System identification using Nuclear Norm & Tabu Search optimization

To cite this article: Asif A. Ahmed et al 2018 IOP Conf. Ser.: Mater. Sci. Eng. 297 012041

View the article online for updates and enhancements.

You may also like
- Applicability of the minimum entropy generation method for optimizing thermodynamic cycles
  Xue-Tao Cheng, Xin-Gang Liang et al.
- Optimal Control of Li-Ion Battery Energy Storage System for Frequency Regulation
  Jihun Lim, Jin Hyeok Choi, Geon-pyo Lim et al.
- Profile-Decomposing Output of Multi-Channel Odor Sensor Array
  Xiaofan Zheng, Yoichi Tomiura, Kenshi Hayashi et al.
System identification using Nuclear Norm & Tabu Search optimization

Asif A. Ahmed, Marco P. Schoen*, and Ken W. Bosworth

Idaho State University, Department of Mechanical Engineering, 921 South 8th Avenue, Stop 8060, Pocatello, ID, 83209, USA

* Corresponding Author: schomarc@isu.edu

Abstract. In recent years, subspace System Identification (SI) algorithms have seen increased research, stemming from advanced minimization methods being applied to the Nuclear Norm (NN) approach in system identification. These minimization algorithms are based on hard computing methodologies. To the authors’ knowledge, as of now, there has been no work reported that utilizes soft computing algorithms to address the minimization problem within the nuclear norm SI framework. A linear, time-invariant, discrete time system is used in this work as the basic model for characterizing a dynamical system to be identified. The main objective is to extract a mathematical model from collected experimental input-output data. Hankel matrices are constructed from experimental data, and the extended observability matrix is employed to define an estimated output of the system. This estimated output and the actual – measured – output are utilized to construct a minimization problem. An embedded rank measure assures minimum state realization outcomes. Current NN-SI algorithms employ hard computing algorithms for minimization. In this work, we propose a simple Tabu Search (TS) algorithm for minimization. TS algorithm based SI is compared with the iterative Alternating Direction Method of Multipliers (ADMM) line search optimization based NN-SI. For comparison, several different benchmark system identification problems are solved by both approaches. Results show improved performance of the proposed SI-TS algorithm compared to the NN-SI ADMM algorithm.

1. Introduction

In recent years, nuclear norm system identification methods have found a renewed interest in the research community due to advances in convex optimization heuristics. Nuclear norm by definition is the sum of singular values of a matrix. In the context of this paper, it involves estimating the system matrices $A$, $B$, $C$ and $D$ of a state-space realization from given input-output data [1]. The method is aimed at utilizing a discrete-time state-space model, as well as the properties of observability and controllability matrices to develop corresponding Hankel and Toeplitz matrices, respectively [1]. Amongst the benefits of this method is that of minimizing the nuclear norm of a given matrix and thereby minimizing its rank without losing the linear matrix structure in the low-rank approximation [3, 4]. Tabu Search optimization involves an iterative search – similarly to the Local Search algorithm – without its shortcomings. A single agent explores the cost surface and moves from its current to a more minimal solution iteratively [5]. Searching is performed within tabu balls of fixed respective radii, to find a number of solutions, from which the solution of least cost is deemed as a probable globally optimal solution [5].
2. Mathematical System Structure

In order to perform minimization, the following deterministic state-space system is used as the basis model [2]:

\[ x(k+1) = Ax(k) + Bu(k) \]  \hspace{1cm} (1)

\[ y(k) = Cx(k) + Du(k) \]  \hspace{1cm} (2)

In the above two equations, \( x(k) \) represents state sequence, \( u(k) \) represents input, \( y(k) \) represents the output, and \( A, B, C, \) and \( D \) represent system matrices, while \( k \) is the time index. Hankel \( H_{u,s} \) and Toeplitz \( T_{u,s} \) matrices for the system are given in equations (3) and (4), respectively [2]:

\[
H_{u,s} = \begin{pmatrix} 0 & \cdots & 0 & D \\ 0 & \cdots & D & CB \\ \vdots & \ddots & \vdots & \vdots \\ D & \cdots & CA^{s-3}B & CA^{s-2}B \end{pmatrix}
\]  \hspace{1cm} (3)

\[
T_{u,s} = \begin{pmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ CA^{s-3}B & CA^{s-2}B & \cdots & D \end{pmatrix}
\]  \hspace{1cm} (4)

Observability matrix \( O_s \) and controllability matrix \( Q_s \) for the system of rank \( s \) are defined in equations (5) and (6), respectively [2]:

\[
Q_s = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{pmatrix}
\]  \hspace{1cm} (5)

\[
W_s = \begin{pmatrix} B & AB & A^2B & \cdots & A^{s-1}B \end{pmatrix}
\]  \hspace{1cm} (6)

3. Iterative Alternating Direction Method of Multipliers

Nuclear Norm minimization is applied through Iterative ADMM Method outlined in [2]. In this method, the cost function to be minimized is shown in equation (7):

\[
\underset{y}{\text{minimize}} \quad \frac{1}{2} \| H(y) U^+ \|_2 + \| y-b \|_2^2
\]  \hspace{1cm} (7)

In (7), \( \gamma \) is the regularization parameter, \( H(y) \) is the output Hankel matrix, and \( U^+ \) is the orthogonal complement of the input Hankel matrix, \( \| H(y) U^+ \|_2 \) is the nuclear norm of \( H(y) U^+ \), \( y \) is the output data matrix, while \( b \) is the measured data matrix, \( \| y-b \|_2 \) is the 2-norm of \( y-b \) [2]. The cost function as given in equation (7) is constrained as shown in (8) and (9), with \( Z = H(y)U^+ \) [2]:

\[
\underset{y, Z}{\text{minimize}} \quad \frac{1}{2} \| Z \|_2 + \| y-b \|_2^2
\]  \hspace{1cm} (8)

subject to \( H(y)U^+ Z = 0 \) \hspace{1cm} (9)

In the line search algorithm described in [1], augmented Lagrangian is used at each step of the ADMM iteration for obtaining the vector \( y \) that yields the minimum solution [2]. The augmented Lagrangian and its gradient are as given in equations (10) and (11) [2]. In these equations, \( y \) is the output data vector, \( L \) is the Lagrange multiplier, \( \tau \) represents trace of matrix, \( ^T \) represents the transpose of a matrix, \( \rho \) is a positive scalar, \( \varepsilon \) represents Frobenius norm, \( H^* \) represents adjoint matrix containing the sum of
entries on respective reverse diagonals of the given Hankel matrix, \( H \left( y^{k+l} \right) \) is the Hankel matrix of \( y^{k+l} \), \( k \) and \( l \) represent the outer loop iteration and inner loop iteration, respectively [2].

\[
L_{\rho}(y, Z, L) = y \left\| Z \right\| + \frac{1}{2} \left\| y - b \right\|^2 + tr \left( L^T \left( H(y) \right) U^* - Z \right) + \frac{P}{2} \left\| \left( H(y) \right) U^* - Z \right\|_F^2
\]

\[
\nabla L_{\rho}(y^{k+l}) = \rho H^\dagger \left( \left( H(y^{k+l}) \right) U^* - \left( Z^k - \left( \rho^{-1} \right) L^k \right) \right) \left( U^{-1} \right)^T + \left( y^{k+l} - b \right)
\]

\[
\text{The step size is computed at each iteration using values from present and past iterations employing the Barzilai Borwein method as given by equation (12) [2]. In this formula, } x' \text{ represents the current output, } x^{l-1} \text{ represents the output from the previous iteration, } \nabla f^l \text{ represents the current gradient, } \nabla f^{l-1} \text{ represents the gradient from the previous iteration, and } \left< x' - x^{l-1}, \nabla f^l - \nabla f^{l-1} \right> \text{ represents the dot product of } x' - x^{l-1} \text{ and } \nabla f^l - \nabla f^{l-1} [2]. \text{ Also, define}
\]

\[
\alpha^{(l)} = \frac{\left< x' - x^{l-1}, \nabla f^l - \nabla f^{l-1} \right>}{\left< \nabla f^l - \nabla f^{l-1}, \nabla f^l - \nabla f^{l-1} \right>}
\]

\[
\text{The use of the gradient descent method for performing a line search involves the use of matrix } D^l, \text{ which in this work is set to an identity matrix. The following are the steps for the Iterative ADMM method, [2]:}
\]

- **Preliminary Step**: Assign initial values for \( y, Z, \Lambda \), and assign \( \rho \) to be a positive scalar.

- **Step 1**: Iteratively calculate \( y \) using inner and outer line search routines as follows:
  \[ y^{(k+1,0)} = y^k \text{ and } y^{(k+1,-1)} = y^k \text{ (if applicable)} \]
  In the case that there is no fixed termination criteria, compute the following:
  \[ y^{(k+l, l+1)} = x'(k+l, l) - \alpha^{(l)} \left( \frac{D^l}{\rho} \right)^{-1} \nabla L_{\rho}(y^{k+l}) \]
  \[ \text{(13)} \]

- **Step 2**: Perform singular-value decomposition and increment \( Z \) in the outer line search as shown in (14) and (15). In these equations, \( W, S, V^T \) matrices are the result of performing singular value decomposition on the right-hand side of equation (14). \( I \) in equation (15) is the identity matrix.
  \[ W S V^T = H \left( y^{k+l} \right) U^* + \left( \rho^{-1} \right) L^k \]
  \[ \text{(14)} \]
  \[ Z^{k+l} = W \max \left\{ \frac{\mathcal{L} - 1}{\rho} I, 0 \right\} V^T \]
  \[ \text{(15)} \]

- **Step 3**: The Lagrange multiplier is incremented in the outer line search as follows:
  \[ L^{k+l} = L^k + \rho \left( H \left( y^{k+l} \right) U^* - Z^{k+l} \right) \]
  \[ \text{(16)} \]

4. **Tabu Search Method**

Tabu Search Optimization aims at finding the best possible solution through iteratively searching and moving from a current solution to a better solution. Through the search, paths taken are stored in the tabu list. With this list, directions for new moves are determined that would yield a better local optimum solution without cycling, in turn ultimately leading to the probable globally optimal solution. Tabu Search can be broken down into the following steps [5, 6]:

- **Step 1**: Choose a random initial solution \( S_0 \) from a search area having an \( n \)-dimensional radius \( R \), where \( n \) is the dimension of the problem. Solution \( S_0 \) will be the current local optimum solution.

- **Step 2**: By random choice, \( N \) new solutions are selected through moving the current solution around \( S_0 \) in the search space. Set \( S_0(r) \) comprises \( N \) solutions.

- **Step 3**: Utilize the objective function to evaluate the cost value of each member in \( S_0(r) \). From these, choose the best solution which is the one with the minimum cost.
Step 4: Compare the cost of the best solution obtained in step (3) to the cost of the solution obtained in step (1).

Step 5: If the cost of the best solution from step (3) is greater than or equal to the cost of the solution from step (1), then leave $S_0$ unchanged. On the other hand, if the cost of the solution from step (3) is less than the cost of the solution from step (1), store the best solution from step (3) in $S_0$. The solution with the minimum cost is the best solution.

Step 6: Store the best solution from step (5) in the tabu list.

Step 7: Repeat the process from step (2) through step (6) until the maximum number of iteration is reached, or if the global minimum is obtained.

The TS algorithm also incorporates a diversification and intensification function by incorporating a promising list – similar to the Tabu list – storing a set number of recent solutions. The Tabu list is used to avoid cycling and hence drive the search to remote places in the search field. The promising list is utilized in a second search, where individual items in the list are further explored. Each item on either list comprises an n-dimensional sphere with a constant radius.

5. Eigen System Realization

Eigensystem Realization Algorithm (ERA) is utilized in order to identify the system using estimated output from the two estimation techniques discussed in Sections III and IV [7]. This algorithm aims to use the discrete impulse response $h$ for constructing a square block Hankel matrix $H(0)$ as shown in (17) [7, 8]. A second square block Hankel matrix $H(1)$ is obtained by shifting every column in the original block Hankel matrix $H(0)$ by one unit to the left. Singular-value decomposition is performed on $H(0)$, the result of which are the matrices $P_n$, $D_n$, and $Q_n^T$. From the results of singular-value decomposition, a matrix $D_m = D_k(1:n,1:n)$ is obtained, where $n$ represents the order of the system. Observability and Controllability matrices are calculated by using (18) and (19) with respect to the order of the system $n$. System matrices $A$, $B$, $C$, and $D$ are calculated through the use of (20) through (23). Once system matrices are known, it is possible to calculate a discrete-time or a continuous-time transfer function [7].

$$H(0) = \begin{pmatrix} h(1) & h(2) & \cdots & h(n) \\ h(2) & h(3) & \cdots & h(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ h(n) & h(n+1) & \cdots & h(2n-1) \end{pmatrix}$$  \hspace{1cm} (17)

$$W_o = P_n(:,1:n) \ast (D_{n,n}^{0.5})$$  \hspace{1cm} (18)

$$C_o = (D_{m,n}^{0.5}) \ast (Q_n^T(:,1:n))^T$$  \hspace{1cm} (19)

$$A = (D_{m,n}^{0.5}) \ast (P_n(:,1:n))^T \ast H(1) \ast (Q_n(:,1:n))^T \ast (D_{m,n}^{0.5})$$  \hspace{1cm} (20)

$$B = C_0 \ast (1,:)$$  \hspace{1cm} (21)

$$C = W_o \ast (1,:)$$  \hspace{1cm} (22)

$$D = h(1,:)$$  \hspace{1cm} (23)

6. Benchmark Simulated Systems

A system associated with a benchmark research problem that was chosen for applying and comparing the two system identification techniques. They yielded estimation results as shown in (24), (25), (26), table 1, figure 1, and figure 2 [9]. Equation (24) represents discrete-time transfer function of the true system, (25) is the estimated discrete-time transfer function from the Iterative ADMM method, and (26) is the estimated discrete-time transfer function using the Tabu Search method. Table 1 represents key
system identification metrics; and simulation plots for the Iterative ADMM method and the Tabu Search method are given in figure 1 and figure 2, respectively.

**System 1 (P. Dreesen, M. Ishteva, and J. Schoukens) [9]:**

\[
G_{\text{true}}(z) = \frac{8.4853 z^2 + 25.4558 z + 16.9706}{z^3 - 1.5z^2 + z - 0.25}
\]  \hspace{1cm} (24)

\[
G_{\text{ItADMM}}(z) = \frac{0.0004246 z^2 + 14.16 z + 13}{z^3 - 1.682 z^2 + 1.19 z - 0.3228}
\]  \hspace{1cm} (25)

\[
G_{\text{TabuSearch}}(z) = \frac{2.203e-05 z^2 + 38.14 z + 7.02}{z^3 - 1.548 z^2 + 1.049 z - 0.2695}
\]  \hspace{1cm} (26)

**Table 1. Estimation & Simulation Results.**

| Metric          | Iterative ADMM | Tabu Search | True System |
|-----------------|----------------|-------------|-------------|
| Iterations      | 201            | 100         | -           |
| CPU Time        | 0.8775         | 128.7338    | -           |
| Gamma Value     | 427.3409       | 427.3409    | -           |
| Error Identification | Percent Error=0% | -          | -           |
| A Eigenvalue 1  | 0.5445 + 0.4982i | 0.5096 + 0.4997i | 0.5000 + 0.5000i |
| A Eigenvalue 2  | 0.5445 – 0.4982i | 0.5096 - 0.4997i | 0.5000 - 0.5000i |
| A Eigenvalue 3  | 0.5927 + 0.0000i | 0.5289 + 0.0000i | 0.5000 + 0.0000i |

**Figure 1. Iterative ADMM Estimation & True System Compared**
Eleven additional benchmark systems are analyzed in a similar manner as System 1 from above [9 – 13]. Respective system identification results are presented as follows in (27) – (59), and tables 2 – 12 [9 – 13]:

System 2 (P. Dreesen, M. Ishteva, and J. Schoukens) [9]:

\[
G_{true}(z) = \frac{0.7071z + 4.2426}{z^2 + 0.750z + 0.125} 
\]

(27)

\[
G_{ItADMM}(z) = \frac{1.717}{z^2 + 0.7677z + 6.903e-07} 
\]

(28)

\[
G_{TabuSearch}(z) = \frac{3.748}{z^2 + 0.7714z + 0.1436} 
\]

(29)

Table 2. Estimation & Simulation Results.

| METRIC              | Iterative ADMM | Tabu Search | True System |
|---------------------|----------------|-------------|-------------|
| Iterations          | 151            | 100         | -           |
| CPU Time            | 0.3828         | 136.4569    | -           |
| Gamma Value         | 25             | 25          | -           |
| Error Identification| -              | -           | -           |
| Error=58.0721%      | -              | -           |
| A Eigenvalue 1      | -0.7677        | -0.4576     | -0.5        |
| A Eigenvalue 2      | 0              | -0.3138     | -0.25       |

System 3 (A. Padilla, H. Garnier, and M. Gilson) [10]:

\[
G_{true}(z) = \frac{-0.7757z^2 - 1.367z^3 + 1.624z + 0.6028}{z^2 - 1.198z^3 + 0.3238z^4 - 0.6383z^5 + 0.6015} 
\]

(30)

\[
G_{ItADMM}(z) = \frac{-0.000235z^2 - 1.821z^3 + 2.043z - 0.1704}{z^4 - 1.493z^5 + 0.7493z^6 - 0.6952z^7 + 0.5122} 
\]

(31)
\[ G_{\text{TabuSearch}}(z) = \frac{-0.0001543z^4 - 2.381z^2 + 1.817z + 0.7007}{z^4 - 1.148z^3 + 0.1312z^2 - 0.4208z + 0.5251} \]  \hspace{1cm} (32)

**Table 3. Estimation & Simulation Results.**

| METRIC       | System Identification Results | Iterative ADMM | Tabu Search | True System |
|--------------|--------------------------------|----------------|-------------|-------------|
| Iterations   | 179                            | 100            | -           | -           |
| CPU Time     | 1.0068                         | 173.6016       | -           | -           |
| Gamma Value  | 8.9437                         | 8.9437         | -           | -           |
| Error        | Identification                 | Percent Error=0%| -           | -           |
| A Eigenvalue 1 | 0.9398 + 0.1843i            | 0.9363 + 0.1839i         | 0.9328 + 0.1834i     |
| A Eigenvalue 2 | 0.9398 - 0.1843i            | 0.9363 - 0.1839i        | 0.9328 - 0.1834i     |
| A Eigenvalue 3 | -0.1932 + 0.7219i         | -0.3621 + 0.6676i       | -0.3338 + 0.7444i    |
| A Eigenvalue 4 | -0.1932 - 0.7219i         | -0.3621 - 0.6676i       | -0.3338 - 0.7444i    |

System 4 (A. Padilla, H. Garnier, and M. Gilson) [10]:

\[ G_{\text{true}}(z) = \frac{-0.7526z^4 - 1.329z^2 + 1.58z + 0.5876}{z^4 - 1.194z^3 + 0.3258z^2 - 0.6458z + 0.6059} \]  \hspace{1cm} (33)

\[ G_{\text{ItADMM}}(z) = \frac{-3.801e-06z^4 - 1.63z^2 + 0.6493z - 0.9706}{z^4 - 0.08649z^3 + 0.3589z^2 - 0.2013z + 0.2105} \]  \hspace{1cm} (34)

\[ G_{\text{TabuSearch}}(z) = \frac{-0.000158z^4 - 2.31z^2 + 1.77z + 0.6786}{z^4 - 1.145z^3 + 0.1359z^2 - 0.4308z + 0.5303} \]  \hspace{1cm} (35)

**Table 4. Estimation & Simulation Results.**

| METRIC       | System Identification Results | Iterative ADMM | Tabu Search | True System |
|--------------|--------------------------------|----------------|-------------|-------------|
| Iterations   | 187                            | 100            | -           | -           |
| CPU Time     | 0.9257                         | 135.1128       | -           | -           |
| Gamma Value  | 17.6022                        | 17.6022        | -           | -           |
| Error        | Identification                 | Percent Error=0%| -           | -           |
| A Eigenvalue 1 | -0.4482 + 0.000i            | 0.9354 + 0.1865i         | 0.9319 + 0.1861i     |
| A Eigenvalue 2 | -0.0708 - 0.8303i           | 0.9354 - 0.1865i         | 0.9319 - 0.1861i     |
| A Eigenvalue 3 | -0.0708 - 0.8303i           | -0.3626 + 0.6719i        | -0.3349 + 0.7476i    |
| A Eigenvalue 4 | 0.6764 + 0.0000i            | -0.3626 - 0.6719i        | -0.3349 - 0.7476i    |

System 5 (A. Padilla, H. Garnier, and M. Gilson) [10]:

\[ G_{\text{true}}(z) = \frac{0.05257z + 0.04601}{z^2 - 1.637z + 0.6703} \]  \hspace{1cm} (36)

\[ G_{\text{ItADMM}}(z) = \frac{0.04767z^2 - 4.084e-05z + 5.438e-05}{z^2 - 1.732z + 0.7571} \]  \hspace{1cm} (37)

\[ G_{\text{TabuSearch}}(z) = \frac{0.08385z + 2.908e-05}{z^2 - 1.69z + 0.7183} \]  \hspace{1cm} (38)
Table 5. Estimation & Simulation Results.

| METRIC                   | System Identification Results                      | Iterative ADMM | Tabu Search | True System |
|--------------------------|------------------------------------------------------|----------------|-------------|-------------|
| Iterations               |                                                      | 149            | 100         | -           |
| CPU Time                 |                                                      | 0.7612         | 135.1274    | -           |
| Gamma Value              |                                                      | 0.9346         | 0.9346      | -           |
| Error                    | Identification                                       |                |             |             |
| Error                    | Percent Error=0%                                      |                |             |             |
|                          | Error=43.3266%                                        |                |             |             |
| A Eigenvalue 1           |                                                      | 0.8662 + 0.0820i | 0.8449 + 0.0665i | 0.8185 + 0.0189i |
| A Eigenvalue 2           |                                                      | 0.8662 – 0.0820i | 0.8449 – 0.0665i | -0.8185 – 0.0189i |

System 6 (A. Y. K. Yong, A. H. Tan, and C. L. Cham) [11]:

\[ G_{true}(z) = \frac{z^2 + 0.5}{z^2 - 1.5z + 0.7} \]  \( (39) \)

\[ G_{ItADMM}(z) = \frac{-2.751e-05z + 0.1489}{z^2 - 1.58z + 0.6947} \]  \( (40) \)

\[ G_{TabuSearch}(z) = \frac{-1.384e-05z + 1.66}{z^2 - 1.411z + 0.6503} \]  \( (41) \)

Table 6. Estimation & Simulation Results.

| METRIC                   | System Identification Results                      | Iterative ADMM | Tabu Search | True System |
|--------------------------|------------------------------------------------------|----------------|-------------|-------------|
| Iterations               |                                                      | 153            | 100         | -           |
| CPU Time                 |                                                      | 0.7766         | 136.7983    | -           |
| Gamma Value              |                                                      | 75             | 75          | -           |
| Error                    | Identification                                       |                |             |             |
| Error                    | Percent Error=0%                                      |                |             |             |
|                          | Error=89.4514%                                        |                |             |             |
| A Eigenvalue 1           |                                                      | 0.7898 + 0.2662i | 0.7054 + 0.3907i | 0.7500 + 0.3708i |
| A Eigenvalue 2           |                                                      | 0.7898 - 0.2662i | 0.7054 - 0.3907i | 0.7500 - 0.3708i |

System 7 (A. Y. K. Yong, A. H. Tan, and C. L. Cham) [11]:

\[ G_{true}(z) = \frac{0.0644z^2 + 0.0249z - 0.0045}{z^2 - 2.451z + 2.129 - 0.6666} \]  \( (42) \)

\[ G_{ItADMM}(z) = \frac{-1.498e-07z^2 + 0.004713z + 0.001619}{z^2 - 2.517z + 2.202 - 0.6771} \]  \( (43) \)

\[ G_{TabuSearch}(z) = \frac{1.727e-06z^2 + 0.1604z - 0.08228}{z^2 - 2.432z^2 + 2.0772 - 0.6368} \]  \( (44) \)

Table 7. Estimation & Simulation Results.

| METRIC                   | System Identification Results                      | Iterative ADMM | Tabu Search | True System |
|--------------------------|------------------------------------------------------|----------------|-------------|-------------|
| Iterations               |                                                      | 125            | 100         | -           |
| CPU Time                 |                                                      | 0.5013         | 142.2242    | -           |
| Gamma Value              |                                                      | 25             | 25          | -           |
| Error                    | Identification                                       |                |             |             |
| Error                    | Percent Error=0%                                      |                |             |             |
|                          | Error=90.3023%                                        |                |             |             |
| A Eigenvalue 1           |                                                      | 0.9461 + 0.0000i | 0.9438 + 0.0000i | 0.9427 + 0.0000i |
| A Eigenvalue 2           |                                                      | 0.7854 + 0.3142i | 0.7431 + 0.3500i | 0.7542 + 0.3720i |
| A Eigenvalue 3           |                                                      | 0.7854 – 0.3142i | 0.7431 - 0.3500i | 0.7542 – 0.3720i |
System 8 (A. Y. K. Yong, A. H. Tan, and C. L. Cham) [11]:

\[
G_{true}(z) = \frac{0.4479z^{-3} + 0.1254z^{-2} - 0.0076}{z^{-3} - 2.114z^{-2} + 1.622z^{-1} - 0.4319}
\]

\[
G_{ItADMM}(z) = \frac{8.524e-06z^{-3} + 0.04927z^{-2} + 0.009867}{z^{-3} - 2.474z^{-2} + 2.156z^{-1} - 0.6554}
\]

\[
G_{TabuSearch}(z) = \frac{-2.562e05z^{-3} + 0.9902z^{-2} - 0.7127}{z^{-3} - 2.3z^{-2} + 1.888z^{-1} - 0.5486}
\]

Table 8. Estimation & Simulation Results.

| METRIC            | System Identification Results | Iterative ADMM | Tabu Search | True System |
|-------------------|-------------------------------|----------------|-------------|-------------|
| Iterations        | 159                           | 100            | -           | -           |
| CPU Time          | 0.7058                        | 134.9639       | -           | -           |
| Gamma Value       | 46.4                          | 46.4           | -           | -           |
| Error Identification | Error=87.1189%            | Percent Error=0% | -           | -           |
| A Eigenvalue 1    | 0.8275 + 0.3400i              | 0.7574 + 0.3540i | 0.7452 + 0.3705i |
| A Eigenvalue 2    | 0.8275 - 0.3400i              | 0.7574 - 0.3540i | 0.7452 - 0.3705i |
| A Eigenvalue 3    | 0.8189 – 0.0000i              | 0.7849 + 0.0000i | 0.6236 + 0.0000i |

System 9 (H. Ase, and T. Katayama) [12]:

\[
G_{true}(z) = \frac{-0.03z + 0.1}{z^{-2} - 1.8z + 0.85}
\]

\[
G_{ItADMM}(z) = \frac{-3.009e-06z^{-2} + 0.02164}{z^{-2} - 1.791z + 0.8437}
\]

\[
G_{TabuSearch}(z) = \frac{-1.278e-05z^{-2} + 0.06723}{z^{-2} - 1.811z + 0.8585}
\]

Table 9. Estimation & Simulation Results.

| METRIC            | System Identification Results | Iterative ADMM | Tabu Search | True System |
|-------------------|-------------------------------|----------------|-------------|-------------|
| Iterations        | 152                           | 100            | -           | -           |
| CPU Time          | 0.7088                        | 145.6996       | -           | -           |
| Gamma Value       | 7.02                          | 7.02           | -           | -           |
| Error Identification | Error=71.2422%            | Percent Error=0% | -           | -           |
| A Eigenvalue 1    | 0.8956 + 0.2038i              | 0.9056 + 0.1959i | 0.9000 + 0.2000i |
| A Eigenvalue 2    | 0.8956 - 0.2038i              | 0.9056 - 0.1959i | 0.9000 - 0.2000i |

System 10 (H. Ase, and T. Katayama) [12]:

\[
G_{true}(z) = \frac{0.06875z^{-3} + 0.13775}{z^{-3} - 2.34z^{-2} + 2.53z^{-1} + 1.24z + 0.368}
\]

\[
G_{ItADMM}(z) = \frac{-2.547e-06z^{-3} + 0.01145z^{-2} + 0.09295z^{-1} - 0.1272z + 0.1945}{z^{-3} - 2.11z^{-2} + 2.747z^{-1} - 2.287z + 1.49z - 0.4002}
\]
$$G_{\text{TabuSearch}}(z) = \frac{2.052e-09z^4 - 1.44e-09z^3}{z^4 - 2.34z^3 + 3.08z^2} \frac{+4.646e-09z^2 + 0.06875z + 0.1378}{z^2 - 2.53z^1 + 1.24z - 0.368}$$  \hspace{1cm} (53)

### Table 10. Estimation & Simulation Results.

| METRIC                      | System Identification Results |
|-----------------------------|-------------------------------|
|                             | Iterative ADMM | Tabu Search | True System |
| Iterations                  | 144             | 100         | -           |
| CPU Time                    | 0.7452          | 133.8792    | -           |
| Gamma Value                 | 4.4725          | 4.4725      | -           |
| Error Identification        | Identification  | Percent     | -           |
| Error=51.4888%              |                  | Error=0%    | -           |
| A Eigenvalue 1              | 0.9131 + 0.0000i| 0.9012 + 0.0000i| 0.9012 + 0.0000i|
| A Eigenvalue 2              | 0.2068 + 0.7576i| 0.3037 + 0.7491i| 0.3037 + 0.7491i|
| A Eigenvalue 3              | 0.2068 - 0.7576i| 0.3037 - 0.7491i| 0.3037 - 0.7491i|
| A Eigenvalue 4              | 0.3926 + 0.7460i| 0.4157 + 0.6724i| 0.4157 + 0.6724i|
| A Eigenvalue 5              | 0.3926 - 0.7460i| 0.4157 - 0.6724i| 0.4157 - 0.6724i|

System 11 (M. Galrinho, C. R. Rojas, and H. Hjalmarsson) [13]:

$$G_{\text{true}}(z) = \frac{1}{z^2 - 0.2z + 0.5}$$ \hspace{1cm} (54)

$$G_{\text{ItADMM}}(z) = \frac{-1.43e-07z + 0.6673}{z^2 - 0.5887z + 0.4752}$$ \hspace{1cm} (55)

$$G_{\text{TabuSearch}}(z) = \frac{-1.623e-08z + 1}{z^2 - 0.2z + 0.5}$$ \hspace{1cm} (56)

### Table 11. Estimation & Simulation Results.

| METRIC                      | System Identification Results |
|-----------------------------|-------------------------------|
|                             | Iterative ADMM | Tabu Search | True System |
| Iterations                  | 175             | 100         | -           |
| CPU Time                    | 0.7678          | 159.8861    | -           |
| Gamma Value                 | 1.677           | 1.677       | -           |
| Error Identification        | Identification  | Percent     | -           |
| Error=67.9169%              |                  | Error=0%    | -           |
| A Eigenvalue 1              | 0.2944 + 0.6233i| 0.1000 + 0.7000i| 0.1000 + 0.7000i|
| A Eigenvalue 2              | 0.2944 - 0.6233i| 0.1000 - 0.7000i| 0.1000 - 0.7000i|

System 12 (A. Padilla, H. Garnier, and M. Gilson) [10]:

$$G_{\text{true}}(z) = \frac{-0.769lz^4 - 1.359z^2 + 1.616z + 0.6008}{z^4 - 1.2z^3 + 0.3309z^2 + 0.6484z + 0.6065}$$ \hspace{1cm} (57)

$$G_{\text{ItADMM}}(z) = \frac{-3.257e-06z^3 - 0.3884z^2 + 0.3775z + 9.63e-06}{z^3 - 3.022z^2 + 3.712z + 2.255z + 0.5847}$$ \hspace{1cm} (58)

$$G_{\text{TabuSearch}}(z) = \frac{-0.0001684z^3 - 2.366z^2 + 1.813z + 0.6935}{z^3 - 1.151z^2 + 0.1404z^2 - 0.4324z + 0.5504}$$ \hspace{1cm} (59)
Table 12. Estimation & Simulation Results.

| METRIC              | System Identification Results | Iterative ADMM | Tabu Search | True System |
|---------------------|-------------------------------|----------------|-------------|-------------|
| Iterations          | 182                           | 100            | -           | -           |
| CPU Time            | 0.9399                        | 134.3384       | -           | -           |
| Gamma Value         | 23.351                        | 23.351         | -           | -           |
| Error Identification| 72.2935%                      |                |             |             |
| Error               | Percent Error=0%              |                |             |             |
| A Eigenvalue 1      | 0.5818 + 0.5584i              | -0.3611 + 0.6721i | -0.3333 + 0.7479i |
| A Eigenvalue 2      | 0.5818 – 0.5584i              | -0.3611 – 0.6721i | -0.3333 – 0.7479i |
| A Eigenvalue 3      | 0.9292 + 0.1889i              | 0.9367 + 0.1838i | 0.9333 + 0.1834i |
| A Eigenvalue 4      | 0.9292 - 0.1889i              | 0.9367 - 0.1838i | 0.9333 - 0.1834i |

7. Optimization Methods Compared

System identification techniques of Iterative ADMM presented in Section III, and Tabu Search presented in Section IV were applied, tested, simulated, analyzed, and compared for twelve systems from benchmark research problems, the results for which are presented in Section VI [9 – 13]. Key findings from the results are summarized as follows:

- Number of Iterations: Iterative ADMM uses more iterations than Tabu Search, due to Tabu Search method arriving at an optimal solution with minimum cost, and Iterative ADMM method not being able to arrive at fully optimal solution, thereby requiring more iterations for its computation.

- Computation Time: Iterative ADMM is faster, using less computation time than Tabu Search.

- Error: Iterative ADMM has certain spikes of very high percent error caused due to a non-optimal value of gamma being used. This could be improved by testing the Iterative ADMM method for a broader range of regularization parameter gamma, to arrive at the value that would yield optimal results [2]. Tabu Search, being fully optimized, is able to give zero percent error for all systems. Identification error for Iterative ADMM gives a measure of how close the estimation is to the actual output, and is calculated using (60) as follows:

\[ e_I = 100 \times \frac{\| y - \hat{y} \|_2}{\| y \|_2} \]  

- Eigenvalues of A Matrix: The extracted eigenvalues of the system matrix \( A \) of Iterative ADMM estimation are close to the true system A matrix eigenvalues for some systems, but not for all systems. That is caused by a non-optimal value of gamma being used, thereby decreasing the accuracy of discrete-time impulse response used for eigensystem realization to obtain system matrices. Eigenvalues of the A matrix of Tabu Search estimation are closer – but not exactly equal for all systems – to those of the true system. Slight inaccuracy in this case is caused by the chosen sample time and the number of samples used for obtaining discrete-time impulse response and eigensystem realization for obtaining system matrices.

8. Conclusions

Applications of two system identification techniques of Nuclear Norm and Tabu Search were investigated in this research. Key metrics show that Tabu Search minimization method yields optimal estimation results, while Iterative ADMM minimization method is able to perform system identification which matches the behavior of the true system to some degree of accuracy but not optimally. A perfect match in a stochastic system is usually a suspect; however, considering the finite data utilized in the algorithm, Tabu Search will find a model that exactly fits the finite set of input/output data with a given model. It does not mean that, for example, if a different realization of the input/output data is supplied,
this "perfect" model will have 0% error, but should yield a reasonable result.

Future work in this research could involve further enhancement of solution for Iterative ADMM to perform estimation for a broad range of values for regularization parameter gamma while observing the identification error, in order to search for a value for gamma that could yield closer to optimum solution for a given system.

In this research, the gradient descent method, which uses $D^I$ as an identity matrix is chosen in the development of the Iterative ADMM method involving inner and outer line search [2]. Other methods that could be investigated for development of Iterative ADMM solution include Newton and Quasi-Newton methods [2]. Newton’s method involves using $D^I$ as Hessian $\nabla^2 L_p(y)$ [2]. Quasi-Newton methods use $D^I$ which is an approximation to the Hessian, evaluated from information about gradient computed at present and past iterates [2]. For the Quasi-Newton approach, the BFGS formula for approximating inverse of Hessian is used [2]; the naming of which stems from the inventors of the formula: Broyden, Fletcher, Goldfarb, and Shanno [2]. Hessian inverse $H^I = (D^I)^{-1}$ is approximated from (61) - (63) [2]. In these equations, $\Delta y^{k+1,l-1}$ is the change in $y$ computed by evaluating difference between current inner line-search $y$ and the $y$ from previous inner line-search iteration [2]. $\Delta y^{k+1,l-1}$ is the change in $y$, calculated by subtracting gradient of Lagrangian at current inner line-search iteration from gradient of Lagrangian of previous inner line-search iteration [2]. $I$ is the identity matrix, $^T$ means transpose of matrix, $H^I$ is the Hessian inverse at the current inner line-search iteration, and $H^{-1}$ is the Hessian inverse from the previous inner line-search iteration [2].

$$
D y^{k+1,l-1} = y^{k+1,l} - y^{k+1,l-1} \quad (61)
$$

$$
D g^{k+1,l-1} = \nabla L_p(y^{k+1,l}) - \nabla L_p(y^{k+1,l-1}) \quad (62)
$$

$$
\hat{\alpha}^l = (I - \frac{D y^{k+1,l-1} y^{k+1,l-1}}{(D y)^T (D y)})^{-1} \quad (63)
$$

This research utilizes the Barzilai – Borwein method for evaluation of step size as represented in (12) [2]. Two other methods that could be researched that involve different methods for calculation of step-size are Backtracking line search method and Wolfe line search [2]. Backtracking line search is based on the requirement that step-size $\alpha^l$ meets the Armijo condition, also known as the sufficient decrease condition in (64) [2]. In this equation, $x^l$ is the value of the output $y$ at current inner line-search iteration, $\alpha^l$ is the step-size at current inner line-search iteration, $d^l$ is the descent direction, $c_1$ is a constant as $c_1 \in (0,1)$, $\nabla f^l$ is the gradient evaluated at current inner line-search iteration, and $^T$ means transpose of matrix [2].

$$
f(x^l + \alpha^l d^l) \leq f(x^l) + \alpha^l (\nabla f^l)^T d^l \quad (64)
$$

Wolfe line search is based on the requirement that step-size $\alpha^l$ meets the curvature condition and the Armijo condition [2]. Curvature condition is given in (65) [2].

$$
\nabla^2 f(x^l + \alpha^l d^l)^T d^l \geq c_2 (\nabla f^l)^T d^l \quad (65)
$$
Acknowledgment

This work is partially supported by the National Science Foundation under Grant No. NSF 1403688. The content is solely the author’s responsibility. The support is greatly appreciated.

References

[1] Sadigh D, Ohlsson H, Sastry S S and Seshia S A 2014 Proc. 19th IFAC World Congress (Cape Town, South Africa) pp 9510-9515
[2] Deshmanc H 2014 System identification via nuclear norm regularization M. S. thesis (Univ. of Minnesota, Minneapolis, MN)
[3] Fazel M 2002 Matrix rank minimization with applications Ph.D. dissertation (Stanford Univ., Stanford, CA)
[4] Liu Z, Hansson A and Vandenberghe L 2013 Systems & Control Lett 62(8) pp 605-612
[5] Areerak K N, Srikaew A, Sujitjorn S and Totarong P 2002 Proc. Intern. Conf. on Indus. Tech., IEEE ICIT, (Bangkok, Thailand) pp 915-920
[6] Sujitjorn S, Kluabwang J, Puangdownreong D and Sarasiri N 2010 ECTI Trans. on Elec. Eng., Electronics, and Communications. 8(1) pp 1-10
[7] Juang J.N and Suzuki H 1988 ASME J. of Vibr. Acous, Stress, and Reliab. in Design 110(1) pp 24-29
[8] Xu W and Qiao S 2006 Proc. 2nd Inter. Conf. on Struct. Matr. (Kowloon Tong, Hong Kong) pp 550-563
[9] Dreesen P, Ishteva M and Schoukens J 2015 Proc. 17th IFAC Symp. on System Identification (Beijing, China) pp 951-956
[10] Padilla A, Garnier H and Gilson M 2015 Proc. 17th IFAC Symp. on System Identification (Beijing, China) pp 757-762
[11] Yong A Y K, Tan A H and Cham C L 2015 Proc. 17th IFAC Symp. on System Identification (Beijing, China) pp 939-944
[12] Ase H and Katayama T 2015 Proc. 17th IFAC Symp. on System Identification (Beijing, China) pp 638-643
[13] Galrinho M, Rojas C R and Hjalmarsson H 2015 Proc. 17th IFAC Symp. on System Identification (Beijing, China) pp 98-103
[14] Liu Z and Vandenberghe L 2010 SIAM J. on Matrix Ana. and Applic. 31(3) pp 1235 – 1256
[15] Hansson A, Liu Z and Vandenberghe L 2012 Proc. 2012 IEEE 51st IEEE Conf. on Decision and Control (Maui, Hawaii) pp 3439-3444