# 4. COMPUTER IMPLEMENTATION

The algorithm described in this paper was developed in the C++ language for its portability and object-orientated features. The Matlab software package was then used to graphically display the workspace. The entire implementation consists of five steps:

1. Define a function, called TestPose, which returns the evaluation result, either true or false.
2. Implement a search routine to determine the approximate workspace center.
3. Implement a criterion to determine the workspace boundary.
4. Evaluate the workspace volume.
5. Graphically display the workspace.

## 4.1 TestPose Function

The *TestPose* function is defined as a Boolean function. The function tests an arbitrary position and orientation of the platform, or a pose of the platform, for constraint violation.

$$\textit{Boolean} = \textbf{\textit{TestPose}}(\, T_x, T_y, T_z, a, b, g \,)$$

The function first computes inverse kinematics and then verifies whether the constraint sets I, II, III, & IV are satisfied. If all of the constraints are satisfied, then the function returns a value of true. If any constraint is not satisfied, it returns a value of false. In other words, if the *TestPose* function is true, the given position and orientation of the platform is reachable by the Hexapod.

## 4.2 Workspace Search Algorithm

The workspace algorithm conducts a search to determine the workspace of the Hexapod machine tool with a given platform orientation, say $\mathbf{O}_1 = \{\alpha_1, \beta_1, \gamma_1\}$. The first step in the workspace search is to determine the approximate workspace center, a general requirement to initialize the workspace search. This is done by searching a three dimensional cubic containing positions of the platform within the work volume of the machine. For the position at the center of the cubic, the *TestPose* function checks for any constraint violations. Using the results obtained from testing these central positions, the approximate center of the workspace can be determined (Figure 5). This center will then serve as the origin of a spherical coordinate system used to define the actual workspace boundary. The following procedure outlines the major steps involved in the search process.

As illustrated in Figure 5, a workspace search vector, **V**, expressed in the spherical coordinates, **V** = {ρ, θ, ϕ}, is introduced. The vector, **V**, is considered to be at or near the workspace boundary if the following workspace boundary condition is satisfied:

*Workspace Boundary Condition*

***TestPose***( $\rho, \theta, \phi, \mathbf{O}_1$ ) = *True*

***TestPose***( $\rho+\varepsilon, \theta, \phi, \mathbf{O}_1$ ) = *False*

The process of searching the entire space is accomplished by rotating the search vector in discrete intervals Δθ and Δϕ. The algorithm is presented by the following pseudocode.

*Spherical Search Algorithm*

```
$\theta = 0$ degrees
loop while $\theta < 360$ degrees
    $\phi = -90.0$ degrees
    loop while $\phi < 90$ degrees
        find the radius that satisfies the Workspace Boundary Condition
        $\phi = \phi + \Delta\phi$
    end loop
```

# Dynamic Variation of the Workspace of an Octahedral Hexapod Machine Tool During Machining

*by J.P. Conti, C.M. Clinton,*
*G. Zhang, A.J. Wavering*

T.R. 97-28

ISR
INSTITUTE FOR SYSTEMS RESEARCH

*Sponsored by*
*the National Science Foundation*
*Engineering Research Center Program,*
*the University of Maryland,*
*Harvard University,*
*and Industry*

# DYNAMIC VARIATION OF THE WORKSPACE OF AN OCTAHEDRAL HEXAPOD MACHINE TOOL DURING MACHINING

**Joseph P. Conti, Charles M. Clinton, and Guangming Zhang**
Department of Mechanical Engineering & Institute for Systems Research
University of Maryland,
College Park, MD 20742, USA

**Albert J. Wavering**
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899, USA

## ABSTRACT

A method is presented to evaluate the workspace variation of a Stewart platform based machine tool. Four sets of constraints, covering strut lengths, platform spherical joint angles, base spherical joint angles, and strut collisions, are formulated using inverse kinematics. Recognizing the need for varying the platform orientation during machining, an algorithm to efficiently calculate the workspace is developed. Computer implementation provides a powerful tool to study the dynamic variation of the workspace as the spindle platform rotates away from the horizontal orientation. A case study is presented on the workspace variation of an Ingersoll Octahedral Hexapod[1] machine tool during machining. The results demonstrate the shift in size and location of the workspace as the platform orientation changes. Guidelines for NC coding are suggested to maximize the versatility of Stewart platform based machine tools, while avoiding the violation of constraint conditions on the workspace.

[1] Certain commercial equipment, instruments, or materials are identified in this paper to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

# 1. INTRODUCTION

With the rapid advancement of computer technology, information-based manufacturing is revolutionizing the process of product development. Improving product quality, reducing product cost, and shortening the product development cycle are becoming critical for companies to stay in competition. Thus, there is an increasing need to control machining accuracy while performing heavy-duty and high speed machining. Such pressing needs pose new challenges to the machine tool research community and call for redefining the design of machine tools. Parallel manipulators offer unique solutions to meet these challenges.

Among parallel mechanisms, the six degree-of-freedom Stewart platform has recently been used in a number of new machine tool designs (Stewart 1965). The Stewart platform is characterized by high force-torque capacity, high structural rigidity, and high accuracy in motion and position control. Several prototypes of Stewart-platform-based machine tools, typically called Hexapods, have been produced by Giddings & Lewis, Hexel Corporation, and Ingersoll.

Although the versatility of Hexapod machine tools has been recognized, their acceptance by industry as manufacturing equipment has been slow. A major obstacle to this is their current high cost and unproven performance in a production environment. Furthermore, the fact that their workspace is not constant, but varies with platform orientation, means that in certain cases their use is not as straightforward as traditional machine tools.

During the past decade, studies of the workspace of Stewart platforms have been performed by several researchers (Gosselin 1990; Kumar 1992; Ji 1994; Luh et al. 1996). Most of these studies focused on the workspace restrictions due to limited strut lengths and/or joint angles. Few of them have integrated other important constraints, such as strut collisions, into workspace modeling. In addition, little work has been done on determining dynamic variation of the workspace caused by changes in platform orientation. Such a study is essential since the variation of the platform orientation is an integral part of Hexapod machining operations.

Recognizing the need to provide information for the planning of machining operations, this paper presents a new study to characterize the workspace of Hexapod-type machine tools. In the analytical aspect, the paper presents the formulation of four constraint sets, including the consideration of strut collision, using the method of inverse kinematics. In the experimental aspect, numerical values of the limiting strut lengths and joint angles are taken from an Ingersoll-made Octahedral-Hexapod machine tool which is currently located at the National Institute of Standards and Technology. Visualization of the workspace in 3-dimensional and 2-dimensional space is presented to depict the dynamic nature of the workspace as the platform changes orientation. To demonstrate the efficiency and robustness of the proposed approach, the workspaces obtained from the early work done by Gossselin (1990) and Luh et al. (1996) will be reconstructed.

## 2. WORKSPACE VARIATION DURING MACHINING

When Stewart introduced his six degree-of-freedom mechanism in 1968, it was suggested by a reviewer of his paper that the mechanism could provide the basis for a new type of machine tool. It has not been until recently, though, that computing power and control algorithms have reached the level of cost and sophistication necessary to make this idea feasible. The versatility provided by six degree-of-freedom, parallel manipulators offers unique opportunities to increase machining performance in regards to accuracy and productivity, as compared to the current generation of serial-mechanism-based machine tools. Reduction of machining costs from higher material removal rates and eliminating the need for complex fixtures are among the many advantages which are evident to the entire manufacturing community. However, efficient use of Stewart platform based machine tools requires operators and production engineers to have an understanding of how parallel mechanism machine tools carry out machining operations. In particular, a basic understanding of the machine's workspace will allow users to determine the suitability of Hexapod machine tools to their own machining tasks. This knowledge will also allow users to identify locations of the machine tool's work volume where specific machining operations may take place.

For conventional machine tools, the concept of workspace is straightforward. It is usually a cubic volume specified by three indexed numbers representing the maximum machine travel in the x, y, and z directions. The workspace, or these three indexed numbers, remain unchanged during machining, except when changes are made to the tool offset. For a Stewart platform, however, the irregularly shaped workspace varies in volume, shape, and position as the platform orientation changes. Such differences characterize the advantages and challenges presented to operators on the shop floor when using this new type of machine tool. For example, Hexapod users may have to estimate the maximum size of parts able to be machined based on the complexity of the part geometry presented to them. Therefore, this workspace study is important in increasing the usability of Hexapod machine tools.

Before we present the algorithm developed in this research to determine the workspace, we need to define the term *workspace* as the three dimensional space the tip of the cutting tool can reach at a given platform orientation. For simplicity, in this research we define the workspace as the volume of space that the nose of the spindle can reach. The tool offset can be arbitrarily set in the algorithm by the user for specific workspace studies. Since an infinite number of platform orientations are possible, it follows that an infinite number of workspace plots may exist for a specific parallel manipulator. This fact points out the necessity and importance of the work presented in this paper.

## 3. FORMULATION OF WORKSPACE CONSTRAINTS

The Ingersoll Octahedral Hexapod machine tool (Figure 1) located at the National Institute of Standards and Technology serves as a case study for our proposed method of workspace determination. As illustrated, the machine tool consists of a movable platform supported by six struts. A cutting tool, such as an end mill, is placed in the spindle of the movable platform. In this configuration, a part to be machined is fixed on the table and kept stationary during machining. It is the motion of the platform in three dimensional space which carries out the machining operation. Unlike most Stewart platforms, however, the Hexapod under study inverts the 6-DOF movable platform and suspends it from an octahedral space frame. The workspace of the NIST Ingersoll Hexapod is defined by three constraints:

1. The limiting lengths, or maximum and minimum lengths of the six struts.
2. The limiting angular travel limits of the twelve spherical joints. Among them, six joints are associated with the stationary base, the other six joints are associated with the movable platform.
3. Possible collisions between individual struts.

To formulate the three constraint sets determining the workspace, the machine's inverse kinematics are calculated. For a given position and orientation of the platform in the three-dimensional work volume, the corresponding strut lengths and spherical joint angles can be determined (Fichter 1986). In this research, two coordinate systems, the base and the platform coordinate systems, are introduced, as illustrated in Figure 2. The position and orientation of the movable platform can be represented by a position vector and a set of three angles in the base coordinate system as follows:

$$^{BASE}\vec{T} = \begin{bmatrix} T_x & T_y & T_z \end{bmatrix}^T \qquad \text{and} \qquad \mathbf{O} = (\alpha, \beta, \gamma)$$

where $\alpha$ =roll, $\beta$ =pitch, $\gamma$ = yaw. The Euler angle convention assumes that the platform is first rotated about the x-axis (roll), followed by rotation about the y-axis (pitch), and finally by rotation about the z-axis (yaw). The composite rotation matrix, $R_{\alpha\beta\gamma}$, is computed by pre-multiplication of the individual rotation matrices, $R_\alpha$, $R_\beta$, and $R_\gamma$, namely,

$$R_{\alpha\beta\gamma} = R_\gamma R_\beta R_\alpha,$$ where

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \quad R_\beta = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \quad R_\gamma = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In addition, the Stewart platform geometry gives us the positions of the six platform joints relative to the platform coordinate system, **P**$_i$, along with the six base joints relative to the base coordinate system, **B**$_i$, as illustrated in Figure 2. To represent the platform joint position vectors in the base coordinate system, the transformation calls for rotation and translation of the platform vectors. The rotation transform is given by:

$$^{R}\vec{P}_i = R_{\alpha\beta\gamma}\ \vec{P}_i$$

The translation transformation is a sum of the rotated platform vectors and the platform position vector, giving the position of the platform joints relative to the base coordinate system as follows:

$$^{BASE}\vec{P}_i = \vec{T} + {}^{R}\vec{P}_i$$

As illustrated in Figure 3, the strut vectors are determined by:

$$\vec{S}_i = {}^{BASE}\vec{P}_i - \vec{B}_i$$

### 3.1 Constraint Set I: Limiting Strut Lengths

Strut lengths, L$_i$, represent the first major workspace constraint for parallel linked manipulators.

$$L_i = \left|\vec{S}_i\right| = \left|\vec{T} + R_{\alpha\beta\gamma}\vec{P}_i - \vec{B}_i\right|$$

For each of the six struts, its length must adhere to the following physical limits.

$$L_{\min} < L_i < L_{\max} \qquad \textit{for i=1,2, 3, 4, 5 and 6}$$

## 3.2 Constraint Sets II and III: Spherical Joint Angular Limits

The Hexapod under study has twelve spherical joints that connect the struts to the platform and base. As illustrated in Figure 2, six of these spherical joints are located on the platform, and the other six are on the base. These spherical joints allow the struts to freely change orientation as the platform moves. However, the spherical joints themselves are physically limited to a maximum angle ($\theta_{max}$) due to the physical constraints of the spherical joints and to prevent collisions with the base structure.

Note that the spherical joint angles of the Stewart platform are measured relative to the *nominal strut vectors*. The strut vectors are nominal when the spherical joint angles (both base and platform) are zero degrees. The nominal strut vector simply provides a reference for spherical joint angle calculations from the static Hexapod geometry. The six unit strut vectors, $\hat{s}_i$, and six nominal unit strut vectors, $\hat{s}_{i_{nom}}$, are determined by:

$$\hat{s}_i = \frac{\bar{S}_i}{|\bar{S}_i|} \qquad \hat{s}_{i_{nom}} = \frac{\bar{S}_{i_{nom}}}{|\bar{S}_{i_{nom}}|} \qquad \textit{for } i = 1, 2, 3, 4, 5 \textit{ and } 6$$

The magnitude of the base spherical joint angle is determined by:

$$\theta_{i\,Base} = \cos^{-1}\left(\hat{s}_i \cdot \hat{s}_{i_{nom}}\right) \qquad \textit{for } i = 1, 2, 3, 4, 5 \textit{ and } 6$$

To determine the platform spherical joint angles and represent them in the base coordinate system, the following transformation is needed. We rotate the nominal unit strut vectors.

$$^{R}\hat{s}_i = R_{\alpha\beta\gamma}\ \hat{s}_{i_{nom}}$$

The dot product of the unit strut vector and the vector after rotation transformation gives the magnitude of the corresponding platform spherical joint angle.

$$\theta_{i\,Platform} = \cos^{-1}\left(\hat{s}_i \cdot {}^{R}\hat{s}_{i_{nom}}\right) \quad \textit{for } i = 1, 2, 3, 4, 5 \textit{ and } 6$$

It should be pointed out that the sphere angle of a spherical bearing has its allowable range because of its structural design. Therefore, the six base sphere angles and six platform sphere angles represent another important set of constraints to define the workspace of the Hexapod machine tool. In other words, the base and platform sphere angles must adhere to the following physical limits. These constraints are given by

$$\left|\theta_{i\ Platform}\right| < \theta_{i\ Max\ Platform} \qquad for\ i = 1, 2, 3, 4, 5\ and\ 6 \qquad Constraint\text{-}II$$

$$\left|\theta_{i\ Base}\right| < \theta_{i\ Max\ Base} \qquad for\ i = 1, 2, 3, 4, 5\ and\ 6 \qquad Constraint\text{-}III$$

### 3.3 Constraint Set IV: Avoiding Strut Collision

Examining Figure 1, the geometry of the Hexapod under study does not preclude the possibility of collisions between the individual struts during the movement of the platform. To avoid collisions, limitations on the space in which an individual strut can reach are enforced in the machine tool design stage. These limitations certainly have significant effects on reducing the machine's workspace. Physical interpretation of avoiding strut collision for the purpose of formulating a corresponding set of constraints, is to keep a pair of struts apart at a certain distance from each other. A mathematical formulation of such a constraint set is described in the following procedure:

(1) Model two individual struts as two vectors, namely, $\vec{u}_1$ and $\vec{u}_2$ (Figure 4). By choosing a platform configuration in the workspace, two vectors ($v_{1a}$ and $v_{1b}$) are formed with respect to a strut, say strut 1. Note that $v_{1a}$ is a vector pointing to one end point of the strut segment and $v_{1b}$ is a vector pointing to the other end point. The role of vector sum gives $\vec{u}_1 = \vec{v}_{1a} - \vec{v}_{1b}$. Following a similar procedure, we have $\vec{u}_2 = \vec{v}_{2a} - \vec{v}_{2b}$ where $v_{2a}$ and $v_{2b}$ are the two vectors constructed from a chosen reference point for strut 2.

(2) A cross product of $\vec{u}_1$ and $\vec{u}_2$ gives a new vector, $\hat{u}$, perpendicular to both $\vec{u}_1$ and $\vec{u}_2$.

$$\hat{u} = \frac{\vec{u}_1 \times \vec{u}_2}{|\vec{u}_1 \times \vec{u}_2|}$$

(3) The distance between the two struts is determined by a dot product of the two vectors, $\hat{u}$ and $\vec{r}_{21}$:

$$\vec{r}_{21} = \vec{v}_{2a} - \vec{v}_{1a}$$

$$D_{1,2} = \vec{r}_{21} \cdot \hat{u}$$

It should be pointed out that with the geometry of the NIST Ingersoll Hexapod, only strut collisions between struts 1 & 2, joints 3 & 4, and joints 5 & 6 are a concern. This represents the formulation of the final constraint set, or constraint IV, on the work volume of the NIST Ingersoll Hexapod.

$$\{ D_{1,2}\ D_{3,4}\ D_{5,6} \} \geq D_{min}$$

### 3.4 Other Workspace Constraints

In this paper, we have presented the formulation of four constraint sets for determining the workspace of a particular Hexapod machine tool, considering limitations on the strut lengths, joint angles, and strut collision. The presented algorithm should be able to include other forms of workspace constraints. Such constraints could be due to possible collisions of the struts with the part being machined or other external objects, and/or a different strut/joint arrangements than the Hexapod machine tool under the current investigation.

```
        $\theta = \theta + \Delta\theta$
    end loop
```

## 4.3 Workspace Boundary Algorithm and Error Analysis

The workspace search algorithm is based upon finding a ρ that represents the boundary of the workspace. The algorithm to determine this boundary depends on one assumption made in this research regarding the geometry of the workspace map. We have assumed that if a point $(\theta,\phi,\rho_1)$ is found to violate the Hexapod machine tool constraints, all points $(\theta,\phi,\rho_2)$ such that $\rho_2>\rho_1$ will also violate the Hexapod constraints. Based on this assumption, the procedure for finding the boundary can be given by the following pseudocode for a successive approximation approach.

Workspace Boundary Algorithm

```
ρ = value outside of the workspace
$\Delta\rho = \rho / 2$
loop while $\Delta\rho > \varepsilon$ desired precision
    if TestPose($\theta,\phi,\rho$, $\mathbf{O}_1$) = true
        $\rho = \rho + \Delta\rho$
    otherwise
        $\rho = \rho - \Delta\rho$
    $\Delta\rho = \Delta\rho / 2$
end loop
```

In terms of the accuracy at the end of the search process, Figure 6 graphically depicts the variation of the search vector at the final stage of the search process. As illustrated, the search vector changes in ρ until $\Delta\rho = \varepsilon$ is reached, where ε is set to 0.001 meters in the current work. Although this algorithm does not guarantee that the final value for ρ is inside the workspace, it guarantees that the error level of ρ is within ± ε of the workspace boundary (Figure 6). The last step in the procedure is to subtract the final value of ρ by ε to guarantee that **V** lies within the workspace boundary. Thus, the possible error introduced by the search algorithm is between zero and -2ε.

### 4.4 Evaluation of the Workspace Volume

The workspace map is defined by numerous data points, **V**, centered about a search origin, **C**. Let a *sector* be defined by four adjacent workspace data points, {**V**$_1$, **V**$_2$, **V**$_3$, **V**$_4$} and the workspace search origin, **C**, as illustrated in Figure 8. Summing the volume of all unique *sectors* determines the total volume of a workspace map. The volume of a sector (Figure 7) is approximated as a four sided pyramid by the following method:

1. Determine the average length of these four vectors.

$$l = \frac{|\bar{v}_1| + |\bar{v}_2| + |\bar{v}_3| + |\bar{v}_4|}{4}$$

2. Approximate the arc lengths as the base and width of the pyramid.

$$base = \Delta\phi \cdot l$$

$$width = \Delta\alpha \cdot l$$

3. The height of the pyramid is given by

$$height = \sqrt{\left(l^2 - \frac{\left(width^2 + base^2\right)}{4}\right)}$$

4. The volume of a specific sector can be calculated by the known base, width, and height and is given by

$$V_{\text{sector}} \approx V_{pyramid} = \frac{base \times height \times width}{3}$$

5. The total workspace volume is determined by summing the calculated volumes of all the sectors.

$$V_{workspace} = \sum_i V_{sector_i}$$

The actual error for a workspace map defined by 1200 sectors in the current work is approximately 0.5%. This value was determined by using the program to calculate the volume of a sphere with a radius equal to one and comparing it to the analytical solution,

$$V_{sphere} = {}^4/_3\, \pi\, r^3 = 4.1888 \quad \textit{where } r = 1$$

**Table 1. Sphere Volume Error Analysis**

| Sectors | Volume | % Error | Runtime |
|---|---|---|---|
| 40 | 3.6962 | 11.8 | $t_o$ |
| 180 | 4.0676 | 2.9 | $2\ t_o$ |
| 760 | 4.1586 | 0.7 | $18\ t_o$ |
| 1200 | 4.1695 | 0.5 | $38\ t_o$ |

Table 1 displays sphere volume calculations for various sector quantities. From this it can be concluded that the error of the volume calculation algorithm decreases as the number of sectors that define a workspace map increases. This is expected as the sectors become finer and increase in quantity. Although the actual Hexapod workspace maps are far from perfect spheres, we can expect the error to have a similar order of magnitude.

# 5. DISCUSSION OF RESULTS

## 5.1 Hexapod Workspace Plots

Figure 8a presents the workspace of the Hexapod machine tool for a horizontal platform. As shown in this figure, the upper half of the workspace is predominantly defined by the minimum strut length limit. Due to the geometry of the Hexapod, the minimum strut length constraint is the limiting factor in the height of the workspace. The Hexapod platform is at its maximum height when all six struts are at their minimum length. The middle section of the workspace is defined by the base spherical joint angle limits and the bottom of the workspace is defined by maximum strut lengths. For a horizontal platform, the workspace boundary does not contain constraint-II, platform joint angles, or constraint-IV, strut collision. These constraints, however, become prominent for platform orientations with high rotation angles.

Figure 8b gives a workspace simulation for a platform roll of 10 degrees. The general shape is similar to the previous workspace with the exception of the platform joint limits. This platform orientation shifts the overall workspace in the positive *y* direction. This shifting trend continues with 20 degrees of roll (Figure 8c). In this case, the negative *y* side of the workspace becomes mostly defined by platform sphere limits. The Hexapod workspace continues to shift in the positive *y* direction as the platform roll deviates from the horizontal orientation.

An important aspect of the Hexapod machine tool is that some sets of platform orientations lead to workspaces which are not continuous with one another. Figure 9 displays three workspace plots for three different platform orientations. This discontinuity implies that it is impossible for the Hexapod platform to move freely during machining operations when three platform orientations are required, such as (roll = 28 deg, pitch= 0 deg), ( roll = -13.6 deg, pitch= 24.7 deg), and (roll = -13.6 deg, pitch= -24.7 deg), at the same point in space. This fact calls for special attention in planning machining operations and points out the need to maintain a minimum deviation of the platform orientation while the platform is traveling between various regions of the work volume.

In addition to the Hexapod under study, the method of workspace determination presented in this paper can be implemented on various other Stewart platforms. The workspace of the INRIA prototype (Gosselin 1990) for a horizontal platform is presented in Figure 10. This figure was created by the workspace program using the geometry and workspace constraints of the INRIA

Stewart platform. This workspace plot compares well with the analytical plot produced by Gosselin.

The workspace of a Spatial Stewart Platform described in Luh et al.(Luh 1996) is graphed in Figure 11. This manipulator is used for flight simulator applications and has no joint angle constraints.

### 5.2 2-D Area Plots of Workspace

Quantitative insight into the workspace of the NIST Ingersoll Hexapod can be obtained if we plot the cross section of the workspace at various heights. Figure 12 displays the cross section of the Hexapod workspace as it rolls from 0 to 20 degrees. These plots represent the cross section at three different heights, -2.6, -2.1, and -1.6 meters, measured from the base of the manipulator. Examining the two-dimensional plots, we see that the cross sectional area decreases and shifts its location in the positive *y* direction as the platform rolls (rotates about the x-axis). Figure 13 a, b, and c, display the normalized cross sectional areas of the Hexapod's workspace as a function of roll for z levels of -2.6, -2.1, and -1.6 meters, respectively. Inspection of the plots clearly shows that a rapid reduction in cross sectional area occurs at approximately a roll of 11 degrees. Before this point, however, the cross sectional area remains more or less constant. This points out the fact that the available workspace may be significantly reduced when the production of complex contoured parts calls for large orientation changes of the spindle platform.

The calculations of workspace volume further demonstrate the workspace variation due to spindle platform orientation changes. Figure 14 plots the workspace volume of the Hexapod machine tool as a function of roll. For a vertical spindle, the workspace has a volume of approximately 2.3 meters cubed. The volume remains mostly constant until a roll angle of 11 degrees is reached. Beyond 11 degrees, the workspace volume decreases in a linear fashion. From examination of the workspace plots, it can be concluded that the drastic decrease in workspace after 11 degrees is due to the platform spherical joint angle constraint.

## 6. CONCLUSIONS

A method to determine the workspace of a Stewart Platform using a numerical spherical search has been presented and a case study completed using the NIST Ingersoll Octahedral Hexapod. Workspace plots show that as the platform orientation of the Hexapod increases, the overall workspace volume decreases. Thus, a tool path that requires high platform rotations will be more constrained in available workspace than tool paths with near vertical platform orientations.

The numerical search method was applied to two other Stewart Platforms previously explored by researchers. Reconstructed workspace plots match the solutions developed by other researchers. More importantly, this method of workspace determination is beneficial in designing new Stewart platform based manipulators. The workspace for various platform geometries may be explored by simply inputting the platform's geometry. With a runtime of less than one minute, the workspace plots can be quickly examined.

## 7. FUTURE WORK

One difficulty in presenting workspace research is that an infinite amount of workspace plots exist for most Stewart platform based manipulators. However, one cannot expect to examine all possible workspace figures to gain an understanding of the manipulator. The difficulty lies in communicating the characteristics of the workspace of a manipulator with the fewest number of workspace plots. This is essential in order to compare the performance of different parallel linked mechanisms. For example, if manipulator A has a larger workspace for a horizontal platform than manipulator B, does this necessarily mean it has a greater workspace at other platform orientations? Certainly not. A standard method to express the Stewart platform workspace needs to be developed.

# 8. REFERENCES

Fichter, E.F., 1986, "A Stewart Platform Based Manipulator: General Theory and Practical Construction", *International Journal of Robot Research*, 5(2): 157-182.

Gosselin, C., 1996, "Determination of the Workspace of 6-DOF Parallel Manipulators, " *Journal of Applied Mechanical Design*, Vol. 112, 331-336.

Hunt, K. H., 1983, "Structural Kinematics of In-Parallel-Actuated Robot Arms", *Trans. ASME J. of Mechanisms, Transmissions, and Automation in Design*, 105:705-712.

Ji, Z., 1994, "Workspace Analysis of Stewart Platforms via Vertex Space," *Journal of Robotic Systems*, 11(7), 631-639.

Kumar, V., 1992, "Characterization of Workspaces of Parallel Manipulators," *ASME Journal of Mechanical Design*, Vol. 114, 368-375.

Lebert, G., Liu, K., Lewis, F.L., 1995, "Dynamic Analysis and Control of a Stewart Platform Manipulator", *Journal of Robotic Systems*, 10(5), 629-655.

Luh, C., Adkins, F., Haung, E., and Qiu., C., 1996, "Working Capability Analysis of Stewart Platforms", *ASME Journal of Mechanical Design*, Vol 118, 220-227.

Stewart, D., 1965, "A Platform with six degrees of freedom", *Proc. Inst. Mech. Engineering*, 371-386.

Wang, J., 1992, "Workspace Evaluation and Kinematic Calibration of Stewart Platforms", Ph.D. Dissertation, Florida Atlantic University, Department of Mechanical Engineering.

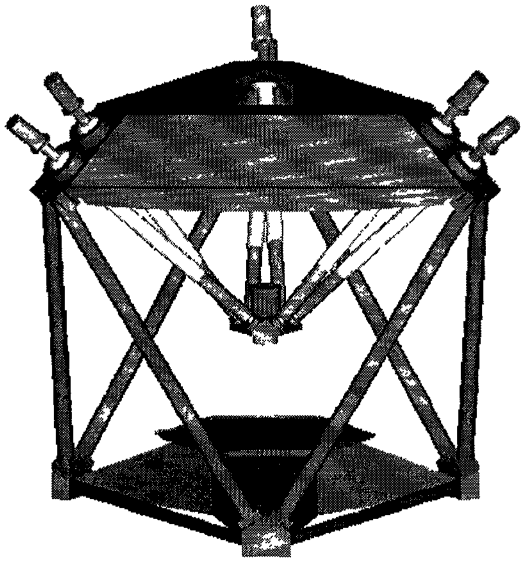


Figure 1. NIST Ingersoll Octahedral Hexapod (Spindle Motor Not Shown)

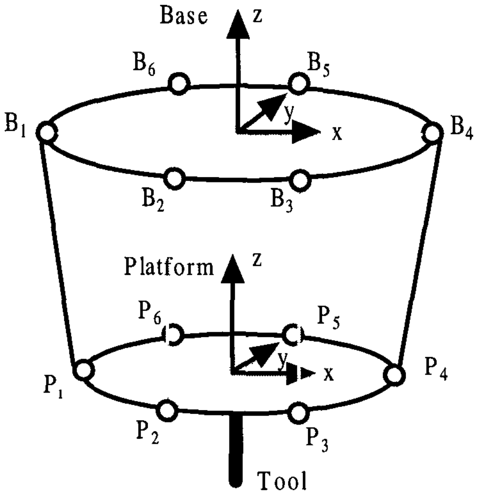


Figure 2. Base and Platform Coordinate Systems

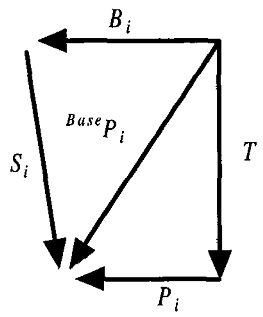


Figure 3. Inverse Kinematic Vectors

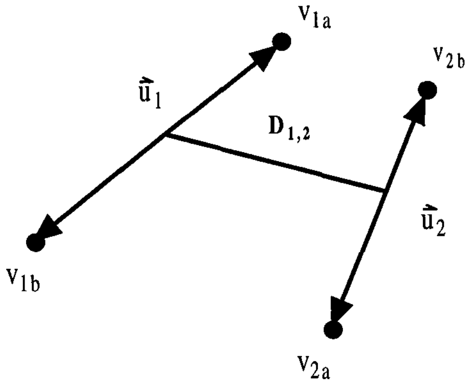


Figure 4. Strut Vectors and Interaction Distance

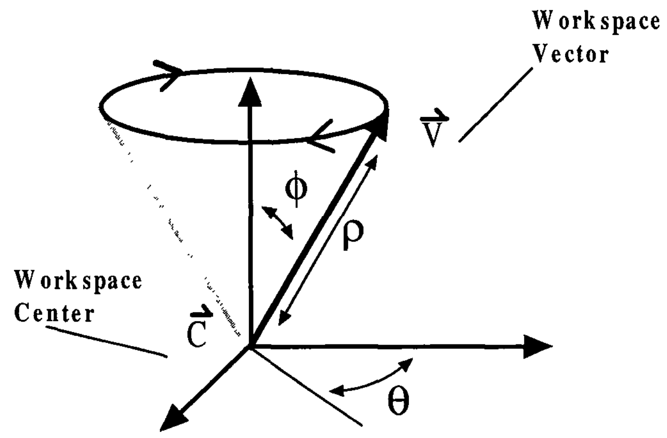


Figure 5. Workspace Search Vectors

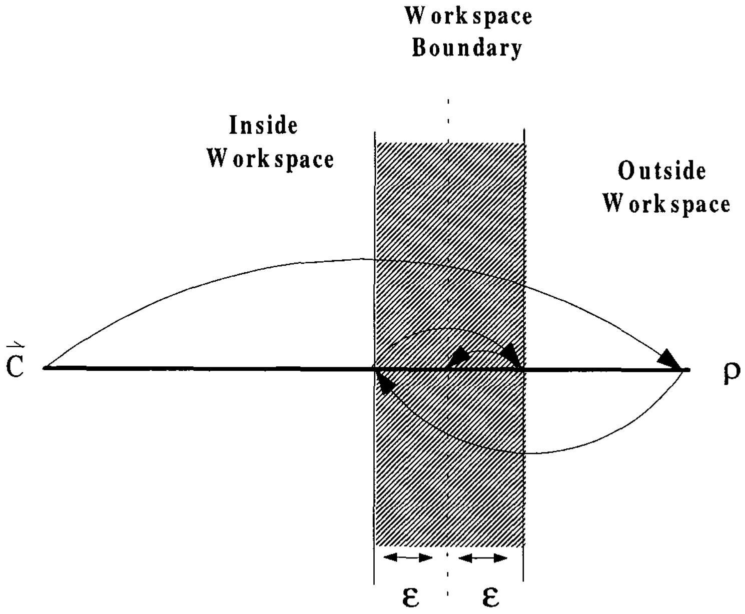


Figure 6. Search Algorithm Error

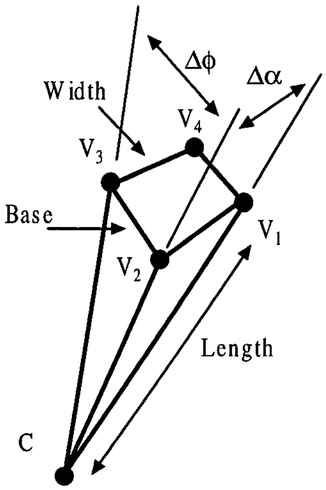
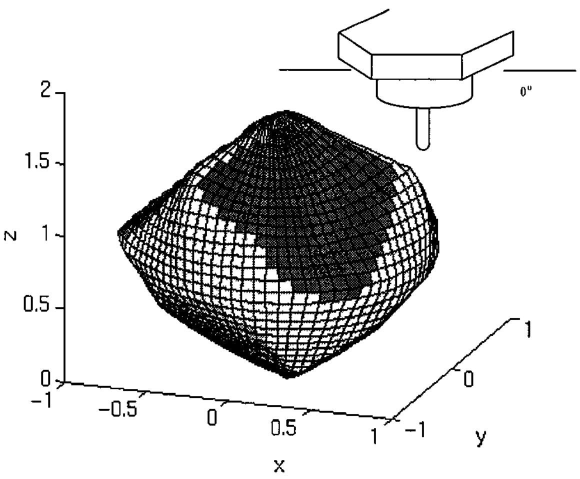
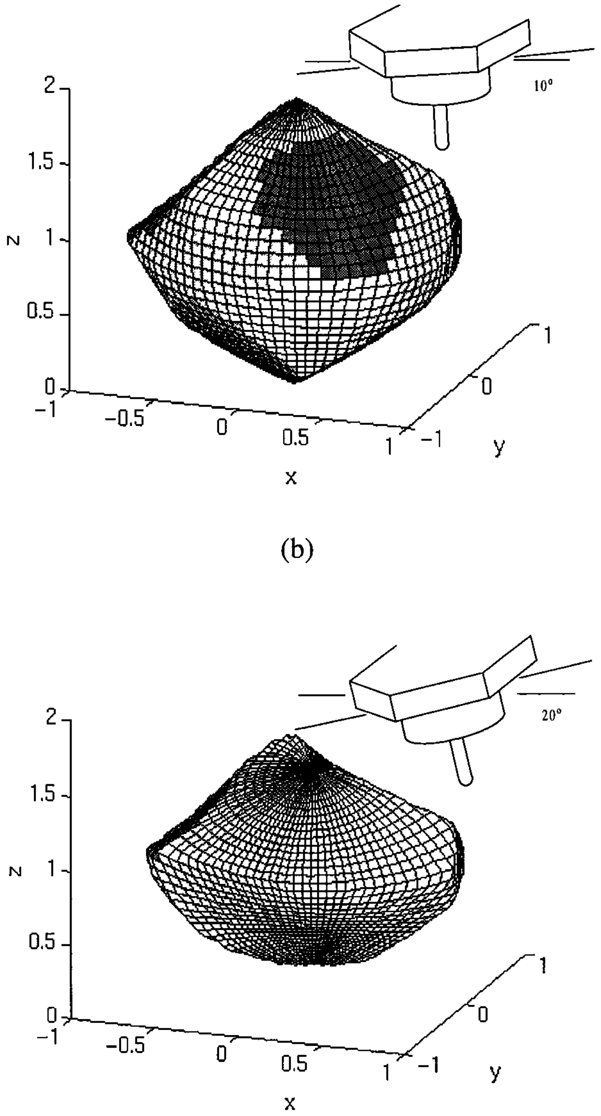


Figure 7. Workspace Volume Calculation



(a)



(b)

(c)

Figure 8. Hexapod Workspaces

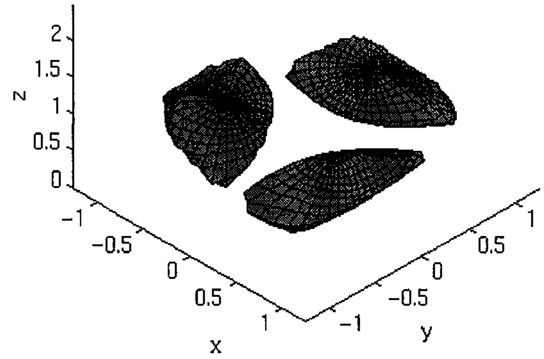


Figure 9. Workspaces for (Roll = 28°, Pitch = 0°), (Roll = -13.6°, Pitch = 24.7°), and (Roll = -13.6°, Pitch = 24.7°),

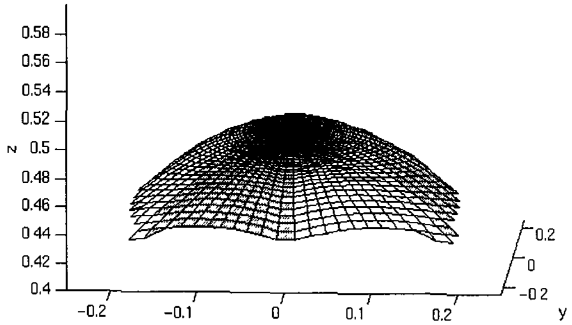


Figure 10. Workspace for INRIA Stewart Platform (Gosselin 1990)

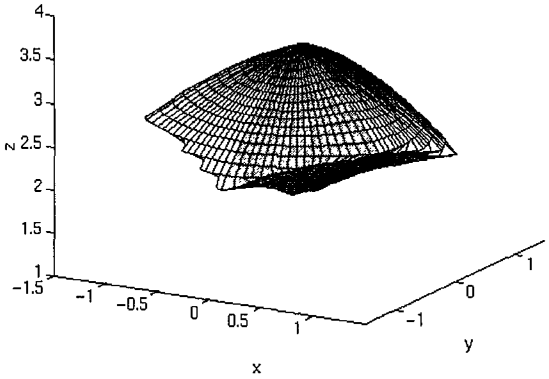


Figure 11. Workspace for Spatial Stewart Platform (Luh 1996)

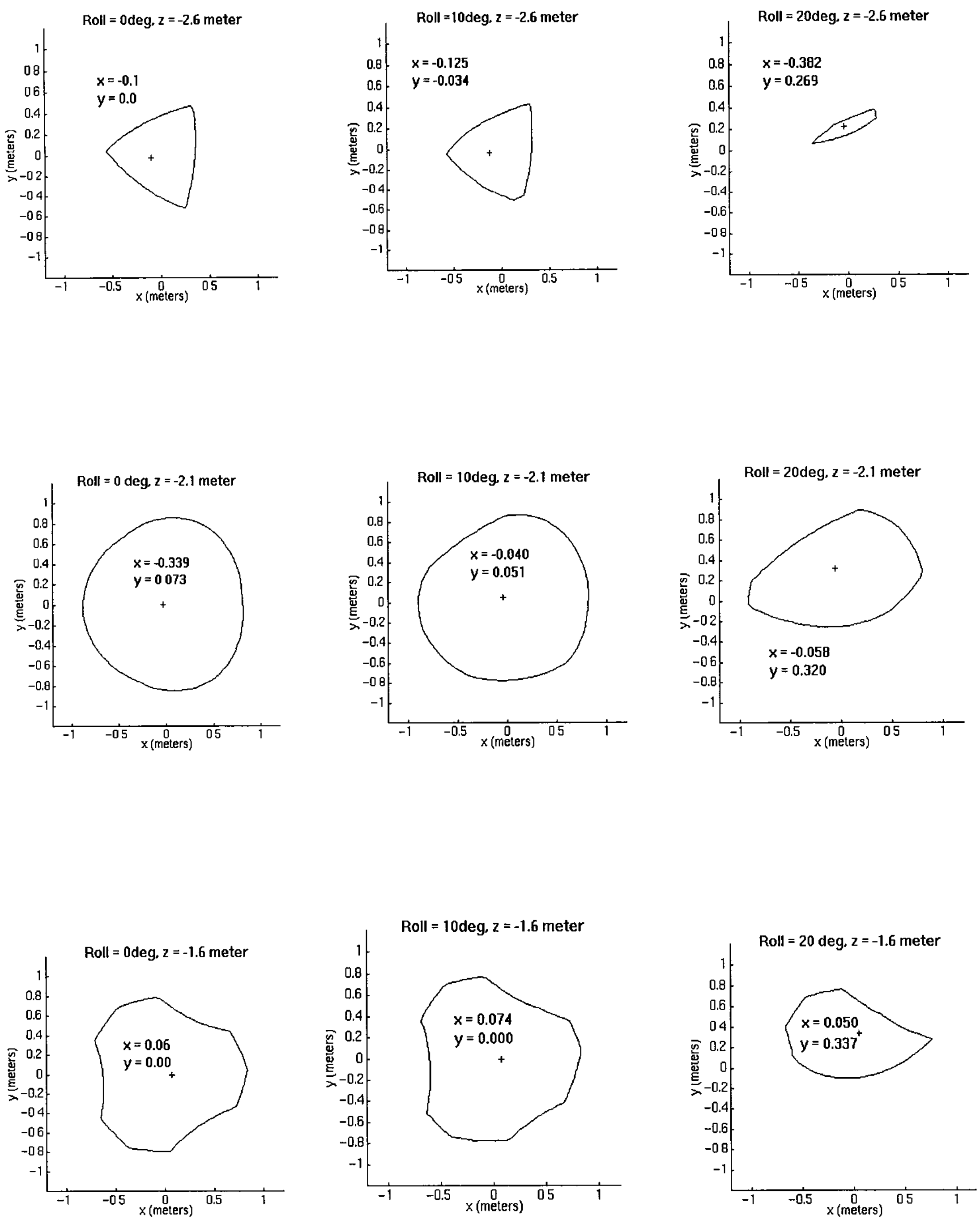
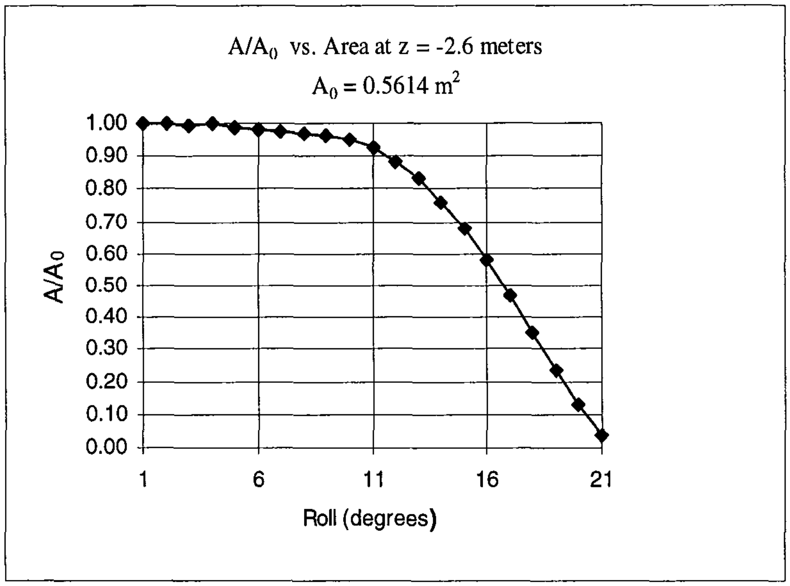
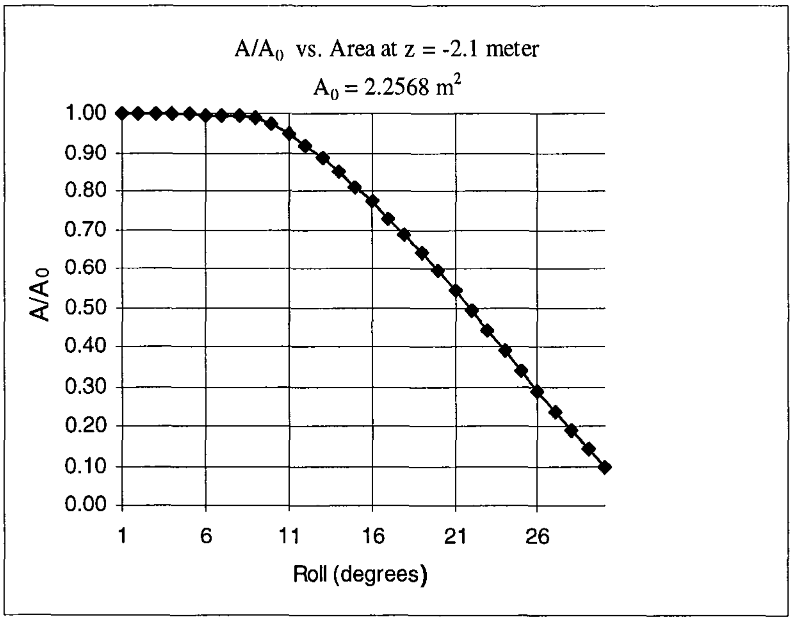


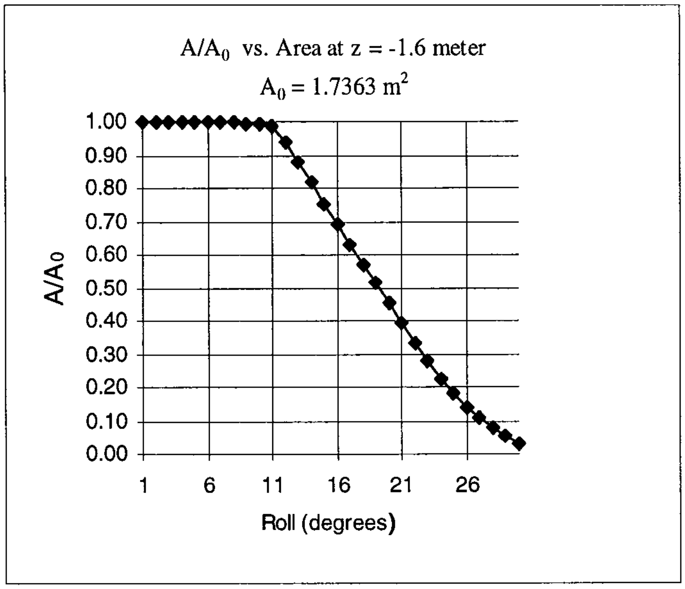
Figure 12. NIST Ingersoll Hexapod Workspace Slices at Z = -2.6, -2.1, -1.6 meters (Base Coordinates)


A/A$_0$ vs. Area at z = -2.6 meters
A$_0$ = 0.5614 m$^2$
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
A/Ao
1
6
11
16
21
Roll (degrees)

(a)


A/A$_0$ vs. Area at z = -2.1 meter
A$_0$ = 2.2568 m$^2$
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
A/Ao
1
6
11
16
21
26
Roll (degrees)

(b)



(c)

Figure 13. a, b, c Normalized Workspace Area In X-Y Plane Versus Spindle Platform Roll

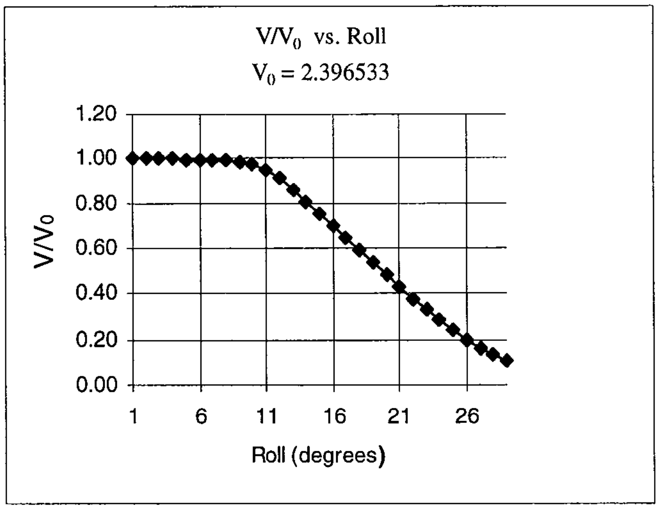


Figure 14. Workspace Volume Versus Spindle Platform Roll

# Stochastic Unit Commitment Scheduling
# and
# Dispatch of Electric Power Systems

By

I. Yan G.L. Blankenship

# STOCHASTIC UNIT COMMITMENT SCHEDULING AND DISPATCH OF ELECTRIC POWER SYSTEMS

I. Yan, Student Member IEEE G. L. Blankenship, Fellow IEEE

Electrical Engineering Department
University of Maryland
College Park, MD 20742

## *ABSTRACT*

Unit commitment, including economic dispatch, is a key component of short term operation scheduling of an electric energy system. Common industry practice is based on the use of a "priority list" for generation scheduling and a deterministic model for power/energy demand. The priority list specifies the next unit to be started or shutdown in response to an increase or decrease in load. A common problem in the use of priority lists is that the next unit is improperly sized to meet the actual change in load.

The algorithm proposed here is more accurate than the priority list method and much faster than dynamic programming which can hardly be applied to systems of more 5 machines. For a system of 41 machines, the algorithm can determine schedules in 0.1 second which is fast enough to do on-line control. Furthermore, the total generating cost is supeiior to that obtained by dynamic programming successive approximations.

## 1. *INTRODUCTION*

In this paper we propose a scheduling algorithm (including economic dispatch) which is fast enough to be used for on-line scheduling in response to *random* changes in demand. The algorithm does not represent a radical departure from current practice. It uses a quadratic function for the fuel cost (heat rate) of thermal units, and a standard exponential function of the cumulative down time to model unit startup costs. A pumped storage facility which may be a composite of several (pumped and unpumped) energy storage systems is included. It uses a *scheduling table* for the generation system which is reminiscient of the priority list, but more comprehensive; and it has an efficient "off-line" scheduling procedure to compute the optimal unit commitment/economic dispatch to meet the (deterministic) expected demand. Based on the solution to the deterministic scheduling problem, a fast, "on-line" algorithm is applied to adjust the commitment and dispatch in response to random fluctuations in demand. The algorithm requires about 0.1 sec to reschedule 41 machines (on a VAX 11/780) over 1 time step. The random fluctuations in load are modeled by white noise or a (non-stationary) Gaussian-Markov process. In the second case, a Kalman filter is used to compute the one-step ahead prediction of the load.

Performance tests for the algorithm are given for systems of 3, 5, 18 and 41 machines. The optimal costs computed are nearly identical to those computed using dynamic programming [1] or a modified dynamic programming successive approximations algorithm [2]. The tests show that the CPU times required to set up the scheduling table and execute the on-line scheduling algorithm grow slower than linearly in the number of machines in the system.

The design and execution of the algorithm are based on the (implicit) assumptions that the starting costs of the generators are substantially smaller than the operating (fuel) costs. We also assume that the short term fluctuations in the load are a small percentage (approximately 4% or less) of the mean load level. The performance of our algorithm reflects the fact that the sensitivity of the system operating cost to perturbations in the demand and "small" changes in the commitment schedule decrease substantially as the number of machines in the system increases. (This fact was exploited in a different way in [3].) We have used a simple model for the scheduling problem. Enhancements to include run time constraints, more complex (Gaussian) load models, and storage system costs (shadow prices) would not substantially impact the performance of the algorithm. Including transmission costs in the scheduling problem would require redesign of the algorithm.

Almost all previous work on unit commitment scheduling has been based on deterministic load models. Work similar to ours in spirit includes the paper of Bertsekas et al[3] who use a duality formulation of the optimal scheduling problem, the work of Turgeon [4] [5] who uses the maximum principle to treat deterministic model similar to the one used here, the thesis of Leguay [6] who uses "impulse control" methods to treat small scale, deterministic scheduling problems, and the recent work of Gonzalez and Rofman [7] [8] who use a clever combinational algorithm to treat modest sized, deterministic scheduling problems including costs for the storage systems (shadow prices). Mathematical treatments of stochastic unit commitment problems were given in Blankenship and Menaldi [9], Li and Blankenship [10]. Starnes's thesis [2] contains an effective *ad hoc* algorithm for treating (deterministic) scheduling problem with more than one performance measure (e.g. system security and cost). More details on the present work may be found in Yan's thesis [11].

## 2. *SYSTEM MODEL*

We consider a system with M machines (thermal or nuclear units) operating over an interval [0,T] (one day or one week). We assume that the system includes a (composite) energy storage system (pumped hydro). The unit commitment problem is to schedule the startup, operating level, and shutdown of the thermal units and the pumping and withdrawal of energy to and from the storage system to meet the time varying demand for power, L(t), t=0,1,2,...,T, at minimum operating cost. In mathematical terms the problem is

$$\min_{t,i} \sum_{t=1}^{T} \sum_{i=1}^{M} [CG(t,i) + CS(t,i)] \tag{2.1}$$

subject to the constraints

$$G(t) = \sum_{i=1}^{M} G(t,i) = L(t) + r(t) \tag{2.2}$$

$$\underline{G}(i) \leq G(t,i) \leq \overline{G}(i)$$

$$\underline{R} \leq r(t) \leq R \tag{2.3}$$

$$S(t) = \min\{\overline{S}, max\{\underline{S}, S(t-1) + r(t-1)\}\} \tag{2.4}$$

where G(t,i) is the generation level of the ith machine in time interval t, r(t) is the energy pumped to (r(t)>0) or withdrawn from (r(t)≤0) storage systems in t, S(t) is the total stored energy at t, and the lower and upper bounds ($\underline{G}(i), \bar{G}(i), \underline{R}, ...$) are physical constraints. The generation cost per unit time is the sum of the running costs, CG(t,i), and starting costs, CS(t,i), of the machines. We assume

$$CG(t,i) = a_1(i) + a_2(i) * G(t,i) + a_3(i) * [G(t,i)]^2 \qquad (2.5)$$

and

$$CS(t,i) = b_1(i) * \{1 - b_2(i) * exp[-b_3(i) * d(t,i)]\} \qquad (2.6)$$

where all the cost coefficients $a_1(i), b_1(i)$, etc., are non-negative and d(t,i) is the cumulative down time of the ith machine at time t.

The *controls* or decision variables for the problem are r(t), G(t,i), t=0,1,...,T, i=1,...,M, and the generation schedule I, an M-vector of 0's and 1's with 1 in the ith position indicating machine i is on, and a zero indicating off. The power set $[(0,1)]^M$ is the collection of all possible schedules. Since continuous controls r(t), G(t,i) are bounded and the schedules I(t) are discrete, the optimization problem involves a non-differentiable objective function.

The *states* of the system are the down time of the machines d(t,i), i=1,2,...,M, and S(t), the stored energy. If a dynamic load model is assumed (see section 6), then the state variables of that model must also be included. If a more elaborate machine model including (minimum) run time constraints or a "banked" state is used, then these states variables must also be added to the state vector. Since our primary concern is to develop an efficient scheduling and dispatch algorithm, we shall not include these features. They do not change the essential structure of the optimization algorithm.

We shall treat the following types of load models : (1) deterministic loads; (2) loads with a pure random fluctuation (white noise) about the mean, and (3) loads with a first order Gaussian-Markov process fluctuation about the mean.

As posed, the problem includes unit commitment (selection of I(t)) and economic dispatch (setting G(t,i) for on-machines and r(t)). The dispatch problem is solved by having all on machines operate at the same incremental cost level

$$\lambda = \frac{d[CG(t,i)]}{d[G(t,i)]} \qquad (2.7)$$

otherwise, by shifting the load of a higher incremental cost unit to machines with lower incremental costs, the overall generation cost could be reduced.

Separating economic dispatch from unit commitment, as was classically done, does not guarantee achievement of the minimum short term scheduling cost.

## 3. *SCHEDULING ALGORITHM*

Our algorithm is designed in two stages: First, an extensive off-line computation is done to compute a "scheduling table," reminiscient of the priority list in common use in the industry. This table need be computed only once for each system. Using the table, the deterministic unit commitment problem is solved "off-line." Then a simple, efficient "on-line" algorithm is used to fine tune the schedule in response to unexpected (or predicted random) changes in load. In effect the off-line algorithm establishes a rough correspondence between the total generation $G(t) = \sum_{i=1}^{M} G(t,i)$ required to serve the load, and the individual generation assignments, G(t,i), in the "first several cheapest generation schedules" for each given level of demand. The on-line scheduling control is then responsible for modest modifications of the machine schedule (and dispatch) to match the actual load. It does this by selecting the best schedule among the "first several cheapest ones" to achieve the optimal total cost when starting costs are taken into account. The implicit assumption in constructing the table is that the starting costs associated with various schedules are substantially less than the fuel costs.

For each load level, the "cheapest" schedules are selected to achieve the corresponding total generation level, with the individual generation levels assigned to the points where all on machines have the same λ (2.7) in the candidate schedules.

### *OFF – LINE ALGORITHM*

First, we build up a scheduling table with K candidate cheapest schedules (G(t,i),i=1,...,M) for total generation G(t) to meet different demand levels. Then given the deterministic demand $\bar{L}(t)$, t=0,1,...,T, with bar implying the stochastic mean, we select the initial power transfer r(t), t=0,1,...,T, to make G(t)=$\bar{L}(t) + r(t)$ as flat as possible. Disturbances are added to r(0),...,r(T) to discover the lowest cost. At each iteration the individual generation levels G(t,i) required to achieve G(t) (total mean load at time t) are selected from the scheduling table. Minor adjustment and compensation of r(0),...,r(T) are made to match physical constraints and demand and achieve minimum cost. This limits the computational burden when dealing with a large number of machines. The final r(t), G(t,i) and G(t) are the mean transfer rates, individual generations and total generation, denoted by $r(t), \bar{G}(t,i)$ and $\bar{G}(t)$, t = 1, .., T, i = 1, ..,M

### *ON – LINE ALGORITHM*

When the load is a random process, on-line scheduling is required to compensate the (planned) deterministic generation and pumping schedule. We regard the stochastic demand L(t),t=0,1,...,T, as a random fluctuation about the mean demand $\bar{L}(t)$. The power transfer $r(t) = \bar{r}(t) + \delta r(t)$ is adjusted to make $G(t) = \bar{G}(t) + \delta G(t)$ as flat as possible for t=0,1,...,T. By using $\bar{r}(t)$ , t=0,1,...,T, as the starting point for the on-line iteration and searching the scheduling table for close feasible solutions, the optimal schedule for the actual current load may be found very rapidly. If the demand fluctuation process can be modeled by Gaussian-Markov process, the Kalman state estimator provides one-step ahead prediction of the demand. This permits a better (lower cost) control since we can smooth out the fluctuations between two time intervals (present and next time steps) by using the storage system and power transfers.

The advantage of this algorithm over the conventional priority list is the consideration of the operating status of "all" the machines in response to a change in demand. The priority list indicates the next machine to be turned on or off in response to an increase or decrease in demand. However, in some cases the optimal response to an increase in demand is to turn some machines on while turning others off. The reverse can happen when demand decreases.

The overall algorithm is summarized in Fig 1.

## 4. *CONSTRUCTION OF THE SCHEDULING TABLE*

A typical scheduling table is shown in Figure 2. The total generation levels G(t) in increments, $\delta G(j)$ which can vary in size, are listed along the left most column. The individual generations G(t,i) required to achieve those levels (including economic dispatch) are listed in the rows of the table, commencing with the cheapest schedule (ignoring starting costs) and continuing to the Kth cheapest schedule. Note for this case, the cheapest schedule to serve level 7420 has machine G1 down and G2 up; the reverse of the solution at level 7400. The number, K, of candidate schedules for each level of demand is chosen in one of two ways:

(1) If t[j,k,$M$ + 1] is the cost entry for the (j,k) row schedule of the table (cost in the right most column), K is chosen in such a way that t[j,K+1,$M$+1]-t[j,K,$M$+1] is larger than the

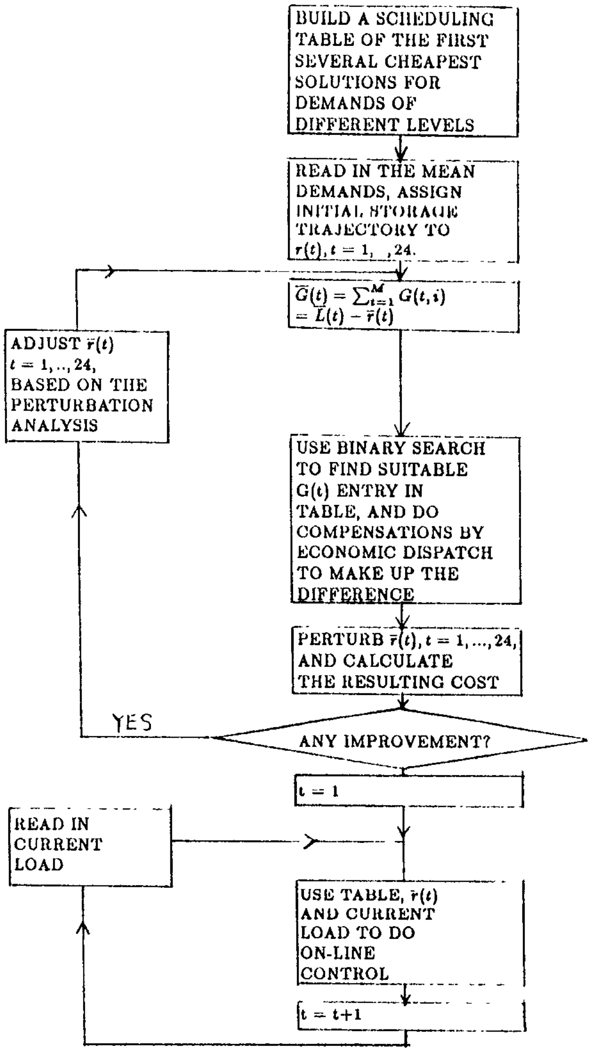

BUILD A SCHEDULING TABLE OF THE FIRST SEVERAL CHEAPEST SOLUTIONS FOR DEMANDS OF DIFFERENT LEVELS
READ IN THE MEAN DEMANDS, ASSIGN INITIAL STORAGE TRAJECTORY TO $r(t), t = 1, \ ,24.$
$\overline{G}(t) = \sum_{i=1}^{M} G(t,i) = \overline{L}(t) - \overline{r}(t)$
ADJUST $\overline{r}(t)$ $t = 1,..,24,$ BASED ON THE PERTURBATION ANALYSIS
USE BINARY SEARCH TO FIND SUITABLE G(t) ENTRY IN TABLE, AND DO COMPENSATIONS BY ECONOMIC DISPATCH TO MAKE UP THE DIFFERENCE
PERTURB $\overline{r}(t), t = 1,...,24,$ AND CALCULATE THE RESULTING COST
YES
ANY IMPROVEMENT?
t = 1
READ IN CURRENT LOAD
USE TABLE, $\overline{r}(t)$ AND CURRENT LOAD TO DO ON-LINE CONTROL
t = t+1

Figure 1. DETERMINISTIC SCHEDULING ALGORITHM

most expensive starting cost of machines which are allowed to switch.

(2) Alternately, empirical evidence from simulation or operation experience may indicate that among the first K cheapest schedules, only the first 2 or 3 are ever applied. Then K may be safely reduced to 4 or 5. This enhances the on-line speed of the algorithm.

The increment of the total generation, $\delta G(j)$, is chosen by the rule

$$\begin{aligned} \delta G(j) &= \min(\delta G, X[G]) \\ &= t[j+1,1,0] - t[j,1,0] \qquad (4.1) \end{aligned}$$

where $\delta G$ is a nominal increment (100MW in our examples) and X[G] is the smallest increment of generation such that G+X[G] has a different optimal unit commitment (I(t,i)) from that at level G. In the example in Figure 2, with $\delta G = 100$, the choices are

$$\min(100, X[7300]) = 100$$

$$\min(100, X[7400]) = 20 \qquad (4.2)$$

since

for j=1 schedule (20,0,40,40,...,600) > commitment (1,0,1,1, .,1)
for j=2 schedule (30,0,30,40,...,600)-> commitment (1,0,1,1,.. ,1)
for j=3 schedule (0,50,60,60,...,600)=> commitment (0,1,1,1,...,1)

The power transfer rates r(t), t=0,1,...,T, are chosen to assure:

(i) the stored energy is periodic S(t)=S(t+T) which implies:

$$\sum_{t=1}^{T} r(t) = 0 \qquad (4.3)$$

(ii) the total generation G(t) is as flat as possible, and

(iii) the constraints $\underline{R} \leq r(t) \leq \overline{R}$ hold.

The key condition (ii) is a consequence of the quadratic form of the generation cost functions CG(t,i).

| G | G1 | G2 | G3 | G4 | ..... | GM | COST | PRIORITY |
|---|---|---|---|---|---|---|---|---|
| 7300 | 22 | 0 | 44 | 41 | .. | 600 | 18000 | first |
| | 0 | 29 | 47 | 41 | ... | 600 | 18120 | second |
| | 30 | 36 | 0 | 45 | ..... | 600 | 18439 | 3rd |
| (j=1) | . | . | . | | . | . | . | |
| | . | . | . | . | . | . | | |
| | . | . | . | . | . | | . | |
| | 30 | 11 | 0 | 67 | .. | 600 | 18700 | Kth |
| 7400 | 34 | 0 | 32 | 49 | ..... | 600 | 18900 | |
| | 31 | 0 | 0 | 87 | .. .. | 600 | 19240 | |
| (j=2) | . | . | . | | | | | |
| | . | . | . | . | . | . | | |
| | 32 | 46 | 48 | 42 | . | 600 | 22320 | |
| 7420 | 0 | 53 | 62 | 67 | ..... | 600 | 24440 | |
| | . | . . | . | . | . | | | |
| (j=3) | . | . | | . | | | | |
| | . | . | . | . | . | | | |
| | | . | . | . | . | | | |
| 15000 | 48 | 42 | 59 | 50 | ..... | 600 | 65229 | first |
| (j=302) | . | . | . | | | | | |

Figure 2. A Typical Scheduling Table.

## 5. *DETERMINISTIC SCHEDULING*

Several test problems with a deterministic demand for power were treated with the (off-line) algorithm to establish a base line for the (on-line) stochastic scheduling algorithm. Operating and starting cost data for the two larger examples, 18 and 41 machines, are listed in Tables 3 and 4 in the Appendix. These systems have been treated earlier using a modified dynamic programming successive approximations algorithm DPSA [2], and the performance results were used as a check on the current algorithm. Smaller examples involving 3 and 5 machines which can be treated by dynamic programming were also used to validate the algorithm. The overall cost figures obtained in these tests are shown in Table 1. The differences in costs in the smaller examples are primarily due to the propagation of quantization errors in the dynamic programming algorithm. The cost differences in the two larger examples are primarily due to quantization effects and the inherent inaccuracy of DPSA.

| NO. OF MACHINES | DPSA OR DP | PRESENT ALGORITHM |
|---|---|---|
| 03 | 0022288.77 | 0022232.448 |
| 03 | 0030210.63 | 0030135.504 |
| 03 | 0022283.14 | 0022251.835 |
| 05 | 0038115.60 | 0038108.249 |
| 18 | 0461047.00 | 0459235.710 |
| 41 | 1623799. 00 | 1584421.308 |

Table 1. OPTIMAL COST COMPARISON

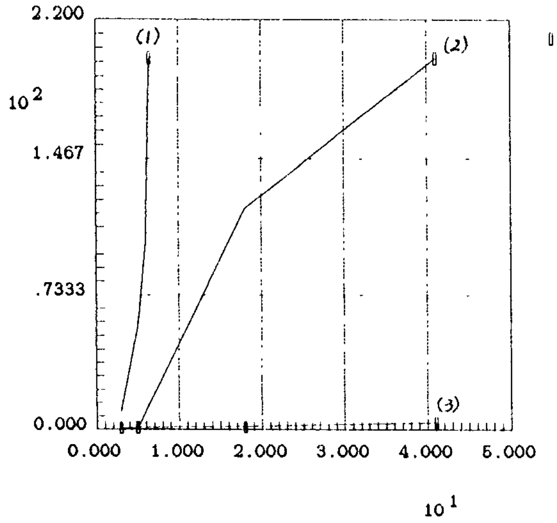
Execution times for the algorithm in this paper are shown in Table 2, including the time required to construct the scheduling table. One day refers to an interval of 24 time steps. The program is written in PASCAL and the times are obtained on a VAX

11/780 operating under Berkeley Unix version 4.1.

| NO. OF MACHINES | BUILD UP SCHEDULING TABLE | ITERATION TO OBTAIN 24 HOUR OPTIMAL SCHEDULE |
|---|---|---|
| 03 | 000:30 | 0:42 |
| 05 | 000:50 | 1:10 |
| 18 | 120:00 | 1:35 |
| 41 | 200:00 | 3:50 |

CPU TIME(MIN:SEC) (OFF-LINE)
Table 2. EXECUTION TIME FOR DETERMINISTIC CASES

The trend in computational times versus number of machines is important. Figure 3 shows that both the time required to build the scheduling table and the unit iteration time for the deterministic algorithm have less than linear growth as functions of machine numbers. The search procedure used to construct the scheduling table was not optimized, and the computational times for this procedure could be substantially reduced if effective search procedures were used.



curve (1) for DPSA
curve (2) for building table
curve (3) for deterministic load iteration
CPU time is in 100 minutes.
Figure 3. CPU TIME VS NUMBER OF MACHINES

## 6. *ON – LINE STOCHASTIC SCHEDULING*

The off-line algorithm determines a schedule and a set of optimal transfer rates $\bar{r}(1), ..., \bar{r}(T)$ corresponding to a known, deterministic demand $\overline{L}(t), t = 0, 1, ..., T$. Starting with $\bar{r}(1), ..., \bar{r}(T)$ and $\overline{G}(t,i), i = 1, 2, ..., M$, the on-line algorithm may be used to determine the best schedule (including starting costs), G(t,i), i=1,...,M., and optimal transfer rates r(1),...,r(T) to meet the actual, observed (random) demand L(t), t=0,1,...,T. The algorithm is fast enough to be used on-line.

In testing the algorithm we used two different models for the demand. The first was simply

$$L(t) = \overline{L}(t) + \delta L(t) \tag{6.1}$$

with $\delta L(t)$ a sequence of independent zero mean random variables uniformly distributed in the interval $[-\alpha\overline{L}(t), \alpha\overline{L}(t)]$ with $\alpha$ a small number. In the tests we used $\alpha = 0.04$, so the demand fluctuations were less than or equal to 4% of the mean at all time steps.

In the second demand model, we assumed that the demand fluctuations were a first order Gaussian-Markov process

$$\delta L(t+1) = a * \delta L(t) + w(t) \tag{6.2}$$

with 'a' a real constant and $w(t), t = 0, 1, ....$ a sequence of zero mean, independent, identically distributed Gaussian-Markov process(i.e. a Gaussian white noise process). We used the model

$$y(t) = \delta L(t) + v(t) \tag{6.3}$$

with $v(t)$ Gaussian white noise process to describe the measurements of the demand fluctuations. We assume

$$E[v(t)w(s)] = 0 \qquad \forall\ t, s$$

$$E[v(t)\delta L(0)] = 0 = E[w(s)\delta L(0)] \quad \forall\ t, s \tag{6.4}$$

$$E[w(t)w(s)] = W(t) * \delta_{st}$$

$$E[v(t)v(s)] = V(t) * \delta_{st}$$

$$E[\delta L(0)] = 0, \quad E[(\delta L(t))^2] = \sigma(t)$$

$$\delta_{st} = \begin{cases} 0, & s \neq t \\ 1, & s = t \end{cases} \tag{6.5}$$

(The parameters a,$w(t)$,$v(t)$ and $\sigma(t)$ must be identified from actual load data.)

The Kalman filter is the best estimator of $\delta L(t)$ given $y(s)$, $s \leq t-1$. The equations are

$$\delta L(t+1|t) = a * \delta L(t|t-1) + K(t) * [y(t) - \delta L(t|t-)] \tag{6.6}$$

where $\delta L(t|t-1)$ is the best (linear) estimate of $\delta L(t)$ given $y(s), s \leq t-1$, and

$$K(t) = \frac{a * P(t)}{P(t) + V(t)}$$

$$P(t+1) = \frac{a * P(t) - a * P(t)^2}{P(t) + V(t) + W(t)} \tag{6.7}$$

$$P(0) = \sigma(0)$$

Both K(t) and P(t) can be computed off-line.

The on-line scheduling algorithm based on the Gaussian-Markov load model is shown in Figure 4. the same algorithm is used for the random fluctuations model, with $\delta L(t+1|t)$ in that Gaussian-Markov model substituted by $\delta L(t)$ in the Figure.

The overall scheduling algorithm is shown in Figure 5.

## 7. *ON – LINE SCHEDULING FOR 41 MACHINES*

In conducting the schedule computations 80 days of (synthetic) hourly demand were used to generate the random load statistics. A break down of the computational times for the algorithm in the two model cases is given in Figure 6. A plot of the computational times required to find the optimal hourly schedule for the two different random demand models versus number of machines is given in Figure 7. Note that the rate of increase is less than linear. The times are longer when the algorithm includes the Kalman predictor, since time is required to execute the additional lines of code. (For the 41 machine case 87 seconds were required to find the optimal schedule for 40 days in 1 hour increments.)

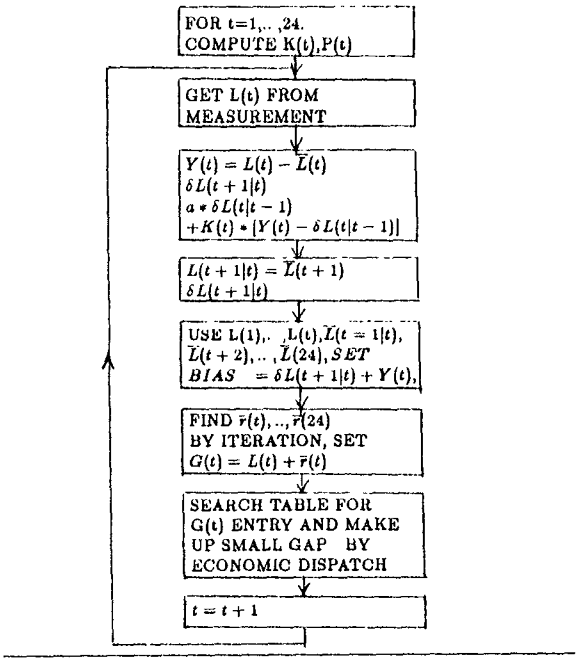
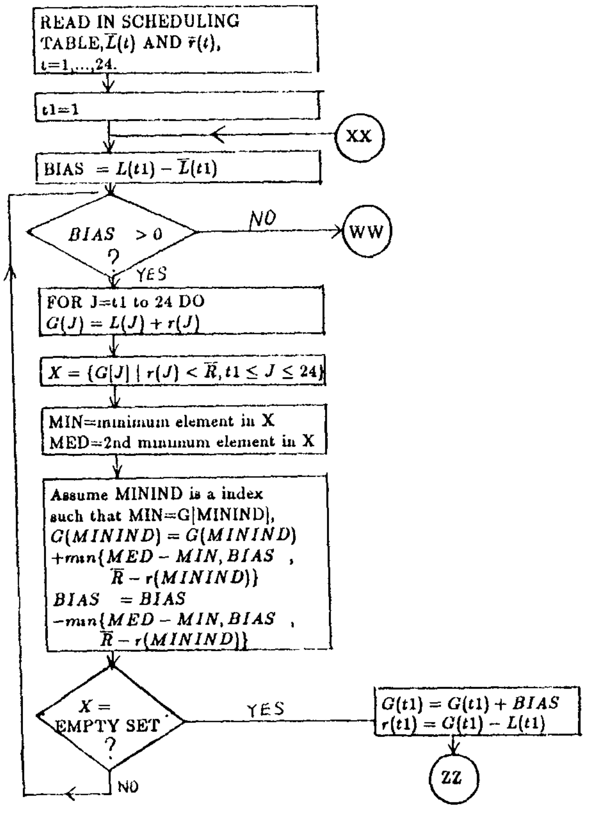
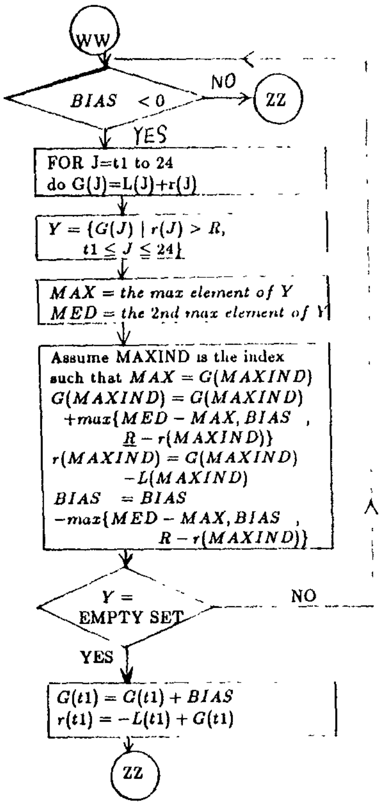

FOR t=1,.. ,24.
COMPUTE K(t),P(t)
GET L(t) FROM MEASUREMENT
$Y(t) = L(t) - \bar{L}(t)$
$\delta L(t+1|t)$
$a * \delta L(t|t-1)$
$+K(t) * [Y(t) - \delta L(t|t-1)]$
$L(t+1|t) = \bar{L}(t+1)$
$\delta L(t+1|t)$
USE L(1),. ,L(t),$\bar{L}(t = 1|t)$, $\bar{L}(t+2)$,.. ,$\bar{L}(24)$, SET $BIAS = \delta L(t+1|t) + Y(t)$,
FIND $\bar{r}(t)$,..,$\bar{r}(24)$ BY ITERATION, SET $G(t) = L(t) + \bar{r}(t)$
SEARCH TABLE FOR G(t) ENTRY AND MAKE UP SMALL GAP BY ECONOMIC DISPATCH
$t = t + 1$

Figure 4. GAUSSIAN MARKOV MODEL SCHEDULING


READ IN SCHEDULING TABLE,$\bar{L}(t)$ AND $\bar{r}(t)$, t=1,...,24.
t1=1
XX
BIAS $= L(t1) - \bar{L}(t1)$
$BIAS > 0$ ?
NO
WW
YES
FOR J=t1 to 24 DO $G(J) = L(J) + r(J)$
$X = \{G[J] \mid r(J) < \bar{R}, t1 \leq J \leq 24\}$
MIN=minimum element in X
MED=2nd minimum element in X
Assume MININD is a index such that MIN=G[MININD], $G(MININD) = G(MININD) + min\{MED - MIN, BIAS, \bar{R} - r(MININD)\}$ $BIAS = BIAS - min\{MED - MIN, BIAS, \bar{R} - r(MININD)\}$
X = EMPTY SET ?
YES
NO
$G(t1) = G(t1) + BIAS$
$r(t1) = G(t1) - L(t1)$
ZZ


WW
$BIAS < 0$
NO
ZZ
YES
FOR J=t1 to 24 do G(J)=L(J)+r(J)
$Y = \{G(J) \mid r(J) > R, t1 \leq J \leq 24\}$
$MAX$ = the max element of Y
$MED$ = the 2nd max element of Y
Assume MAXIND is the index such that $MAX = G(MAXIND)$ $G(MAXIND) = G(MAXIND) + max\{MED - MAX, BIAS, \underline{R} - r(MAXIND)\}$ $r(MAXIND) = G(MAXIND) - L(MAXIND)$ $BIAS = BIAS - max\{MED - MAX, BIAS, R - r(MAXIND)\}$
Y = EMPTY SET
NO
YES
$G(t1) = G(t1) + BIAS$
$r(t1) = -L(t1) + G(t1)$
ZZ

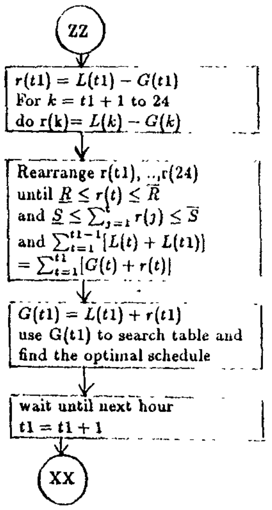
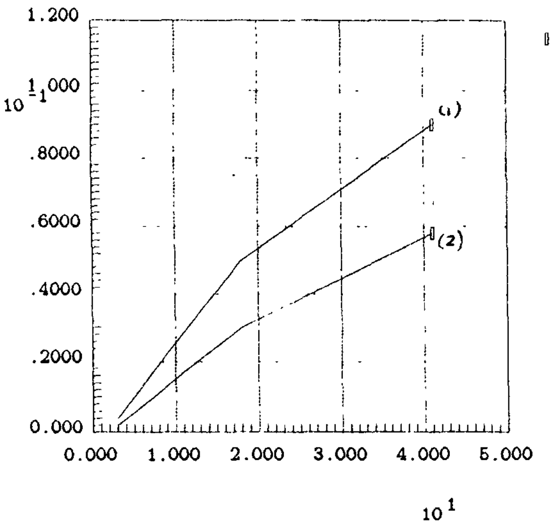

ZZ
$r(t1) = L(t1) - G(t1)$
For $k = t1 + 1$ to 24
do $r(k) = L(k) - G(k)$
Rearrange r(t1), ..,r(24) until $\underline{R} \leq r(t) \leq \bar{R}$ and $\underline{S} \leq \sum_{j=1}^{t} r(j) \leq \bar{S}$ and $\sum_{t=1}^{t1-1}[L(t) + L(t1)] = \sum_{t=1}^{t1}[G(t) + r(t)]$
$G(t1) = L(t1) + r(t1)$
use G(t1) to search table and find the optimal schedule
wait until next hour
$t1 = t1 + 1$
XX

Figure 5. OVERALL ON-LINE ALGORITHM

The algorithm is very effective in smoothing variations in the demand. In a test where the standard deviation of demand fluctuations was 1.42% the average increase of the unit generation cost (defined as the total operating and starting costs divided by total generation in MW) over the deterministic cost was 0.058% for the random disturbance load model and 0.031% for the Gaussian-Markov noise model.

| TYPE OF EXECUTION | RANDOM DISTURBANCE | GAUSSIAN MARKOV DISTURBANCE |
|---|---|---|
| READ IN TABLE | 37.0(off-line) | 30.0(off-line) |
| DETERMINISTIC ONE HOUR SCHEDULE | 0.6667(off -line) | 0.6667(off-line) |
| STOCHASTIC ON-LINE ONE HOUR CONTROL | 0.06063(on-line) | 0.090625(on-line) |

Figure 6. 41 MACHINES SCHEDULING TIMES



curve (1) for Gaussian-Markov disturbance
curve (2) for random disturbance
CPU time is in 0.01 second.

Figure 7. ON-LINE SCHEDULING TIME VS NUMBER OF MACHINES

## 8. *CONCLUSIONS*

The proposed algorithm provides a fast and effective procedure for scheduling unit commitment and randomly varying demand. The speed of the algorithm is a result of the extensive off-line computation done to construct the scheduling table. Since this need only be done once for a given system, the computational expense of this operation is not excessive. The computational requirements of the on-line component of the algorithm are approximately linear in the number of machines - a dramatic improvement over DPSA. By taking advantage of the fact that actual loads different only by a small percentage from the mean expected load for a given day, an extremely accurate initial schedule for generation and storage can be computed (off-line). The (optimal) on-line schedule is essentially a perturbation of the (mean) deterministic schedule, and, as the number of machines increases, the sensitivity of the cost to small perturbations in the demand or in the corresponding schedule is substantially reduced.

The speed of the scheduling algorithm means that it can be used on-line to adjust generation schedules and fine tune dispatch to minimize cost.

*ACKNOWLEDGEMENT*: This research was supported in part by the Division of Electric Energy Systems of the Department of Energy under contract DE-AC01-79ET29244.

## *REFERENCE*

1. F. S. Kuhl, "Unit Commitment and Storage Scheduling via Dynamic Programming," MS Thesis, Electrical Engineering Department, University of Maryland, College Park, 1980.
2. R. A. Starnes, "Multiobjective Optimization in Power System Unit Commitment," MS Thesis, Electrical Engineering Department, University of Maryland, College Park, 1980.
3. D. P. Bertsekas, G. S. Lauer, N. R. Sandell, and T.A. Posbergh, "Optimal Short-Term Scheduling of Large-Scale Power Systems," IEEE Trans. Power Apparatus and Systems, Jan. 1983.
4. A. Turgeon, "Optimal Unit Commitment," IEEE Trans. Automatic Control, Vol. AC-21, PP223-227, April 1977.
5. A. Turgeon, "Optimal Scheduling of Thermal Generaing Units," IEEE Trans. Automatic Control, Vol. AC-23, PP 1000-1005, Dec. 1978.
6. C. Legauy, "Application du Controle Stochastic a un Probleme de Gestion Optimale D'energie," These de Docteur-Ingenieur, Universite de Paris $\overline{IX}$, 1975.
7. R. Gonzalez and E. Rofman, "On Deterministic Control Problems: an Aoproximation Procedure For The Optimal Cost," SIAM J. Control and Optimization, to appear.
8. E. Rofman, "Approximation to Hamalton-Jacobi-Bellman Equation in Deterministic Control Theory: An Application to Energy Production Systems," in Recent Mathematical Methods in Dynamic Programming, I. Capuzzo Dolcetta, W H. Fleming, and T. Zolezzi, eds., Lecture Notes in Mathematics, Vol.1119, Springer Verlag, New York, 1984, PP 152-189.
9. G. L. Blankenship and J.L. Menaldi, "Optimal Stochastic Scheduling of Power Systems With Delays and Large Cost Differentials," SIAM J. Control, 22(1984), pp.121-132.
10. C. W. Li and G. L. Blankenship, "Optimal Stochastic Scheduling of Systems with jump process coefficients," submitted for publication.
11. I. Yan, "An Algorithm For Stichastic Scheduling of Electric Power System," MS Thesis, Electrical Engineering Department, University of Maryland, College Park, 1983.

## *APPENDIX*

*Table 3.* COST COEFFICIENTS FOR 18 MACHINES

GENERATION COST DATA

| M | $\underline{G}$ | $\overline{G}$ | a3[i] | a2[i] | a1[i] |
|---|---|---|---|---|---|
| 1 | 15 | 045 | 0 027039 | 7.4840 | 0025 |
| 2 | 15 | 045 | 0 046586 | 7.1030 | 0044 |
| 3 | 15 | 079 | 0.041011 | 7.2710 | 39 6 |
| 4 | 20 | 053 | 0.023020 | 6.8356 | 42.0 |
| 5 | 20 | 053 | 0.023020 | 6.8356 | 42.0 |
| 6 | 30 | 130 | 0.011685 | 5.0250 | 94.3 |
| 7 | 30 | 105 | 0.012176 | 5.4530 | 68.6 |
| 8 | 30 | 105 | 0.012167 | 5.4530 | 68.6 |
| 9 | 30 | 105 | 0.121670 | 5.4530 | 68 6 |
| 10 | 30 | 105 | 0 012167 | 5.4530 | 68 6 |
| 11 | 35 | 167 | 0.005206 | 5 6787 | 89.0 |
| 12 | 35 | 167 | 0.005206 | 5.6787 | 89 0 |
| 13 | 35 | 167 | 0.005206 | 5.6787 | 89.0 |
| 14 | 35 | 167 | 0.005206 | 5.6787 | 89.0 |
| 15 | 25 | 388 | 0.002269 | 5.8581 | 68.0 |
| 16 | 00 | 425 | 0.000769 | 4 6860 | 28.2 |
| 17 | 50 | 670 | 0.000827 | 6.4870 | 60 4 |
| 18 | 50 | 670 | 0.000827 | 6.4870 | 60.4 |

STARTING COST AS A FUNCTION OF DOWN-TIME

| M | CS(1) | CS(2) | CS(3) | CS(4) | CS(5) | CS(6) |
|---|---|---|---|---|---|---|
| 01 | 083 | 163 | 242 | 318 | 392 | 464 |
| 02 | 083 | 164 | 242 | 318 | 292 | 464 |
| 03 | 065 | 149 | 220 | 290 | 357 | 424 |
| 04 | 040 | 080 | 114 | 149 | 182 | 214 |
| 05 | 040 | 080 | 114 | 149 | 182 | 214 |
| 06 | 133 | 258 | 376 | 486 | 590 | 688 |
| 07 | 181 | 330 | 454 | 555 | 639 | 709 |
| 08 | 039 | 077 | 114 | 150 | 185 | 220 |
| 09 | 039 | 077 | 114 | 150 | 185 | 220 |
| 10 | 055 | 069 | 102 | 135 | 167 | 299 |
| 11 | 156 | 309 | 457 | 602 | 743 | 881 |
| 12 | 156 | 309 | 457 | 602 | 743 | 881 |
| 13 | 156 | 309 | 457 | 602 | 743 | 881 |
| 14 | 156 | 309 | 457 | 602 | 743 | 881 |
| 15 | 9998 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 16' | 9998 | 9990 | 9999 | 9999 | 9999 | 9999 |
| 17 | 9998 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 18 | 9998 | 9999 | 9999 | 9999 | 9999 | 9999 |

*Table 4.* COST COEFFICIENTS FOR 41 MACHINES

GENERATION COST DATA

| M | $\underline{G}$ | $\overline{G}$ | a3[i] | a2[i] | a1[i] |
|---|---|---|---|---|---|
| 1 | 3 5e+01 | 1 7e+02 | 0 00521 | 5.67900 | 89.00000 |
| 2 | 3.5e+01 | 1.7e+02 | 0.00521 | 5.67900 | 89.00000 |
| 3 | 3 5e+01 | 1 7e+02 | 0.00521 | 5.67900 | 89.00000 |
| 4 | 3.5e+01 | 1.7e+02 | 0.00521 | 5.67900 | 89 00000 |
| 5 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 6 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 7 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 8 | 2 5e+01 | 1.0e+02 | 0 00458 | 6 59200 | 35.00000 |
| 9 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 10 | 2 5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35 00000 |
| 11 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 12 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 13 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 14 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 15 | 1.3e+02 | 3.9e+02 | 0.00227 | 5.85800 | 162.00000 |
| 16 | 1.3e+02 | 3.9e+02 | 0.00227 | 5.85800 | 162.00000 |
| 17 | 2.5e+02 | 5.5e+02 | 0.00197 | 6.41700 | 275.00000 |
| 18 | 2.5e+02 | 5.5e+02 | 0.00197 | 6.41700 | 275.00000 |
| 19 | 6.5e+01 | 2.5e+02 | 0.00405 | 6.35400 | 250.00000 |
| 20 | 6.5e+01 | 2 5e+02 | 0.00405 | 6.35400 | 250.00000 |
| 21 | 6.5e+01 | 2.5e+02 | 0.00405 | 6.35400 | 250 00000 |
| 22 | 6 5e+01 | 2.5e+02 | 0.00405 | 6.35400 | 250.00000 |
| 23 | 5.0e+01 | 2.0e+02 | 0.00354 | 6.14200 | 60.000000 |
| 24 | 5.0e+01 | 2.0e+02 | 0.00354 | 6.14200 | 60.000000 |
| 25 | 5.0e+01 | 2 0e+02 | 0.00354 | 6.14200 | 60.000000 |
| 26 | 2.5e+02 | 6.7e+02 | 0.00083 | 6.48700 | 360.00000 |
| 27 | 2.5e+02 | 6.7e+02 | 0.00083 | 6.48700 | 360.00000 |
| 28 | 2.0e+02 | 5.0e+02 | 0.00064 | 7.07600 | 185.00000 |
| 29 | 2.0e+02 | 5.0e+02 | 0.00064 | 7.07600 | 185.00000 |
| 30 | 2.0e+02 | 5.0e+02 | 0.00064 | 7.07600 | 185 00000 |
| 31 | 1.2e+02 | 3.5e+02 | 0.00081 | 6.89900 | 149.00000 |
| 32 | 1.2e+02 | 3.5e+02 | 0.00081 | 6.89900 | 149 00000 |
| 33 | 3.2e+02 | 7.5e+02 | 0.00061 | 6.70400 | 410 00000 |
| 34 | 2.0e+02 | 4.3e+02 | 0.00077 | 6.68600 | 128.00000 |
| 35 | 2.0e+02 | 4.3e+02 | 0.00077 | 6.68600 | 128.00000 |
| 36 | 2.0e+02 | 4 3e+02 | 0.00077 | 6.68600 | 128.00000 |
| 37 | 4.0e+02 | 8.0e+02 | 0.00095 | 4.55100 | 390.00000 |
| 38 | 4.0e+02 | 8.0e+02 | 0.00095 | 4.55100 | 390.00000 |
| 39 | 4.0e+02 | 8.0e+02 | 0.00095 | 4.55100 | 390.00000 |
| 40 | 4.0e+02 | 8.0e+02 | 0.00095 | 4.55100 | 390.00000 |
| 41 | 6.0e+02 | 1.2e+03 | 0.00063 | 3.95100 | 586.00000 |

STARTING COST AS A FUNCTION OF DOWN-TIME

| M | CS(1) | CS(2) | CS(3) | CS(4) | CS(5) | CS(6) |
|---|---|---|---|---|---|---|
| 1 | 156.39 | 308.82 | 457.39 | 602.21 | 743 35 | 880.93 |
| 2. | 56.39 | 308.82 | 457.39 | 602 21 | 743.35 | 880 93 |
| 3 | 156.30 | 308 82 | 457 39 | 602.21 | 743 35 | 880 93 |
| 4. | 156.39 | 308.82 | 457.39 | 602.21 | 743 35 | 880.93 |
| 5. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 6. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 7. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 8. | 180.96 | 330.34 | 453 66 | 555.45 | 639.48 | 708.85 |
| 9. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 10. | 180.96 | 330 34 | 453.66 | 555 45 | 639.48 | 708 85 |
| 11. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 12. | 180.96 | 330.34 | 453 66 | 555 45 | 639 48 | 708 85 |
| 13. | 180 96 | 330.34 | 453 66 | 555 45 | 639 48 | 708 85 |
| 14. | 180.96 | 330 34 | 453.66 | 555.45 | 639.48 | 708 85 |
| 15. | 35.710 | 70.850 | 105.42 | 139.44 | 172.92 | 205.86 |
| 16. | 35.710 | 70.850 | 105 42 | 139 44 | 172 92 | 205 86 |
| 17. | 250.00 | 300.00 | 390.00 | 550 00 | 660.00 | 770 00 |
| 18. | 250.0 | 300.00 | 390.00 | 550 00 | 660 00 | 770 00 |
| 19. | 156.3 | 308.82 | 457.39 | 602.21 | 743 35 | 880 93 |
| 20. | 156 3 | 308.82 | 457.39 | 602.21 | 743.35 | 880 93 |
| 21. | 156.3 | 308.82 | 457.39 | 602.21 | 743 35 | 880.93 |
| 22 | 156.3 | 308.82 | 457.39 | 602.21 | 743.35 | 880 93 |
| 23. | 113.2 | 221.64 | 325.27 | 424 38 | 519 18 | 609.85 |
| 24. | 113.29 | 221.64 | 325.27 | 424.38 | 519.18 | 609.85 |
| 25. | 113.29 | 221.64 | 325.27 | 424.38 | 519.18 | 609.85 |
| 26. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 27 | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 28. | 6320.57 | 8645 | 9501 | 9815 | 9931 | 9974 |
| 29 | 6320.57 | 8645 | 9501 | 9815 | 9931 | 9974 |
| 30. | 6320.57 | 8645 | 9501 | 9815 | 9931 | 9974 |
| 31. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 32. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 33. | 9997 77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 34 | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 35. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 36. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 37. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 38 | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 39. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 40 | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 41. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |