

PC 클러스터링을 이용한 병렬 최적조류계산에 관한 연구

김철홍^{*}, 문경준^{*}, 김형수^{*}, 박준호^{*}, 김진호^{*}, 이화석^{**}
^{*} 부산대학교 전기공학과, ^{**} 거제대학 전기과

Parallel Optimal Power Flow Using PC Clustering

Cheol-Hong Kim^{*}, Kyeong-Jun Mun^{*}, Hyung-Su Kim^{*}, J. H. Park^{*}, Jinho Kim^{*}, Hwa-Seok Lee^{**}
^{**} Dept. of Electrical Engineering Pusan National Univ., ^{**} Koje College

Abstract - Optimal Power Flow (OPF) is becoming more and more important in the deregulation environment of power pool and there is an urgent need of faster solution technique for on-line application. So this paper presents parallel genetic algorithm-tabu search for the solution of the OPF.

The control variables modeled unit active power outputs, generator-bus voltage magnitudes and transformer-tap settings. A number of functional operating constraints, such as branch flow limits, load bus voltage magnitude limits and generator reactive capabilities are included as penalties in the fitness function.

In parallel GA-TS, GA operators are executed for each process. If best fitness of the GA is not changed for several generations, TS operators are executed for the upper three populations to enhance the local searching capabilities. With migration operation, best string of each node is transferred to the neighboring node after predetermined iterations are executed. For parallel computing, we developed a PC-cluster system consisting of 8 PCs. Each PC employs the 2 GHz Pentium IV CPU and is connected with others through ethernet switch based fast ethernet.

To show the usefulness of the proposed method, developed algorithm has been tested and compared on an IEEE 30-bus system in the reference paper. From the simulation results, we can find that the proposed algorithm is efficient for the OPF.

1. 서 론

최적조류계산(Optimal Power Flow : OPF)은 전력계통 운용시 고려해야할 다수의 제약조건을 만족하면서 경제적이고 안전하게 계통을 운용하기 위한 기법이다. 1960년대 초 Dommel과 Tinney[1]에 의해 소개된 이후 많은 연구자들에 의해 연구되고 있는 전력계통의 중요한 분야이며, 최근에는 컴퓨터 기술의 발달에 힘입어 더 많은 연구가 진행되고 있다.

현재 우리나라의 전력산업 환경은 구조 개편으로 인해 경제적, 사회적, 그리고 기술적으로 많은 변화를 겪고 있다. 이러한 변화 속에서 안전하고 높은 신뢰도를 유지하면서 최적으로 계통을 운용하는 것은 점차로 어려운 일이 되고 있다. 따라서 전력계통의 운용과 계획에 가장 기본이 되고 필수적인 최적조류계산에 대한 연구가 요구되고 있다.

종래의 최적조류계산 방법으로 비선형 계획법(Nonlinear Programming)[2], 선형계획법(Linear Programming)[3], 2차계획법(Quadratic Programming)[4]등의 수학적인 프로그래밍 기법 등이 사용되었다. 그러나 이러한 방법들이 전역 최적해를 구하기 위해서는 목적함수가 볼록(convex)해야 하나, 최적조류계산 문제는 비볼록(non-convex)한 특성을

가지므로 기존의 방법으로 전역 최적해를 구하기가 어려운 단점이 있다. 또한 계통의 규모가 큰 경우, 다수의 운용 제약조건, 계통의 비선형성으로 인해 최적의 해를 구하는데 많은 시간이 걸릴 뿐만 아니라 심지어 전역 최적해로 수렴함을 보장할 수가 없다.

최근에는 수학적 프로그래밍 기법의 단점을 극복하고자 유전 알고리즘(Genetic Algorithm)[5,6], 시물레이티드 어닐링(Simulated Annealing)[7], 경험적인 타부 탐색법(Tabu Search)[8]등의 방법이 최적조류계산 문제에 적용되고 있다. 그러나 유전 알고리즘이나 시물레이티드 어닐링은 전역 최적해에 대한 탐색능력을 가지고 있으나, 최적해의 탐색시에 많은 탐색 시간이 걸린다는 단점이 있다. 그리고 타부 탐색법의 경우는 빠른 탐색특성을 가진다는 장점이 있으나 전역적인 최적해 탐색능력이 다소 떨어진다라는 문제가 있다. 따라서 각 알고리즘의 장점을 적절히 활용하여 계산 시간 및 최적해 탐색성능을 개선하려는 연구가 시도되고 있다[9,10].

병렬처리 기법은 여러 개의 프로그램들 또는 프로그램의 분할된 부분들을 다수의 프로세서가 분담하여 동시에 처리하는 기술을 말하며, 단일 컴퓨터에 의해 순차적으로 수행하는 것보다 속도를 크게 향상시킬 수 있다. 기존의 병렬 알고리즘 구현은 주로 병렬 컴퓨터를 이용하였으나 병렬 컴퓨터는 매우 고가여서 쉽게 이용할 수 없고, 확장이 용이하지 않다는 단점이 있었다. 이에 비해 PC 클러스터 시스템은 다수의 저렴한 일반 PC들을 고속 네트워크로 연결하여 마치 하나의 컴퓨팅 시스템처럼 사용함으로써 병렬처리 능력을 가진다. 따라서 PC 클러스터 시스템은 기존의 병렬 컴퓨터와 비교해 저가이며, 추후 확장이 용이하다는 장점이 있다[11].

따라서 본 논문에서는 전역 최적해 탐색 성능이 우수한 유전 알고리즘과 빠른 수렴특성을 가진 타부 탐색법을 이용한 최적조류계산 방법을 제안하였다. 또한 제안한 알고리즘의 최적해 탐색 시간을 단축시키기 위하여 PC 클러스터 시스템으로 알고리즘을 병렬화하였다. 제안한 방법의 유용성을 입증하기 위하여 제안한 방법을 IEEE 30 모선 계통의 최적조류계산 문제에 적용하여 종래의 방법과 비교·검토하였다. 그 결과 제안한 방법이 종래의 방법에 비해 최적해 탐색속도 및 탐색성능 면에서 우수함을 확인하였다.

2. 최적조류계산 문제

최적조류계산은 제어변수를 통하여 계통운용과 관련된 제약조건들을 만족시키면서, 전력계통 운용 시 고려해야 하는 목적함수를 최적화하는 문제로 정의할 수 있으며 목적함수 및 제약조건은 다음과 같다.

$$\begin{aligned} \text{Minimize } & f(x, u) & (1) \\ \text{s.t. } & g(x, u) = 0 & (2) \\ & h(x, u) = 0 & (3) \end{aligned}$$

여기서,

- x : 상태변수
 - 전압의 위상각
 - 부하 모션에서의 전압 크기
 - 발전기의 무효전력 출력
- u : 제어변수
 - 발전기의 유효전력 출력(Slack Bus 제외)
 - 발전기 모션 전압
 - 변압기의 변압비
- $f(\cdot)$: 목적함수
- $g(\cdot)$: 등식제약조건
- $h(\cdot)$: 부등식제약조건

본 논문에서는 유효발전 비용의 최소화를 목적함수로 설정하였으며 이를 식 (4)에 나타내었다.

$$\text{Min}F(P_g) = \sum_{i=1}^{N_g} (a_i + b_i P_{gi} + c_i P_{gi}^2) \quad (4)$$

여기서,

- $F(P_g)$: 총 발전 비용
- P_{gi} : 각 발전기의 유효 전력출력
- i : 발전기의 모션 번호
- N_g : 계통 전체의 발전기 모션의 수
- a_i, b_i, c_i : 각 발전기의 연료계수

등식제약조건(2)은 전력조류방정식을 사용하였다. 그리고 부등식제약조건(3)으로 다음의 제약조건을 고려하였다.

- 선로 조류 제약 (MVA, MW, A)
- 부하 모션의 전압크기
- 발전기의 무효전력 출력용량
- 슬랙 모션의 유효전력 출력량

3. PC 클러스터 시스템을 이용한 유전 알고리즘과 타부 탐색법의 병렬화

3.1 PC 클러스터 시스템

본 논문에서는 PC 클러스터 시스템으로 제안한 알고리즘을 병렬 처리하여 최적해 탐색에 소요되는 계산시간을 단축하고자 하였다. PC 클러스터 시스템이란 다수의 PC 또는 워크스테이션을 고속 네트워크로 연결하여 하나의 컴퓨팅 시스템으로 사용함으로써 고성능 또는 고가용성을 얻을 수 있는 기술을 말한다. PC 클러스터 시스템의 구조를 그림 1에 나타내었다.

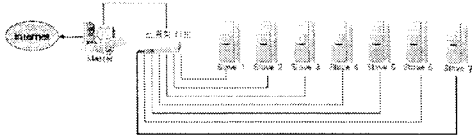


그림 1 PC 클러스터 시스템의 구조
Fig. 1 Structure of PC cluster system

본 논문에서 구현한 PC 클러스터 시스템은 ethernet switch 기반 fast ethernet 방식의 8-노드로 구성하였다. 운영 체제는 master 노드에는 Windows 2000 server, 나머지 노드들에는 Windows 2000 pro를 사용하였으며, PC간에 상호 연결하기 위해 fast ethernet card와 스위칭 허브를 사용하였고, 데이터 전송과정에서는 TCP/IP를 이용한 메시지 전달(message-passing) 방식을 이용하는 MPI 라이브러리를 이용하였다. 각 노드의 원격 관리를 위한 프로그램으로는 symantec PC-anywhere를 사용하였고 병렬 알고리즘은 MS Visual C++ 6.0을 이용

하여 구현하였다.

3.2 병렬 유전 알고리즘-타부 탐색법

본 논문에서는 유전 알고리즘의 전역 최적해 탐색능력과 타부 탐색법의 빠른 수렴특성을 이용한 향상된 최적해 탐색기법을 제안하였고, 이를 병렬 처리 기법인 PC 클러스터 시스템으로 개발하여 최적해 탐색성능 및 탐색속도를 개선하였다. 제안한 방법의 유전 알고리즘 수행 node와 타부 탐색법 수행 node 간의 연결 구조를 그림 2에 나타내었다.

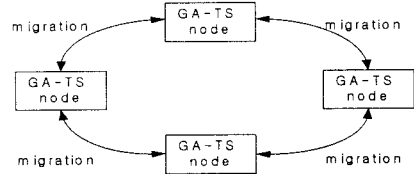


그림 2 유전 알고리즘과 타부 탐색법의 연결구조
Fig. 2 Connection structure of GA and TS

제안한 병렬 유전 알고리즘-타부 탐색법은 유전 알고리즘 연산에 의한 전역 최적해 탐색과정, 유전 알고리즘 연산시 스트링 반복여부 검토과정, 유전 알고리즘 스트링 중 일부에 타부 탐색법의 연산을 적용한 최적해 탐색성능 향상과정, 각 node간의 이주과정으로 구성되어 있으며, 흐름도를 그림 3에 나타내었다.

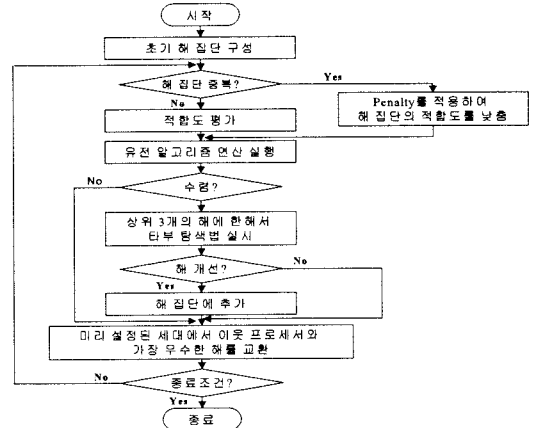


그림 3 제안한 알고리즘의 흐름도
Fig. 3 Flowchart of the proposed method

- (1) 유전 알고리즘 연산에 의한 전역 최적해 탐색과정 : 각 node에 대해 유전 알고리즘의 해집단을 분할하여 할당함으로써 탐색속도 및 탐색성능을 향상시켰다.
- (2) 유전 알고리즘 연산시 매 세대별 평균적합도 이하의 스트링 반복여부 검토과정 : 유전 알고리즘 연산 수행시에 이전에 탐색한 평균적합도 이하의 스트링이 반복되지 않도록 매 세대마다 유전 알고리즘의 스트링 중 평균적합도 이하의 스트링을 일정 반복회수동안 저장한 후 반복되는 스트링에 별점을 부과하여 적합도를 낮춤으로써 유전 알고리즘의 전역최적해 탐색성능을 강화시켰다
- (3) 유전 알고리즘 스트링 중 일부에 타부 탐색법의 연산을 적용한 최적해 탐색속도 향상과정 : 과정 (1),

(2)의 유전 알고리즘 연산에서 구한 해가 일정 세대 동안 개선되지 않는 경우, 현재 세대의 해집단 중 우수한 해의 주변영역을 집중적으로 탐색하기 위해 유전 알고리즘의 해집단 중 상위 3개의 해집단을 타부 탐색법의 현재 해로 설정하여 타부 탐색법을 일정 반복회수동안 진행함으로써 전체 알고리즘의 최적해 탐색성능 및 탐색속도를 개선하였다.

- (4) 각 node간의 이주과정 : 각 유전 알고리즘-타부 탐색법 node를 링 형태로 연결한 후, 이주연산을 통해 일정 간격으로 인접 node에 우수한 해를 상호 전송함으로써 최적해 탐색성능을 향상시킬 수 있다.

3.3 병렬 유전 알고리즘-타부탐색법을 이용한 최적조류계산

최적조류계산 문제는 전력계통운용시 최적의 발전기의 유효전력 출력, 발전기의 전압 크기, 변압기의 변압비등을 결정해야 하는 조합최적화 문제이다. 유전 알고리즘의 부호화과정에서는 각각의 제어변수에 요구되는 정확도에 따라 다른 크기의 염색체를 사용하여 스트링을 부호화하였으며, 제어변수의 최대 및 최소 한계 제약조건을 만족하도록 디코딩하였다.

또한 본 논문에서 사용한 적합도 함수를 식 (5)에 나타내었다. 식 (5)에 나타난 바와 같이 발전기 연료비용의 합이 작아질수록 적합도 값이 커지도록 적합도 함수를 설정하였다. 그리고 최적조류계산에서 운용상의 제약조건들을 적합도 함수에 페널티 항을 두어 제약조건을 위배 정도에 따라 벌점을 부과하였다.

$$FF = \frac{A}{\sum_{i=1}^{N_G} F_i(P_{Gi}) + \sum_{j=1}^{N_C} w_j \cdot Pen_j} \quad (5)$$

여기서,

FF : 적합도 함수

A : 상수

w_j : j번째 계통 운용과 관련된 제약조건 of the 가중치

F_i(P_{Gi}) : i번째 발전기의 연료비용 함수

Pen_j : j번째 계통 운용과 관련된 제약조건의 페널티 함수

N_G : 발전기의 대수

N_C : 계통 운용과 관련된 제약조건의 개수

식 (5)의 적합도 함수를 사용한 후보해의 평가 과정은 다음과 같다.

- (1) 최대 및 최소 제약 조건을 만족하도록 제어변수 z 를 디코딩한다.
- (2) 상태 함수 x 를 계산하기 위해 조류계산을 한다.
- (3) 계통 운용상의 제약조건 위반 여부에 따라 벌점을 부과한다.
- (4) 식 (5)을 사용하여 적합도를 계산한다.

그리고 유전 알고리즘의 복제 과정에서는 위에서 언급한 적합도에 비례하여 복제하는 룰렛휠 방법을 사용하였고, 교배 및 돌연변이 연산을 적용한 후 적합도가 가장 높은 개체를 다음 세대로 복제하는 엘리티즘을 적용하였다.

4. 사례 연구

4.1 IEEE 30 모선에 대한 사례연구

본 논문에서 제안한 방법의 유용성을 입증하기 위해 IEEE 30 모선 계통을 Pentium IV 2.0 GHz를 사용하여 모의하였다. 그리고 모의 계통은 6개의 발전기, 21개의 부하 모선, 41개의 분기로 구성되어 있다. 최적조류계산의 목적함수는 총 유효발전 비용을 최소화하도록 설정하였으며, 제어변수로 5개 발전기의 유효전력 출력량, 6개 발전 모선의 전압 크기, 4개 변압기 탭 비를 사용하였다. 그리고 최적조류계산시 유전 알고리즘 및 타부 탐색법에 사용된 시뮬레이션 계수를 표 1에 나타내었다.

표 1 시뮬레이션 계수

Table 1 Simulation coefficients

계 수	유전 알고리즘				타부 탐색법	
	세대 수	해 집단 수	교배 확률	돌연변이 확률	반복회수	타부 리스트 길이
설정값	100	32	0.9	0.1	3	3

표 2에 제안한 알고리즘에 의해 구해진 변압기 탭 설정 및 탭 비를 나타내었고, 표 3에 구해진 발전기 모선의 전압크기, 유효전력 출력량 및 발전 비용을 나타내었다.

표 2 변압기 탭 설정

Table 2 Transformer tap settings

변압기번호	1(4-12)	2(6-9)	3(6-10)	4(28-27)
변압기 탭비	0.927	0.947	1.000	0.991

표 3 발전기 모선의 전압 크기, 유효전력 출력량 및 발전 비용

Table 3 Generator-bus voltage magnitude, unit real power output, and cost

Bus Number	전압 (p.u.)	P _G (MW)	Cost (\$/hr)
1	1.050	173.36	459.43
2	1.031	49.11	128.13
5	1.000	20.95	48.38
8	1.014	20.98	71.86
11	1.017	13.65	45.61
13	1.050	12.60	41.80
Total		290.65	795.21
Total Losses		7.25	

제안한 알고리즘에 의해 구한 발전비용은 795.21\$/hr로 제안한 방법이 기존의 방법[2,8,9]보다 우수한 해를 구하였으며, 최적해 탐색시간은 14.76초로 비교적 짧은 시간내에 최적해를 구할 수 있었다.

그림 4에 IEEE 30 모선 계통에 대한 각 방법의 세대 별 발전 비용추이를 나타내었다. 그림 4에 나타난 바와 같이 유전 알고리즘-타부 탐색법이 기존의 유전 알고리즘보다 최적해로의 수렴속도가 빠름을 확인할 수 있었다. 따라서 본 논문에서 제안한 유전 알고리즘과 타부 탐색법의 특성을 적절히 조합하는 방법이 최적해 탐색능력이 더 우수함을 확인하였고, 이를 병렬로 연산함으로써 더 빠른 시간에 더욱 우수한 해를 구할 수 있었다.

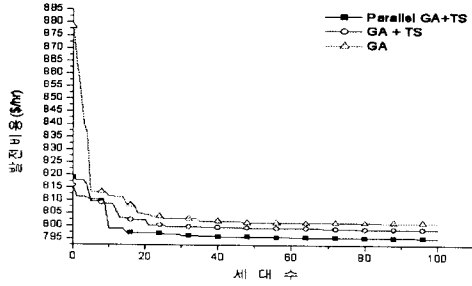


그림 4 GA, GA+TS 및 제안한 방법의 세대별 발전비용곡선
 Fig. 4 Generation cost VS generation curves for GA, GA+TS and the proposed method

또한 본 논문에서는 제안한 알고리즘의 병렬 계산 효과를 보이기 위하여 일반적으로 잘 알려진 아래의 두 성능 지수를 사용하여 평가하였다.

(1) 속도 향상률 (Speedup : S_p)

$$S_p = \frac{T}{T_p} \quad (6)$$

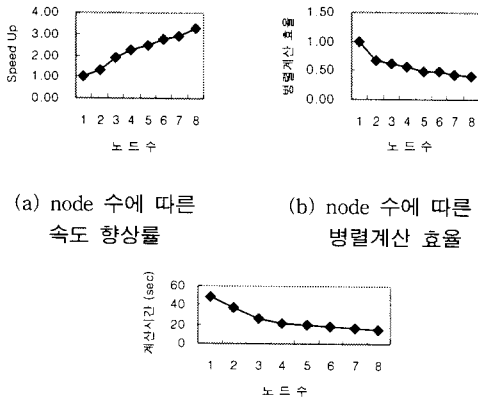
여기서, T : 프로세서 한 대를 사용한 경우의 실행시간
 T_p : P 대의 프로세서를 사용한 경우의 실행시간

(2) 병렬계산의 효율성

$$E_p = \frac{S_p}{P} \quad (7)$$

여기서, P : 사용한 프로세서의 수

node 수에 따른 속도 향상률, 병렬계산 효율 및 계산 시간을 그림 5에 나타내었다.



(a) node 수에 따른 속도 향상률
 (b) node 수에 따른 병렬계산 효율
 (c) node 수에 따른 최적해 탐색시간

그림 5 node 수 증가에 따른 속도 향상률, 병렬계산효율 및 최적해 탐색시간
 Fig. 5 Speedup, efficiency, and computation time according to the node number

그림 5에 보이는 것처럼 node의 수를 증가시킴에 따라 똑같은 최적해를 구할 때 탐색 시간이 단축됨을 확인

하였다. 속도 향상률은 node 수와 거의 선형적으로 비례하여 향상되었으나, node 수의 증가에 따라 다소 저하되는 경우가 발생하였는데, 이는 여러 node들 간의 상호통신 시에 발생할 수 있는 병목현상과 유전 알고리즘 평가과정에서 소요되는 계산시간 때문으로 생각된다. 그러나 속도 향상률이 꾸준히 증가하므로 더욱 더 복잡한 계통을 PC 클러스터 시스템으로 병렬처리를 한다면 최적해 탐색시간을 크게 줄일 것으로 생각된다.

5. 결 론

본 논문에서는 최적조류계산 문제의 최적해 탐색성능 및 탐색속도를 개선하기 위해서 PC 클러스터 시스템을 이용하였고, 새로운 병렬 유전 알고리즘-타부 탐색법을 제안하였다. 제안한 방법의 유용성을 입증하기 위해 제안한 방법을 IEEE 30 모선 최적조류계산 문제에 적용하였다. 그리고 종래의 유전 알고리즘과 타부 목록을 가진 유전 알고리즘과 비교하였다. 그 결과, 제안한 방법이 종래의 방법에 비해 손실을 줄이면서 발전비용을 감소시킴을 확인할 수 있었다. 또한 제안한 알고리즘을 병렬처리 함으로서 다른 알고리즘보다 빠른 시간 내에 최적해를 탐색할 수 있었으며, 병렬처리 node가 증가할수록 최적해 탐색시간이 단축됨을 확인할 수 있었다. 따라서 본 논문에서 모의한 계통보다 보다 대규모의 계통에서 본 논문에서 제안한 알고리즘을 검증한다면 실제계의 실시간 제어 및 최적 운용에 크게 기여할 수 있을 것으로 기대된다.

[참 고 문 헌]

- [1] Hermann W. Dommel and William F. Tinney, "Optimal Power Flow Solutions," IEEE Trans. on Power Apparatus and Systems, Vol. PAS-87, pp. 1886-1876, October 1968.
- [2] O. Alsac and B. Stott, "Optimal Load Flow with Steady State Security," IEEE Trans. on Power Apparatus and Systems, Vol. PAS-93, pp. 745-754, 1974.
- [3] O. Alsac, J. Bright, M. Prais, and B. Stott, "Further Developments In LP-Based Optimal Power Flow," IEEE Trans. on Power Systems, Vol.5, pp.697-711, August 1990.
- [4] G. F. Reid and L. Hasdorf, "Economic Dispatch Using Quadratic Programming," IEEE Trans. on Power apparatus and Systems, Vol. PAS-92, pp. 2015-2023, 1973.
- [5] A. G. Bakirtzis, P. N. Biskas, C. E. Zoumas, and V. Petridis, "Optimal Power Flow by Enhanced Genetic Algorithm," IEEE Trans. on Power Systems, Vol. 17, No. 2, pp. 229-236, May 2002.
- [6] L. L. Lai, J. T. Ma, R. Yokoyama, and M. Zhao, "Improved genetic algorithms for optimal power flow under both normal and contingent operation states," Elec. Power Energy Syst., Vol. 19, No. 5, pp. 287-292, 1997.
- [7] C. A. Roa-Sepulveda and B. J. Pavez-Lazo, "A Solution to the optimal power flow using simulated annealing," PPT 2001 IEEE Porto Power Tech Conference, Sep. 2001
- [8] T. Kulworawanchpong, S. Sujitjorn, "Optimal Power Flow Using Tabu Search," IEEE Power Engineering Review, June 2002.
- [9] Y. Huang, T. Kashiwagi, and S. Morozumi, "A parallel OPF approach for large-scale power systems," IEE Power System Management and Control, 17-19 April 2002 Conference Publication No.488
- [10] H. Mori and K. Takeda, "Parallel simulated annealing for power system decomposition," IEEE Proc. of PICA 93, Scottsdale, Arizona, pp. 366-372, May 1993.
- [11] J.-Y. Choi, J.-R. Shin, and M.-H. Kim, "Parallel computing environment based on windows operating system," 한국항공우주학회지, May 2002.