55 Reference Line

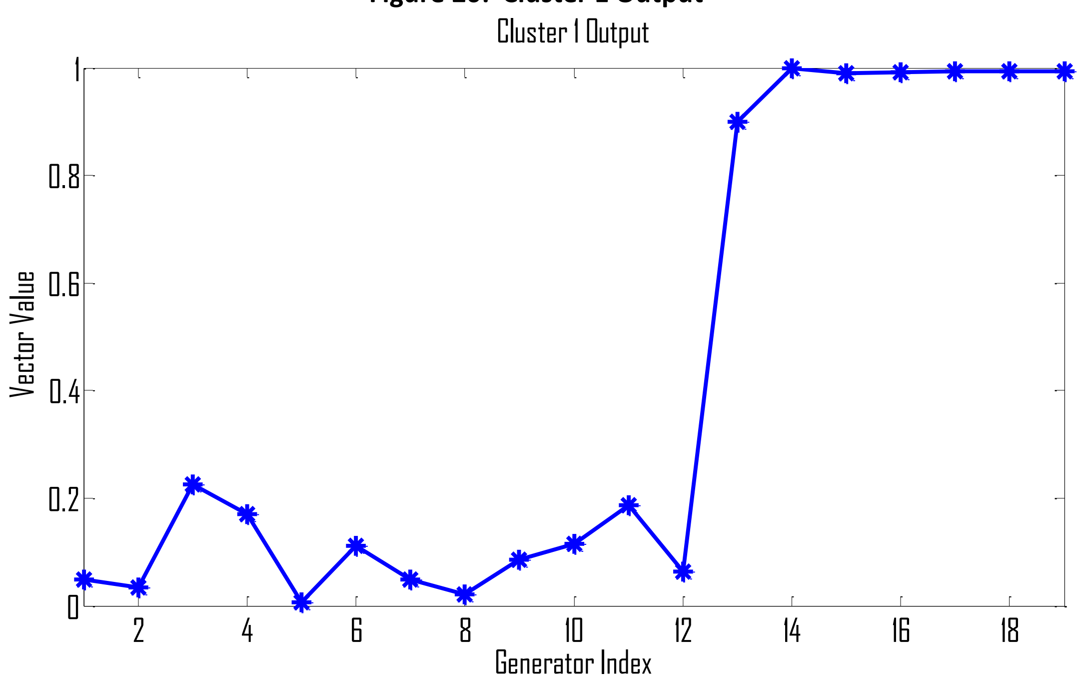
6.4 Output Price Perturbation Vectors

Lastly this midway quality measurement will be used to select output price perturbation vectors for the cluster examples of Table 8 - Table 13 from Section 5.2 K-Means Column Clustering Example.

$$\begin{bmatrix} 0.0497 \\ 0.0338 \\ 0.2258 \\ 0.1707 \\ -0.0061 \\ 0.111 \\ 0.0488 \\ -0.0215 \\ -0.0851 \\ -0.1158 \\ -0.1861 \\ -0.064 \\ 0.8988 \\ 1 \\ 0.9897 \\ 0.9918 \\ 0.9944 \\ 0.9944 \\ 0.9944 \end{bmatrix}$$

Table 18: Output 1

Figure 20: Cluster 1 Output

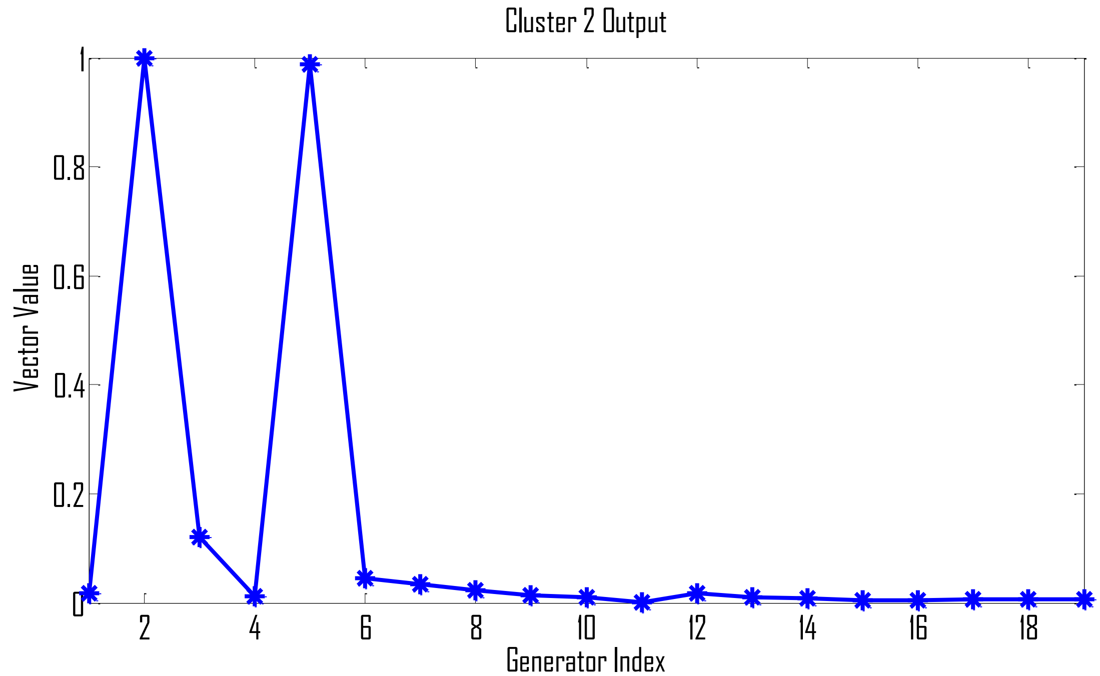


(See Cluster Chosen From Figure 9)

$$\begin{bmatrix} -0.0173 \\ 1 \\ -0.1199 \\ -0.0121 \\ 0.9888 \\ 0.0447 \\ 0.0336 \\ 0.0234 \\ 0.0141 \\ 0.0097 \\ -0.0005 \\ 0.0172 \\ -0.0096 \\ 0.0092 \\ 0.0037 \\ 0.0048 \\ 0.0062 \\ 0.0062 \\ 0.0062 \end{bmatrix}$$

Table 19: Output 2

Figure 21: Cluster 2 Output

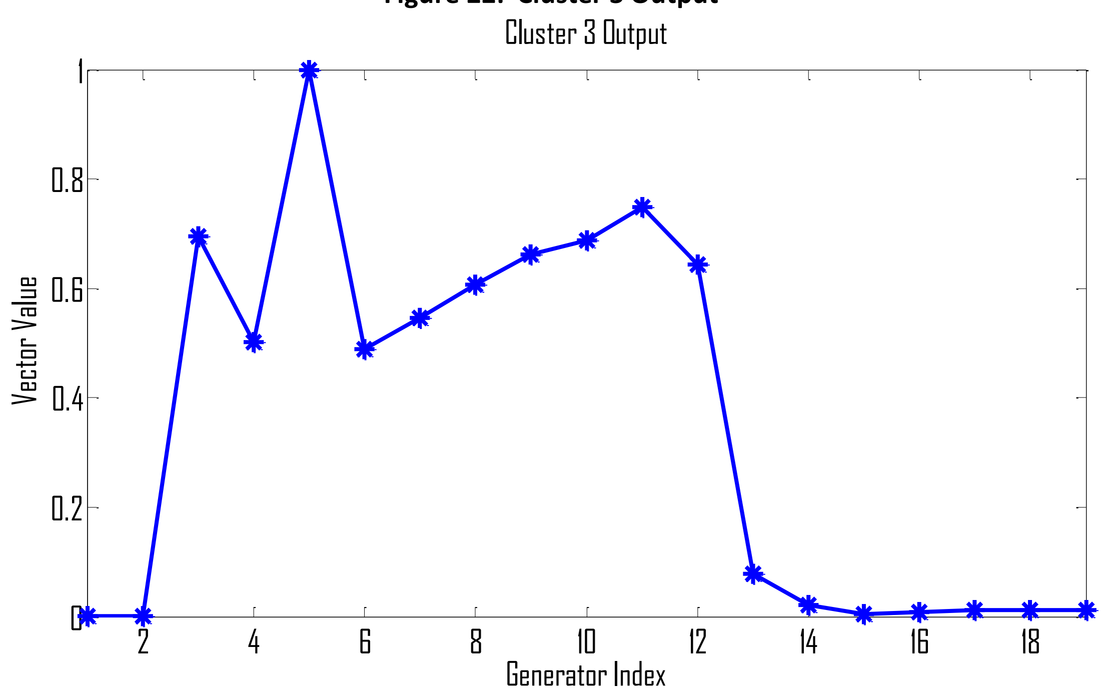


(See Cluster Chosen From Figure 10)

$$\begin{bmatrix} 0.0001 \\ 0.0003 \\ -0.6958 \\ -0.5022 \\ 1 \\ -0.4886 \\ -0.5457 \\ -0.6066 \\ -0.6618 \\ -0.6884 \\ -0.7494 \\ -0.6435 \\ -0.0778 \\ 0.0211 \\ 0.0047 \\ 0.008 \\ 0.0121 \\ 0.0121 \\ 0.0121 \end{bmatrix}$$

Table 20: Output 3

Figure 22: Cluster 3 Output

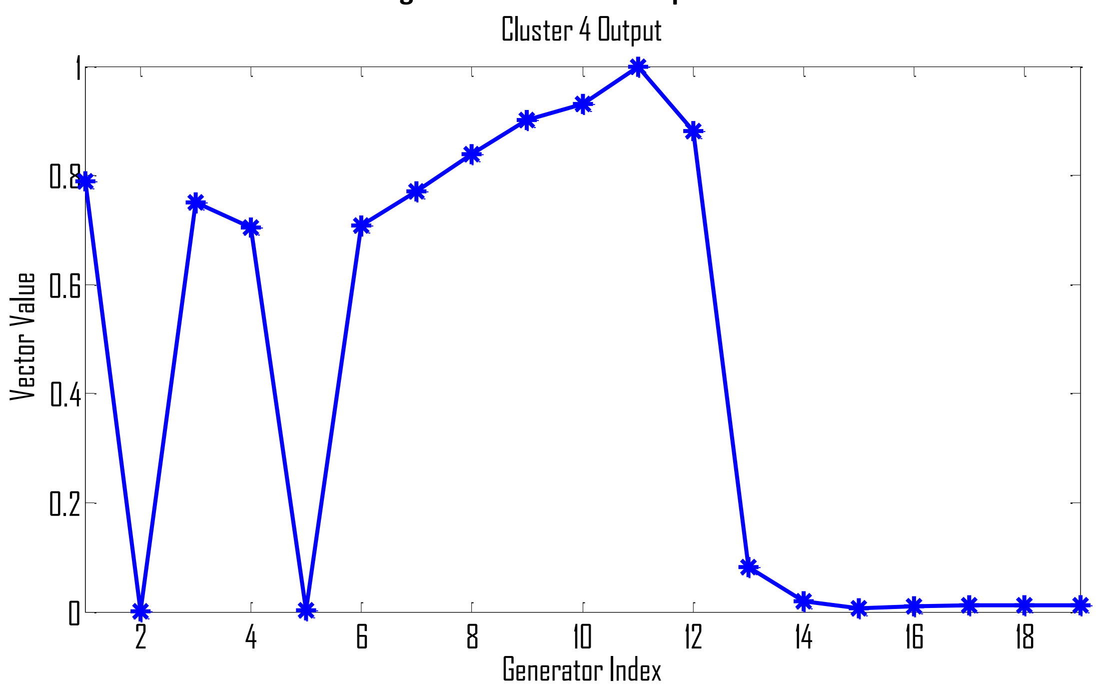


(See Cluster Chosen From Figure 11)

$$\begin{bmatrix} 0.7887 \\ -0.0008 \\ 0.7509 \\ 0.7039 \\ 0.0021 \\ 0.7081 \\ 0.7706 \\ 0.8392 \\ 0.9014 \\ 0.9313 \\ 1 \\ 0.8807 \\ 0.0824 \\ -0.0201 \\ -0.0067 \\ -0.0093 \\ -0.0128 \\ -0.0128 \\ -0.0128 \end{bmatrix}$$

Table 21: Output 4

Figure 23: Cluster 4 Output

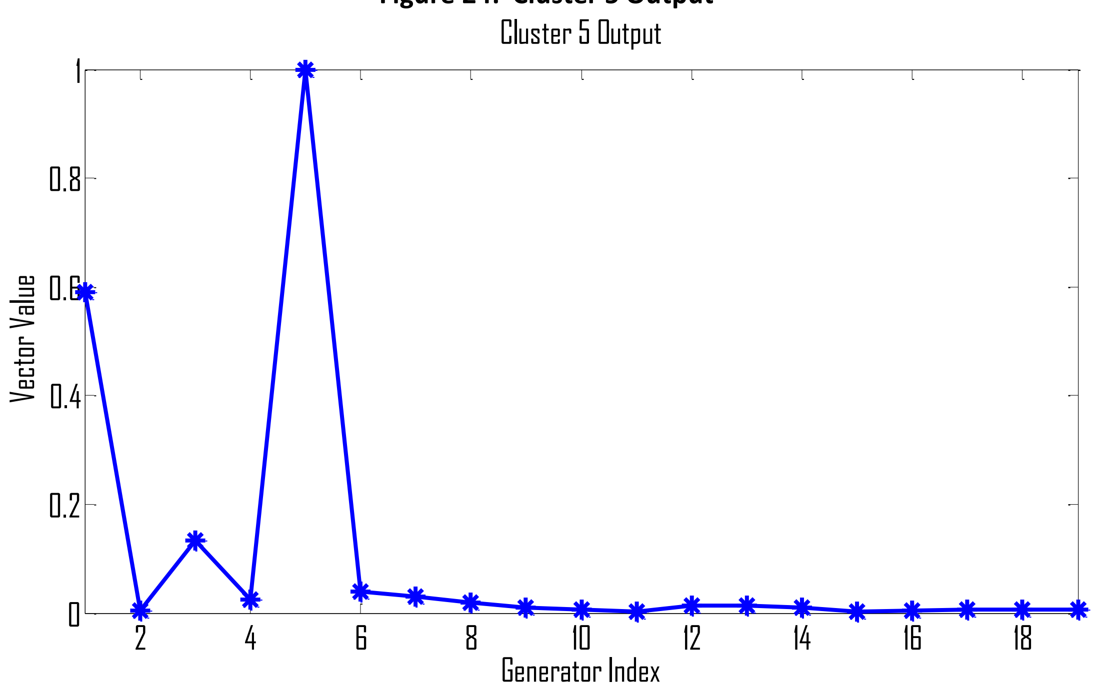


(See Cluster Chosen From Figure 12)

$$\begin{bmatrix} 0.5904 \\ -0.0037 \\ -0.1341 \\ 0.0240 \\ 1 \\ 0.0404 \\ 0.0298 \\ 0.0199 \\ 0.0109 \\ 0.0066 \\ -0.0034 \\ 0.0139 \\ -0.0132 \\ 0.0095 \\ 0.0031 \\ 0.0044 \\ 0.006 \\ 0.006 \\ 0.006 \end{bmatrix}$$

Table 22: Output 5

Figure 24: Cluster 5 Output

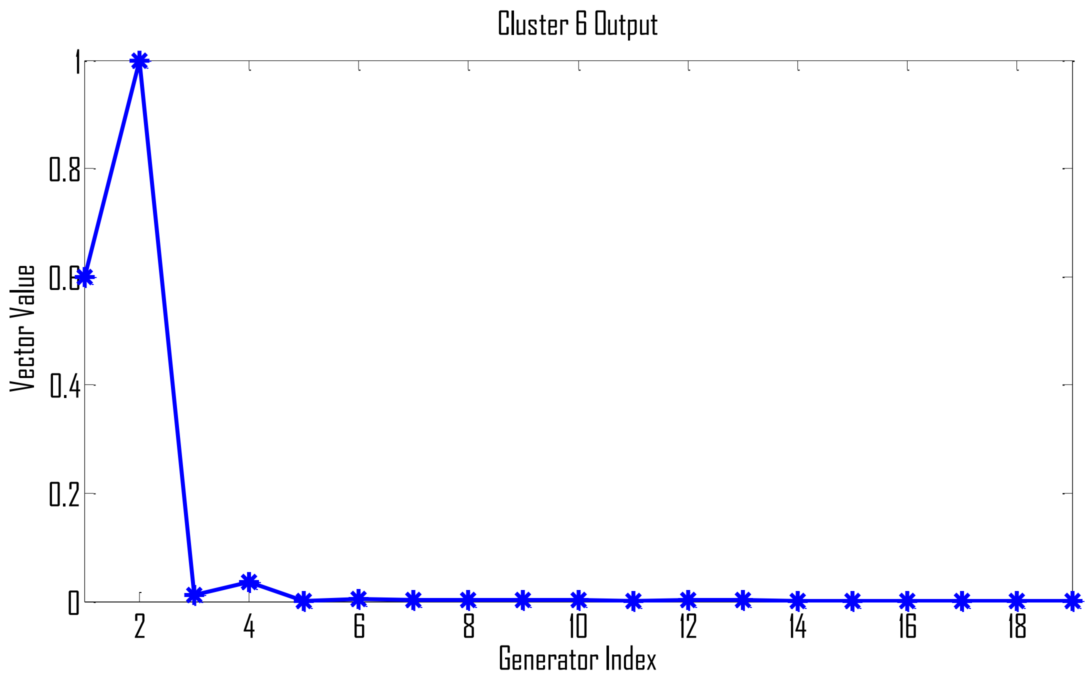


(See Cluster Chosen From Figure 13)

$$\begin{bmatrix} -0.5987 \\ 1 \\ 0.0117 \\ -0.0364 \\ 0.0017 \\ 0.004 \\ 0.0033 \\ 0.0028 \\ 0.0024 \\ 0.0022 \\ 0.0017 \\ 0.0026 \\ 0.0032 \\ -0.0003 \\ 0.0006 \\ 0.0004 \\ 0.0002 \\ 0.0002 \\ 0.0002 \end{bmatrix}$$

Table 23: Output 6

Figure 25: Cluster 6 Output



(See Cluster Chosen From Figure 14)

7. Conclusions

The methods presented in this paper can be used to quickly and efficiently identify generator suppliers possessing market power potential. A DC optimal power flow is first performed on the network to identify binding constraints on congested lines. With the binding constraints known a dispatch/offer price sensitivity matrix can be easily computed from a linear programming tableau, and a basis B is constructed from the null space of the sensitivity matrix. Linear combinations of the columns of B produce new price perturbation vectors as a combined effect of all line constraints. The desired form of the price perturbations vectors is having few large entries and many small entries, so eigen-analysis is used to optimize this separation.

Next the rows of basis B are organized into k_1 clusters, specified by the user, using the K-means clustering algorithm. Similar generators are clustered together to avoid an unreasonably large number of future computations. All combinations of row clusters are evaluated in the eigen-analysis equation (2) which produces 2^{k_1} new price perturbation vectors. To avoid redundant results the price perturbation vectors are organized into k_2 clusters, specified by the user, using the cosine angle between vectors and the K-means clustering algorithm. Finally the best performing price perturbation from each cluster is selected for output using the norm described in Section 6.3 Midpoint Quality Assessment.

Appendix – Matlab Routine

```
function price_perturbation_vectors =
MPP(B,num_gen_clusters,num_output_clusters)

tic

%--------------------------------------------------------------------------
-
% B: add column of ones and/or orthonormalize if necessary

[num_rows num_cols] = size(B);
b=ones(num_rows,1);
x=B\b;
if roundn(max(abs(B*x-b)),-5)>0
    B=[B b]; % add column ones to end
    num_cols=num_cols+1;
    B=orth(B);
elseif min(sum(B'*B==eye(num_cols)))~=num_cols % orthonormalize if needed
    B=orth(B);
end

%--------------------------------------------------------------------------
-
% Clustering

size_vecs=2^num_gen_clusters;
warning('off','stats:kmeans:EmptyCluster') % turn off K-means error
message
try clusters=kmeans(B,num_gen_clusters,'replicates',100); % generator
clusters
catch ME
    clusters=(1:num_rows)';
    size_vecs=2^num_rows;
end


%--------------------------------------------------------------------------
-
% Finding 1-Norms

length_counter=1;
counter=[1 zeros(1,num_gen_clusters-1)];
counter2=1;
midpoint_norms=zeros(1,size_vecs); % quality assessment
vecs=zeros(num_rows,size_vecs);
B1s=zeros(num_rows,size_vecs);
B1_lengths=zeros(1,size_vecs);
while length_counter<=num_gen_clusters
    B1_uncluster=[];
    for i=1:length_counter
        B1_uncluster=[B1_uncluster; find(clusters==counter(i))];
    end
```

```
    B1_uncluster=sort(B1_uncluster);
    B1=B(B1_uncluster,:); % B+
    B2=B(setdiff(1:num_rows,B1_uncluster),:); % B-
    [X D]=eig(B1'*B1-B2'*B2);vec=B*X(:,end);
    peak=find(abs(vec)==max(abs(vec)),1,'first');
    vec=vec/vec(peak);
    midpoint_norms(counter2)=norm(abs(vec)-0.55,1);
    B1s(1:length(B1_uncluster),counter2)=sort(B1_uncluster);
    B1_lengths(counter2)=length(B1_uncluster);
    vecs(:,counter2)=vec;
    if counter(length_counter)<num_gen_clusters
        counter(length_counter)=counter(length_counter)+1;
    elseif num_gen_clusters-counter(1)>=length_counter
        for i=1:length_counter-1
            if counter(length_counter-i)<num_gen_clusters-i
                temp=counter(length_counter-i);
                for j=1:i+1
                    counter(length_counter-i+j-1)=temp+j;
                end
                break
            elseif i==length_counter-1
                length_counter=num_gen_clusters+1;
            end
        end
    else
        length_counter=length_counter+1;
        for i=1:length_counter
            counter(i)=i;
        end
    end
    counter2=counter2+1;
end

% appending case where B1=[] and B2=B
[X D]=eig(-B'*B);vec=B*X(:,end);
peak=find(abs(vec)==max(abs(vec)),1,'first');
vecs(:,size_vecs)=vec/vec(peak);
midpoint_norms(size_vecs)=norm(abs(vec)-0.55,1);

%--------------------------------------------------------------------------
-
%Filter Results

try
angle_clusters=kmeans(abs(vecs'),num_output_clusters,'distance','cosine','
replicates',100)'; % cluster price perturbation vectors by cosine angle
catch ME
    angle_clusters=1:size(vecs,2);
end

vecs2=[];
B1s2=[];
B1_lengths2=[];
many_clustered=[];
```

```
num_clustered=[];
angle_clusters2=[];
for i=1:size_vecs
    clustered=find(angle_clusters==i);
    if ~isempty(clustered)
        if size(clustered,2)>1
            [max_midpoint_norm position]=max(midpoint_norms(clustered));
            vecs2=[vecs2 vecs(:,clustered(position))];
            num_clustered=[num_clustered size(clustered,2)];
            B1s2=[B1s2 B1s(:,clustered(position))];
            B1_lengths2=[B1_lengths2 B1_lengths(clustered(position))];
            many_clustered=[many_clustered 1];
            angle_clusters2=[angle_clusters2 i];
        else
            vecs2=[vecs2 vecs(:,clustered)];
            num_clustered=[num_clustered 1];
            B1s2=[B1s2 B1s(:,clustered)];
            B1_lengths2=[B1_lengths2 B1_lengths(clustered)];
            many_clustered=[many_clustered 0];
            angle_clusters2=[angle_clusters2 i];
        end
    end
end

price_perturbation_vectors=vecs2;

j=1;
set(0,'DefaultFigureWindowStyle','docked')
for i=1:size(num_clustered,2);
    if many_clustered(i)==0
        figure(j), plot(abs(vecs2(:,i))), axis([1 num_rows 0 1])
        B1=B1s2(1:B1_lengths2(i),i)';
        title(['B+ = ' num2str(B1)])
        j=j+1;
    else
        clustered=find(angle_clusters==angle_clusters2(i));
        figure(j), plot(abs(vecs(:,clustered))), axis([1 num_rows 0 1])
        title(['number similar vectors = ' num2str(size(clustered,2))])
        figure(j+1), plot(abs(vecs2(:,i))), axis([1 num_rows 0 1])
        B1=B1s2(1:B1_lengths2(i),i)';
        title(['B+ = ' num2str(B1)])
        j=j+2;
    end
end
figure(j), plot(abs(vecs2)), axis([1 num_rows 0 1])
title(['total number vectors = ' num2str(size(num_clustered,2))])

warning('on','stats:kmeans:EmptyCluster') % return K-means warning to on

toc

end
```

References

[i] A. K. David, F. Wen, "Market power in electricity supply," *IEEE Transactions on Energy Conversion* , vol. 16, no.4, pp.352-360, Dec 2001.

[ii] J. C. Dalton, "Assessing the competitiveness of restructured generation service markets," *The Electricity Journal*, vol. 10, Issue 3, April 1997, Pages 30-39, ISSN 1040-6190, DOI: 10.1016/S1040-6190(97)80375-2.

[iii] Peng Wang; Yu Xiao; Yi Ding; , "Nodal market power assessment in electricity markets," *Power Systems, IEEE Transactions on* , vol.19, no.3, pp. 1373- 1379, Aug. 2004

[iv] S. Borenstein, J. Bushnell, C. R. Knittel, "Market power in electricity markets: Beyond concentration measures," *Energy Journal*, vol. 20, no. 4, p 65-88, 1999.

[v] PJM Market Monitoring Unit, "2007 State of the Market Report, Volume 2: Detailed Anlaysis," Appedix L- Three Pivotal Supplier Test, March, 2008.

[vi] Lesieutre, B.C. Dec. 9, 2008. Real Time Market Power Monitoring. PSERC Teleseminar.

[vii] Lesieutre, B.C.; Thomas, R.J.; Mount, T.D.; , "A revenue sensitivity approach for the identification and quantification of market power in electric energy markets," *Power Engineering Society General Meeting, 2003, IEEE* , vol.2, no., pp. 842 Vol. 2, 13-17 July 2003

[viii] C.E. Murillo-Sanchez, S.M. Ede, T.D. Mount, R.J. Thomas, and R.D. Zimmerman, "An Engineering Approach to Monitoring Market Power in Restructured Markets for Electricity", the 24th Annual International Association for Energy Economics International Conference (Houston, TX, Apr. 2001).

[ix] B. C. Lesieutre, O. HyungSeon, R. J. Thomas, V. Donde, "Identification of Market Power in Large-Scale Electric Energy Markets," *System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on* , vol. 10, pp. 240b-240b, 04-07 Jan. 2006

[x] D. Cheverez-Gonzalez, C. L. DeMarco, "Admissible Locational Marginal Prices via Laplacian Structure in Network Constraints," *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp.125-133, Feb. 2009.

[xi] D. C. Lay, *Linear Algebra and its Applications*, 3rd ed., Addison Wesley, 2003.

[xii] IEEE 118-Bus Test System, Power System Test Case Archive, [Online]. Available: http://www.ee.washington.edu/research/pstca/

[xiii] Matlab Help Files. "pinv.m" Mathworks, Cambridge MA. 2002.

[xiv] Matlab Help Files. "cluster.m" Mathworks, Cambridge MA. 2002.

[xv] Matlab Help Files. "kmeans.m" Mathworks, Cambridge MA. 2002.