

Delaunay Triangulation의 폴리간 검색속도 개선을 위한 T-Search와 Dynamic-Window 개념의 결합

강현주^{*} · 윤석준[†] · 공지영^{**} · 김강수^{**}
(2002년 3월 5일 접수, 2003년 3월 24일 심사완료)

Integration of T-Search and Dynamic-Window Concept for Accelerated Searching Speed in Delaunay Triangulation

Hyun-joo Kang, Sugjoon Yoon, Ji-Young Kong and Kang-soo Kim

Key Words: Delaunay Triangulation(디로네이 삼각형), T-search(삼각형 검색), Dynamic-Window(동적창), Real-time Simulation(실시간 시뮬레이션)

Abstract

Terrain surfaces have to be modeled in very detail and wheel-surface contacting geometry must be well defined in order to obtain proper ground-reaction and friction forces for realistic simulation of off-road vehicles. Delaunay triangulation is one of the most widely used methods in modeling 3-dimensional terrain surfaces, and the T-search is a relevant algorithm for searching resulting triangular polygons. The T-search method searches polygons in a successive order and may not allow real-time computation of off-road vehicle dynamics if the terrain is modeled with many polygons, depending on the computer performance used in the simulation. The dynamic T-search, which is proposed in this paper, combines conventional T-search and the concept of the dynamic-window search which uses reduced searching windows or sets of triangular surface polygons at each frame by taking advantage of the information regarding dynamic characteristics of a simulated vehicle. Numerical tests show improvement of searching speeds by about 5% for randomly distributed triangles. For continuous search following a vehicle path, which occurs in actual vehicle simulation, the searching speed becomes 4 times faster.

1. 서 론

자동차나 군용 전차와 같이 지표면 위를 움직이는 동체의 운동을 발생시키는 근본적인 에너지는 연료와 그 연료를 이용한 엔진장치이다. 하지만, 차량이 이동할 수 있는 최종적인 근거는 차륜과 지면 사이에 이루어지는 작용 및 반작용이다. 즉, 차량의 운동을 엄밀하게 시뮬레이션하기 위해서는 지표 반발력과 마찰력을 정확히 산출할 수 있어야 하는데, 이를 위해서는 지형과 지표면의

정밀한 모델링이 차량 자체의 모델링 못지 않게 중요하게 대두된다. 특히, 차량이 비 포장 도로나 들판과 산악지형을 주행할 경우, 지표면은 평평한 형태를 갖고 있지 않기 때문에 지표면과 동체가 맞닿은 면의 정보 즉, 경사도와 그에 따른 지표반발력이 동체의 움직임이나 반동을 정확히 표현하기 위해 매우 중요하다 할 것이다.

보통의 지형 데이터는 직교형 데이터로 한 점을 데이터에 추가하고자 할 때, 부수적으로 수많은 점들이 자동적으로 데이터에 추가된다. 따라서 정밀한 데이터를 표현하기 위해서는 데이터 테이블의 크기가 커지는 것을 피할 수 없다. Delaunay triangulation 기법^(1,2)을 사용하게 되면, 보다 적은 데이터를 사용하여 지형의 특징을 보다 근사하게 표현할 수 있다. 따라서 일반적인 좌표축 분할 방법에 비해 적은 양의 데이터로 정밀한 지형을 표현할 수 있게 된다.

* (주)시스코프

† 책임저자, 회원, 세종대학교 기계항공우주공학부

E-mail : sjyoon@sejong.ac.kr

TEL : (02)3408-3283 FAX : (02)3408-3333

** 세종대학교 대학원 항공우주공학과

Delaunay 방법으로 형성된 지형 데이터 베이스에서 원하는 삼각형을 찾기 위해서 일반적으로 사용하는 검색방법이 T-search(Triangle search)이다.^(3,4) T-search는 Delaunay 방법으로 삼각형을 형성하는 과정에서 결정된 초기 삼각형에서부터 검색하고자 하는 삼각형의 방향으로 순차적으로 삼각형을 이동하면서 원하는 삼각형을 검색하는 알고리즘이다. T-search의 순차검색은 방대한 데이터 베이스를 대상으로 검색을 수행할 경우, 일반 컴퓨터 환경 내에서 실시간 시뮬레이션을 보장할 만큼 빠른 검색속도를 얻기 힘들다. 본 논문에서 새로이 제시된 dynamic T-search는 주어진 데이터 테이블에서 검색 대상이 되는 영역을 차량과 같은 동적 개체의 동 특성 정보를 토대로 축소함으로써 검색 속도를 향상시키는 검색 창 검색법(dynamic-window search)^(5,6)에 기존의 T-search를 결합시키는 방식이다. 검색에 전적으로 소요된 시간만을 비교하는 수치시험결과를 검토해 보면, 불연속 데이터를 검색하는 지형 데이터베이스 내의 모든 데이터 검색의 경우에도 5% 정도의 속도개선 효과가 나타났으며, 차량의 이동에 따른 곡선경로 상의 연속적 데이터 검색의 경우는 검색시간이 무려 1/4로 단축되는 효과를 확인할 수 있었다.

2. 3D 지표면 모델링

3D 지형을 표현하는 방법을 생각해 보자. 가장 평범하게 쓰일 수 있는 방법은 수평 좌표축과 수직 좌표축을 분할하여 그 교차지점의 데이터를 이용하여 표현하는 방법이다. 지표면의 형상과 사용 목적에 따라서는 다각형을 이용하여 모델링하는 것이 보다 효과적일 수도 있다. 하지만, 사각형이나 그 이상의 다각형으로 이루어지는 단위 폴리건들은 각 꼭지점의 통상적으로 다른 고도 정보 때문에 평면을 보장할 수 없다. 평면을 보장할 수 있는 최소 단위인 삼각형으로 지형을 표현하기 위하여 좌표축을 분할하여 데이터를 형성하는 방법은 원하는 데이터를 추가하기 위해 많은 양의 부수적인 데이터들이 요구된다. 하지만, Delaunay triangulation 방법을 사용하게 되면 Fig. 1과 같이 보다 적은 데이터를 사용하여 지형의 특징을 보다 근사하게 표현할 수 있다. 따라서 일반적인 좌표축 분할 방법에 비해 적은 양의 데

이터로 정밀한 지형을 표현할 수 있게 된다.

Delaunay triangulation 방법은 데이터가 입력될 때마다 기존의 데이터 point들과 연결하여, Fig. 2와 같이 각 삼각형의 내접이 다른 point들이 포함되지 않도록 삼각형을 형성한다. 이렇게 삼각형을 형성함으로써 지형의 본래 형태가 그대로 유지되는 것이다.

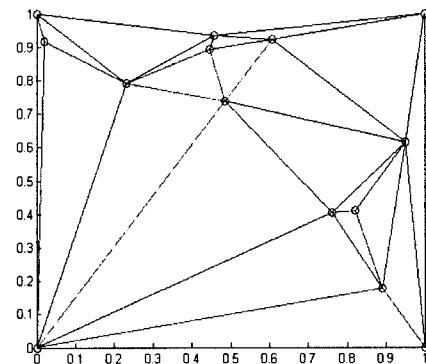


Fig. 1 Delaunay triangulation map

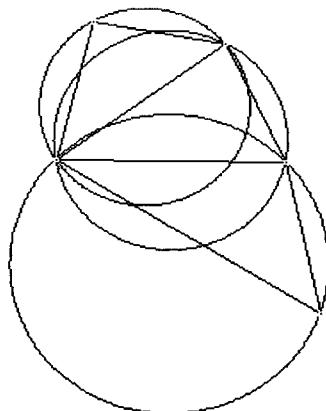


Fig. 2 Delaunay triangulation and Inscribed circles

3.1 T-search 기법

앞에서 적은 양의 데이터를 이용하여 3차원의 지형을 정확하게 표현하기 위한 방법으로 Delaunay triangulation을 소개하였다. Delaunay 방법으로 형성된 지형 데이터 베이스에서 원하는 삼각형(동체의 바퀴가 위치한 지역을 포함하는 삼각형)을 찾기 위해서 일반적으로 사용하는 검색방법이 T-search(Triangle search)이다. T-search는 Delaunay 방법으로 삼각형을 형성하는 과정에서 결정된 초기 삼각형에서부터 검색하고자 하는 삼각형의 방향으로 순차적으로 삼각형을 이동하면서 원하는 삼각형을 검색하는 알고리즘이다. Fig. 3은 위에서 제시한 Delaunay triangulation 방법으로 형성된 데이터 테이블에서 T-search 방법을 이용하여 검색했을 때의 검색 경로를 나타내는 것이다. 1번 삼각형이 초기 삼각형이고 × 표시가 동체가 맞닿은 면이라면, 다음의 그림처럼 초기 삼각형에서 5번 이동하여 검색하고자 하는 삼각형을 찾게 된다. 따라서 검색과정에서 초기 삼각형의 위치와 검색 대상이 되는 삼각형 사이에 거치게 되는 삼각형들의 개수가 검색에 소요되는 시간을 결정하게 되고, worst case로 검색속도를 결정하게 되는 실시간 시뮬레이션의 조건을 고려할 때에는 검색 대상이 되는 데이터 베이스의 크기(전체 지형을 구성하는 삼각형의 총 개수)가 검색속도를 좌우하게 된다. T-search 알고리즘은 다음과 같다.

T-search 알고리즘

1. 검색대상이 되는 point가 초기 삼각형 내부에 존재하는지 체크한다.
2. 초기검색 삼각형에 접해있는 삼각형들과 검색대상이 되는 point 사이의 거리 계산한다.
(※ 삼각형의 무게중심과 point의 거리)
3. 계산된 거리 중 가장 짧은 삼각형으로 초기 삼각형으로 수정
4. 1.~ 3. 과정을 반복한다.

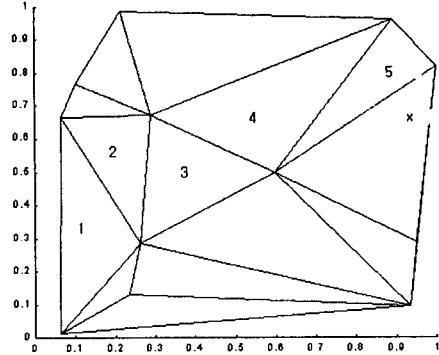


Fig. 3 An example of search sequence in a Delaunay triangulated map

실시간 시뮬레이션에서는 고정된 타임 프레임에서 검색을 완료해야하므로 검색 알고리즘의 속도를 정의하고자 할 때, 기준이 되는 것이 worst case의 속도이다.^(7,8) T-search의 순차검색은 방대한 데이터 베이스를 대상으로 검색을 수행할 경우, 일반 컴퓨터 환경 내에서 실시간 시뮬레이션을 보장할 만큼 빠른 검색속도를 얻기 힘들다. 따라서, 다음에 소개하는 새로운 검색기법을 접목하여 처리속도의 개선을 시도하게 되었다.

3.2 Dynamic-window search

실시간 시뮬레이션에서 시간적 제약을 만족시키기 위한 방법 중 하나로 주어진 데이터 테이블에서 검색 대상이 되는 영역을 차량과 같은 동적 개체의 동 특성 정보를 토대로 축소함으로써 검색 속도를 향상시키는 검색 창 검색법(dynamic-window search)이 제시되었다[5, 6]. 검색하고자 하는 값이 시스템의 상태변수, 적분 기법, 적분 프레임 크기 등과 연관되어 임의의 원칙 즉, dynamics를 갖고 매 프레임마다 변한다면, 이 정보를 활용하여 주어진 데이터 테이블 중 일부의 영역으로 검색 대상을 축소할 수 있다는 것이 dynamic window의 개념이다. 즉, 데이터가 이전 프레임에서의 위치 주변에서 이동하기 때문에 데이터의 위치를 찾기 위하여 데이터베이스 전체를 검색할 필요가 없이, 바로 직전의 검색 결과를 기반으로 검색 대상을 축소할 수 있다는 것이다. 데이터의 초기 위치는 실시간 시뮬레이션

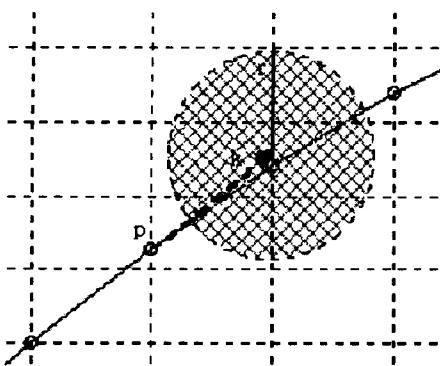


Fig. 4 Dynamic-window search

시작 이전에 추출이 가능하며, 이후의 데이터 위치는 시스템의 dynamics와 시뮬레이션에서 사용되는 단위 프레임의 크기에 근거하여 이동하게 된다.

예를 들어, 데이터 테이블이 주어지고 P 가 검색된 point라고 가정하자. Fig. 4에서와 같이 P 에서의 속도 및 방향을 기준으로 다음 검색결과를 포함하리라 예상되는 테이블의 부분집합 즉 검색 창의 중심 k 와 검색창의 반경 r 을 구할 수 있다. 따라서 P 를 검색한 직후의 프레임에서는 검색 결과를 구하기 위하여 테이블 전체를 검색하는 것이 아니라 k 를 중심으로 일정한 반경 r 을 갖는 창을 검색의 대상으로 하게 된다.

3.3 Dynamic-window 개념과 T-search의 결합

Dynamic-window search의 개념을 그대로 T-search와 결합시키면, ① 검색된 결과를 기반으로 다음에 검색 결과를 포함할 것이라 예상되는 삼각형을 검색하고, ② 검색된 삼각형을 중심으로 검색 대상이 될 삼각형 집합(검색창)을 형성해야 한다. 그러나 동체의 움직임을 정확히 예측할 수 있다고 가정한다면, delaunay triangulation 방법으로 형성된 map을 대상으로 검색 결과를 포함할 것이라 예상되는 삼각형을 검색하는 데 소요되는 시간이 검색결과를 포함하는 삼각형을 검색하는 데 소요되는 시간과 비슷하다. 따라서 예상 삼각형을 검색하는 시간에 이를 중심으로 검색창을 형성하는 시간이 더해진다면 기존의 T-search를 이용해 검색하는 것보다 더 많은 시간이 소요될 것이 예상된다. 따라서 T-search 알고리즘에 dynamic window 개념 중 검색결과를 바로

다음 검색에 재활용한다는 개념만 도입하는 것이 더 바람직하다.

앞서 설명한 바와 같이 기존의 T-search 알고리즘은 모든 검색에 대해 동일한 초기 삼각형에서 검색을 시작하게 된다. 동체가 일정한 dynamics를 가지고 운동한다면, 이 정보를 이용하여 검색이 시작되는 초기 삼각형을 매 프레임 바꾸어 줌으로서 검색속도를 향상시킬 수 있다. Dynamic-window 개념을 도입한 방법과 기존의 방법을 Fig. 5에서 두 가지 ‘●’를 검색하는 예로 비교하면 다음과 같이 나타낼 수 있다. 그림에서 ○는 동체의 이동경로를 나타낸다.

기존의 방법이 검색을 수행하는 과정은 점선의 얇은 화살표로 표시된 것이고, dynamic-window 개념을 도입한 경우는 실선의 두꺼운 화살표로 표시된 경로를 따라 검색을 수행한다. 첫 번째 경우는 검색을 시작할 때의 검색경로를 나타내고 있다. T-search 알고리즘은 초기 삼각형(1번 삼각형)에서 검색을 시작하고 Dynamic-window 개념을 도입한 방법의 초기삼각형은 5번 삼각형에서 검색을 시작하게 되어 결과적으로 기존의 방법은 5번만에 동체가 위치한 point를 검색하고, dynamic-window 개념을 도입한 경우에는 3번만에 point를 검색하게 된다. 두 번째는 계속적으로 검색을 수행하는 경우에서 동체가 위치한 삼각형의 검색을 보여주고 있다. T-search 알고리즘은 항상 동일한 초기 삼각형(1번 삼각형)에서 검색을 시작하고 Dynamic-window 개념을 도입한 방법은 전 단계에 동체가 위치했던 삼각형을 초기삼각형(9번 삼각형)으로 검색을 수행한다. 그 결과 기존의 방법은 9번만에 동체가 위치한 point를 검색하고, dynamic-window 개념을 도입한 경우에는 1번만에 point를 검색하게 된다.

4. 수치시험

앞서 설명한 dynamic-window 개념의 도입 시 검색속도의 향상을 다음의 수치시험을 통해 검증해보았다. 수치시험에 사용된 데이터는 다음의 Fig. 6처럼 각각 50개의 난수(random number)를 X와 Y, Z 좌표로 생성하여 Delaunay triangulation 방법을 이용하여 형성하였다. 수치비교는 이 지형 데이터베이스에 대하여 두 가지 조건 하에서 수행되었다. 첫 번째 시험조건은 임의의 순서로

지형 데이터베이스 내의 모든 삼각형을 검색하는 경우이며, 두 번째 조건은 Fig. 7에서처럼 차량의 경로를 사인함수와 같이 가정하여 기하학적으로 연속된 삼각형들을 순차적으로 검색하는 것이다.

다음의 수치시험들을 수행하는데 사용된 컴퓨터의 주요 사양은 다음과 같다:

- CPU : AMD Duron 1.0 G
- RAM : DDR 256MB
- MainBoard: Soltek SL75DRV

T-search와 dynamic window 개념이 도입된 수정된 T-search 기법을 비교하기 위해, Matlab 6.0에 toolbox의 형태로 제공되는 tsearch 함수가 이용되었다. 이 함수는 tsearch.m 파일에 정의되어 있으며, tsrchmx.dll이라는 s-function을 호출한다. tsrchmx.dll은 win32 platform에서 구동되는 Matlab 6.0을 위한 s-function 형태의 binary 파일로서 수치 비교를 위해 그의 소스 파일인 tsrchmx.c을 수정하였다.

두 가지 검색기법들의 검색소요시간 비교를 위해 처음으로 시도된 방법은 dynamic T-search 기법을 적용할 수 있도록 tsrchmx.c에서 I/O에 관한 부분을 수정하는 것이었다. 두 가지 검색기법들의 검색 소요시간 비교를 위해서 Matlab 6.0의 내장 함수인 tic과 toc 사이에 100,000번의 반복된 검색을 수행하도록 하였으며, Table 1과 같은 시험 결과를 얻었다.

Table 1의 수치시험 결과를 검토해 보면 곡선 경로의 경우 두 검색기법들 사이에 상당한 검색 속도 차이가 존재함을 알 수 있으나, 불연속적으로 놓인 모든 삼각형 데이터의 검색에서는 별 차이가 드러나지 않는다. 즉, dynamic window 개념이 도입된 새로운 dynamic T-search 기법은 차량의 주행 경로를 따라 검색하게 되는 기하학적으로 연결된 삼각형들의 검색에서 보다 효과적임을 알게 되었다.

위의 실험에서 s-function structure의 데이터 이동에 따른 소요시간과 같이, 검색 알고리즘과는 직접 상관없는 컴퓨터 상의 여타 프로세스 처리 부분에서 소요된 시간을 최대한 배제하고, 더욱 정확한 검색소요시간 비교를 위하여 다음과 같이 2차 수치시험을 수행하였다.

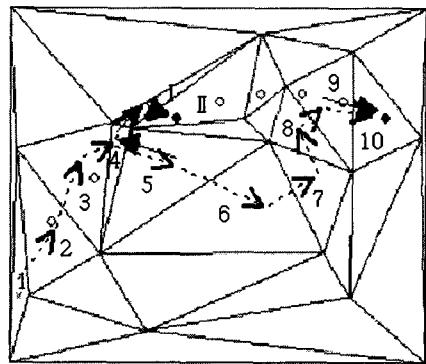


Fig. 5 Comparison of Dynamic T-search and original T-search

1차 실험에서 수정한 파일인 tsrchmx.c의 내부에 현재의 CPU 소요시간을 계산하는 함수인 clock()을 이용하여, 알고리즘 연산부의 시작과 끝 부분의 CPU 타임을 읽어들여 이들의 차를 이용하여 소요시간을 측정하는 알고리즘을 추가하였다. 하지만, 이 경우 알고리즘 연산부의 소요시간이 너무 짧아서 1/1000(sec)의 해상도를 가지는 clock() 함수를 이용한 측정은 불가능하였다. 따라서, clock() 함수 사이에 1,000번의 반복적 연산을 수행하도록 추가로 수정하여 총 100,000번의 반복 연산을 수행하면서, 각각의 소요시간을 합산하여 총 소요시간을 추출하였다. 이 수치시험 결과는 Table 2와 같다.

검색에 전적으로 소요된 시간만을 비교하는 2차 수치시험 결과를 검토해 보면, dynamic T-search의 시간단축 효과는 더욱 두드러진다. 불연속 데이터를 검색하는 모든 데이터 검색의 경우에도 5% 정도의 속도개선 효과가 나타났으며, 차량의 이동에 따른 곡선경로 상의 연속적 데이터 검색의 경우는 검색시간이 무려 1/4로 단축되는 효과를 확인하였다.

비록 제한된 수치시험을 통하여 이러한 결론을 도출하였지만, 어떠한 임의의 차량 경로도 삼각 함수의 조합으로 표현할 수 있으므로, 위의 실험 결과와 dynamic T-search의 시간단축 효과는 임의의 차량 경로에서 그대로 유효하다고 할 수 있다.

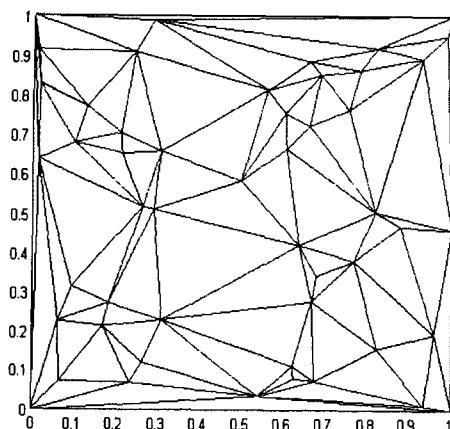


Fig. 6 A terrain database modeled by Delaunay Triangulation

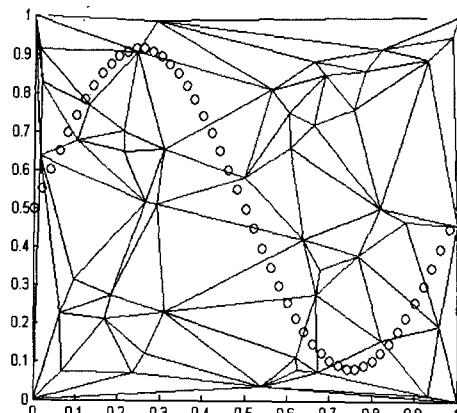


Fig. 7 The wheel track of a moving vehicle

Table 1 1st Comparison of searching times between T-search and Dynamic T-search (sec)

	횟수	Tsearch	D-W 기법 도입
모든 데이터 (불연속)	1	37.7440	37.1100
	2	37.4400	36.2720
	3	37.7140	36.352
곡선 경로 (연속)	1	45.8960	14.3110
	2	45.8660	14.3610
	3	45.8560	14.3660

Table 2 2nd Comparison of searching times between T-search and Dynamic T-search (sec)

	횟수	Tsearch	D-W 기법 도입
(불연속)	1	33.2370	31.8560
	2	33.4270	31.8560
	3	33.1060	31.9400
(연속)	1	41.2400	10.2280
	2	41.4540	10.3750
	3	41.1450	10.1090

5. 결 론

Delaunay triangulation 방법으로 Off-road 차량이 주행할 3차원 데이터베이스를 형성하였을 때, 주로 사용되는 검색 기법은 T-search 방법이다. 그러나 T-search 방법은 순차적 검색기법이므로 실시간의 제약을 만족시키기 어렵다. 이에 dynamic window 개념을 도입한 새로운 검색방법을 시도하였다. 수치시험 결과, 검색에 소요되는 시간이 기존 T-search 방법보다 개선되었는데, 특히 차량의 이동에 따른 곡선경로 상의 연속적 데이터 검색의 경우는 검색시간이 무려 1/4로 단축되는 효과를 확인하였다.

후 기

본 연구내용은 2000년도 한국학술진흥재단의 연구비(KRF-2000-003-E00071)와 대학중점연구소 연구비(Y00175) 지원으로 수행되었으며 지원에 감사드립니다.

참고문헌

- (1) Edwin Boender, 1994, "Reliable Delaunay-Based Mesh Generation and Mesh Improvement," *Communications in Numerical Methods in Engineering*, Wiley, Vol 10, pp. 773~783.
- (2) Baker, Timothy J., 1989, "Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delaunay

- Triangulation," *Engineering with Computers*, Springer-Verlag, Num 5, pp. 161~175.
- (3) Danne Hanselman, Bruce Littlefield, 1998, Mastering MATLAB5, Prentice-Hall, Inc.
- (4) <http://prosys.korea.ac.kr/~tclee/matlab/techdoc/ref/tsearch.html>
- (5) Yoon S., Kang J., 2000, "A Study on Searching Speeds of Data-Searching Algorithms for Real-Time Simulation," *The Korean Society for Aeronautical & Space Sciences*, Vol. 28 No. 1, pp. 126~132.
- (6) Yoon S., Kang J., 2000, "Dynamic-Window Search for Real-Time Simulation of Dynamic Systems," *The Korean Society for Aeronautical & Space Sciences*, Vol. 28 No. 4, pp. 86~92.
- (7) Howe, R. M., 1986, "Dynamics of Real-Time Digital Simulation," Applied Dynamics International.
- (8) Yoon S., 1990, "Real-Time Simulation of Constrained Dynamic Systems," Ph.D. Dissertation, Dept. of Aerospace Engineering, University of Michigan.