

발간등록번호

11-1360395-000469-01



국립기상연구소 지구환경시스템연구과

기술노트 NIMR-TN-2014-010

해수면 변화 관측자료를 이용한 지진해일 검출 알고리즘 개발

이 준 환 · 박 순 천
전 영 수 · 함 인 경 · 이 창 욱 · 조 은 영

2014. 11.

발간등록번호

11-1360395-000469-01



국립기상연구소 지구환경시스템연구과

기술노트 NIMR-TN-2014-010

해수면 변화 관측자료를 이용한 지진해일 검출 알고리즘 개발

이 준 환 · 박 순 천
전 영 수 · 함 인 경 · 이 창 욱 · 조 은 영

2014. 11.

차 례

1. 서론	1
2. KTDA v1.0	2
2.1 KTDA v1.0 구조	2
2.2 KTDA v1.0 작동 방법	3
2.3 포맷 변경	6
2.4 Spike 제거	9
2.5 결측 구간 파고 예측	12
2.6 DART 알고리즘	16
2.7 경사 알고리즘	20
2.8 Detide 알고리즘	25
3. KTDA.m 코드	28
부록	58
참고문헌	60

그림 차례

Figure 2.1.1. KTDA v1.0 구조	2
Figure 2.2.1. KTDA v1.0 실행 흐름도	5
Figure 2.3.1. 울릉도 해일파고계 데이터	6
Figure 2.3.2. 포맷 변경 결과 파일 데이터	8
Figure 2.3.3. 포맷 변경 시계열	8
Figure 2.4.1. Spike 제거 결과 파일 데이터	11
Figure 2.4.2. Spike 제거 시계열	11
Figure 2.5.1. EP 개념도	13
Figure 2.5.2. EP 흐름도	13
Figure 2.5.3. 결측 구간 파고 예측 결과 파일 데이터	15
Figure 2.5.4. 결측 구간 파고 예측 시계	15
Figure 2.6.1. DART 알고리즘 개념도 (Beltrami, 2008, Figure 1)	17
Figure 2.6.2. DART 알고리즘 수행 결과	19
Figure 2.7.1. 경사 알고리즘 개념도 (Bressan and Tinti, 2011, Figure 1)	21
Figure 2.7.2. 경사 알고리즘 수행 결과	24
Figure 2.8.1. Detide 알고리즘 수행 결과	27

표 차례

Table 2.2.1. KTDA v1.0 작동을 위한 파일	3
Table 2.2.2. 컨트롤 파일 (input.dat)을 구성하는 변수 그룹	4
Table 2.3.1. 포맷 변경 변수	7
Table 2.4.1. Spike 제거 변수	10
Table 2.5.1. 결측 구간 파고 예측 변수	14
Table 2.6.1. DART 알고리즘 변수	17
Table 2.7.1. 경사 알고리즘 변수	22
Table 2.8.1. Detide 알고리즘 변수	26

1. 서론

지진해일이란 해저에서 발생한 지진·화산폭발 등의 급격한 지각변동으로 발생한 해수의 긴 파동이 비정상적으로 높아져 해안가에 도달하는 현상을 말한다. 2004년 수마트라 지진해일, 2011년 동일본 지진해일 등에 의해 전 세계적으로 대규모 피해가 발생하였다. 우리나라에서도 1983년과 1993년에 일본 서쪽 해역에서 발생한 지진해일로 인해 인명과 재산 피해가 발생하였다.

지진해일을 대비하기 위하여 독일, 호주, 미국, 캐나다, 일본 등의 국가에서는 지진해일 감시 및 조기경보 시스템을 구축하였다. 기상청은 지진해일로부터 국민의 생명과 재산을 보호하기 위하여 울릉도 해일파고계 및 유관기관의 조위관측소 자료를 활용하고 있다. 그러나 현재 관측 자료를 단순 모니터링하는 실정이며, 지진해일 검출 정보 및 지진해일 파고 정보를 제공하지 못한다.

국립기상연구소에서는 지진해일 통보 및 조기 대응을 위한 기초 연구로써 해수면 변화 관측자료를 이용한 지진해일 검출 알고리즘 개발을 진행하였다. 본 기술노트에서는 연구를 통해 개발한 지진해일 검출 알고리즘인 KTDA v1.0 (Korea Tsunami Detection Algorithm) 모델의 원리와 사용법에 대하여 기술하였다.

2. KTDA v1.0

2절에서는 KTDA v1.0의 원리와 사용법을 울릉도 해일파고계 자료를 대상으로 기술하였다. KTDA v1.0의 변수명은 기울임체로 나타내었다.

2.1 KTDA v1.0 구조

KTDA v1.0의 구조는 Figure 2.1.1과 같다. 먼저 원본 데이터의 포맷을 KTDA v1.0에 적용 가능한 포맷으로 변경한다. 지진해일 검출 알고리즘의 정확도와 연산 속도 향상을 위하여 spike 제거 및 결측 구간 파고 예측을 수행한다. 그리고 파고 변화로부터 지진해일을 검출하는 DART 알고리즘과 경사 변화로부터 지진해일을 검출하는 경사 알고리즘을 통해 지진해일을 검출한다. 조화분해를 통해 조석 성분을 제거하여 지진해일의 파고 정보를 포출한다.

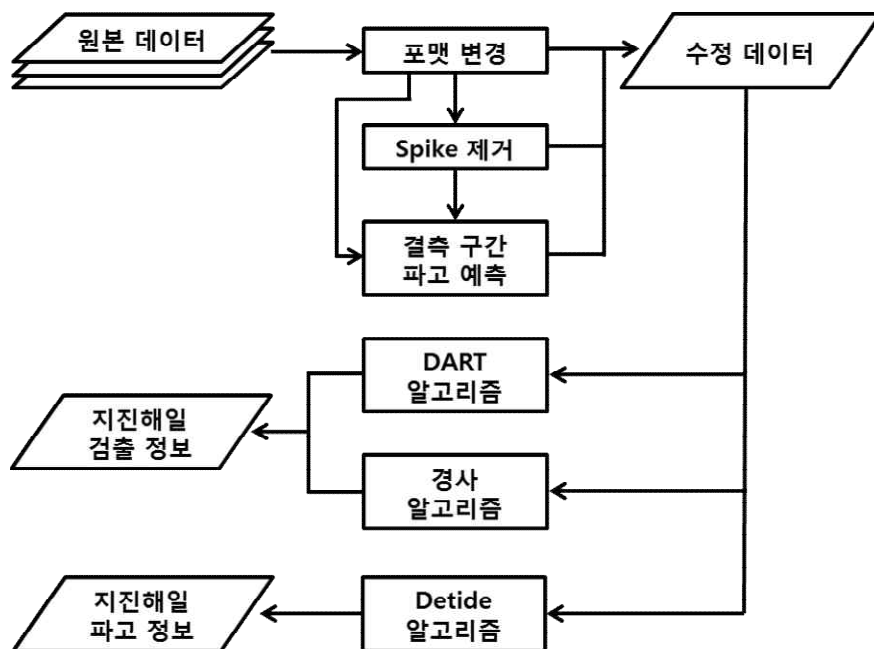


Figure 2.1.1. KTDA v1.0 구조

2.2 KTDA v1.0 작동 방법

○ 파일 준비

KTDA v1.0을 작동하기 위하여 Table 2.2.1의 파일을 한 폴더 안에 준비한다.

Table 2.2.1. KTDA v1.0 작동을 위한 파일

파일명	내용
yyyymmddhhmm.kan	울릉도 해일파고계 데이터 (예) 201103111230.kan
input.dat	KTDA v1.0 컨트롤 파일
KTDA.m	KTDA v1.0 MATLAB 메인 코드
saveascii.m	수행 결과를 ascii 형태로 저장하는 MATLAB 함수 다운로드: http://www.mathworks.com/matlabcentral/fileexchange/10243-saveascii-m-v6-2--oct-2009-
vlookup.m	EXCEL의 vlookup 함수와 동일한 기능을 하는 MATLAB 함수 다운로드: http://www.mathworks.com/matlabcentral/fileexchange/29233-vlookup-similar-to-ms-excel-function
gfit2.m	상관관계, 평균오차 등 통계 분석 기능의 MATLAB 함수 다운로드: http://www.mathworks.com/matlabcentral/fileexchange/22020-goodness-of-fit--modified-
T_TIDE package	조화분해를 통해 조석 성분을 예측하는 MATLAB 함수 및 데이터 시트 구성: t_equilib.dat, tide3.dat, t_astron.m, t_equilib.m, t_errors.m, t_getconsts.m, t_predic.m, t_readme.m, t_synth.m, t_tide.m, t_vuf.m, t_xtide.m, g.mat, t_18constituents.mat, t_constituents.mat, t_example.mat, t_xtide.mat 다운로드: http://www.eos.ubc.ca/~rich/#T_Tide

(참고) KTDA v1.0을 작동하기 위한 필수 파일은 울릉도 해일파고계 데이터 (.kan), 컨트롤 파일 (input.dat) 그리고 메인 코드 (KTDA.m)이다. 이 밖의 파일들은 컨트롤 파일 (input.dat)의 SWITCH 그룹 내 변수가 아래와 같이 설정되어 있는 경우에만 사용된다.

- saveascii.m : SWITCH 그룹 내 변수가 1개라도 2로 설정되어 있는 경우
- vlookup.m : *switch1*이 1 또는 2로 설정되어 있는 경우
- gfit2.m : *switch3*이 1 또는 2로 설정되어 있는 경우
- T_TIDE package : *switchtide*가 1 또는 2로 설정되어 있는 경우

○ input.dat 설정

KTDA v1.0 컨트롤 파일 (input.dat) 내 변수들을 설정한다. 컨트롤 파일을 구성하는 변수 그룹은 Table 2.2.2와 같으며 그룹 내 변수의 의미와 설정 방법은 2.3절에서 2.8절에 기술하였다.

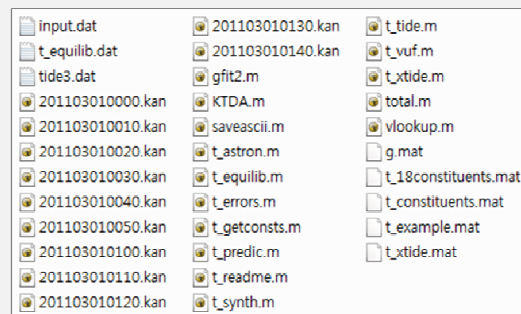
Table 2.2.2. 컨트롤 파일 (input.dat)을 구성하는 변수 그룹

줄번호	그룹명	내용
04~09	SWITCH	알고리즘 수행 및 저장 여부
13~14	BASIC INFORMATION	시계열 기본 정보
18~19	FORMAT CHANGE	포맷 변경
23~24	DESPIKE	Spike 제거
28~32	GAP FILLING	결측 구간 파고 예측
36~39	DART ALGORITHM	DART 알고리즘
43~51	SLOPE ALGORITHM	경사 알고리즘
55~57	DETIDE ALGORITHM	Detide 알고리즘
61~66	SAVE DATA	저장 데이터 파일 이름

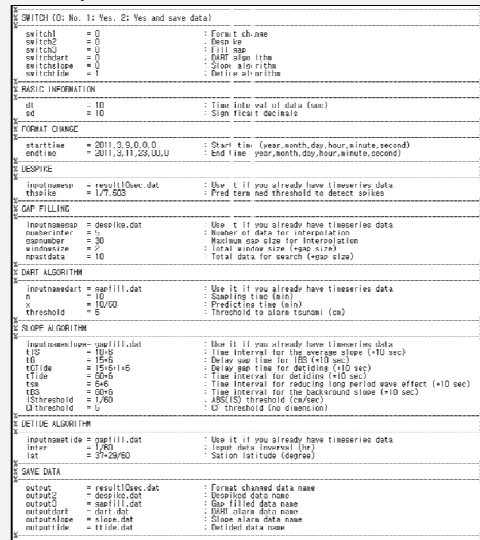
○ KTDa v1.0 실행

리눅스 운영체제의 컴퓨터에서 MATLAB을 실행한다. 파일 준비 단계의 폴더를 MATLAB 현 폴더 (Current Folder)로 지정한다. MATLAB 명령 창 (Command Window)에 KTDa를 입력한 후 엔터를 누르면 KTDa v1.0이 실행된다.

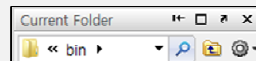
(1) 파일 준비



(2) input.dat 설정



(3) 폴더 지정



(4) KTDa v1.0 실행

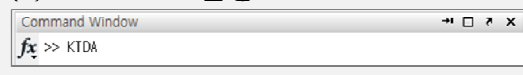


Figure 2.2.1. KTDa v1.0 실행 흐름도

2.3 포맷 변경

○ 개요

울릉도 해일파고계 파일은 201103111230.kan과 같이 날짜와 시간을 파일명으로 하며, 10분 동안의 데이터로 구성되어 있다. 데이터는 Figure 2.3.1과 같이 첫 번째 줄에 날짜와 수심기준면으로부터의 센서 거리, 두 번째 줄에 관측 지표, 그리고 세 번째 줄부터 10초 간격의 관측값으로 구성되어 있다. 울릉도 해일파고계 파일 중 일부는 다음과 같은 특이사항이 있다.

- (1) 파일 미존재: 장시간 결측으로 인하여 kan파일이 없는 경우이다. 예를 들어 201103111230.kan 파일 다음에 201103111240.kan 파일 없이 201103111250.kan 혹은 더 이후 시간의 파일로 넘어간다.
- (2) 데이터 결측: 10분 동안의 데이터가 10초 간격으로 존재하므로 한 파일 안에 데이터가 60개 존재해야 하는데 결측에 의해 데이터가 60개보다 적은 경우이다.
- (3) 데이터 과다: 한 파일 안에 60개 이상의 데이터가 저장되어 있는 경우이다.

```
1 "Date","2011/03/11","Datum Level(From Sensor to DL)","1030 cm"  
2 "Time","Level","Difference","Temperature"  
3 "12:20:00"," 12","0","6.5"  
4 "12:20:10"," 13","0","6.4"  
5 "12:20:20"," 13","0","6.4"
```

Figure 2.3.1. 울릉도 해일파고계 데이터

○ 원리

울릉도 해일파고계 파일의 특이사항을 고려하여 KTDA v1.0에 적합하도록 데이터 포맷을 변경하는 원리는 다음과 같다.

- (1) 시간과 파고로 구성된 2열 데이터를 생성한다. 1열은 지정한 시작 시간 (*starttime*)부터 끝 시간(*endtime*)까지 일정한 간격 (*dt*)의 시간이다. 2열은 결측 파고를 의미하는 NaN으로 입력된 파고이다.
- (2) 울릉도 해일파고계 데이터의 연, 월, 일, 시간, 분, 초 정보를 읽는다. (1)에서 생성한 시간과 동일한 시간에 해당하는 울릉도 해일파고계의 파고 데이터로 (1)의 파고 (NaN)를 대체한다.

○ 변수 설정

포맷 변경을 수행하기 위하여 컨트롤 파일 (input.dat) 내 아래의 변수들을 설정해야 한다.

Table 2.3.1 포맷 변경 변수

줄번호	변수	내용
04	<i>switch1</i>	포맷 변경 수행 여부 (0: 미수행, 1: 수행, 2: 수행+ 저장) (예) 2
13	<i>dt</i>	데이터 간격 (초) (예) 10
14	<i>sd</i>	결과 데이터의 유효 소수점 수 (예) 10
18	<i>starttime</i>	데이터 시작 시점 (연, 월, 일, 시, 분, 초) (예) 2011,3,1,0,0,0
19	<i>endtime</i>	데이터 종료 시점 (연, 월, 일, 시, 분, 초) (예) 2011,3,31,23,0,0
61	<i>output</i>	<i>switch1</i> 을 2로 설정 시 결과를 저장할 파일명 (예) result10sec.dat

○ 수행 결과

- *switch1*을 0으로 설정한 경우

MATLAB 명령 창에 'Without Format Change'를 출력한다.

- *switch1*을 1로 설정한 경우

1열에 MATLAB 시간, 2열에 파고 (결측 구간에는 NaN이 입력됨)가 입력된 *totaldata*를 생성한다. MATLAB 명령 창에 사용한 울릉도 해일파고계 데이터명과 'Data Format Change Complete'를 출력한다.

- *switch1*을 2로 설정한 경우

ouput 이름으로 된 파일에 *totaldata*를 저장한다. MATLAB 명령 창에 사용한 울릉도 해일파고계 데이터명과 'Data Format Change Complete'를 출력한다.

- *switch1*을 0, 1, 2 이외의 값으로 설정한 경우

MATLAB 명령 창에 'Check "switch1"'을 출력한다.

734573.7143518518	23.0000000000
734573.7144675925	22.0000000000
734573.7145833334	22.0000000000
734573.7146990740	NaN
734573.7148148149	22.0000000000
734573.7149305556	21.0000000000

Figure 2.3.2. 포맷 변경 결과 파일 데이터

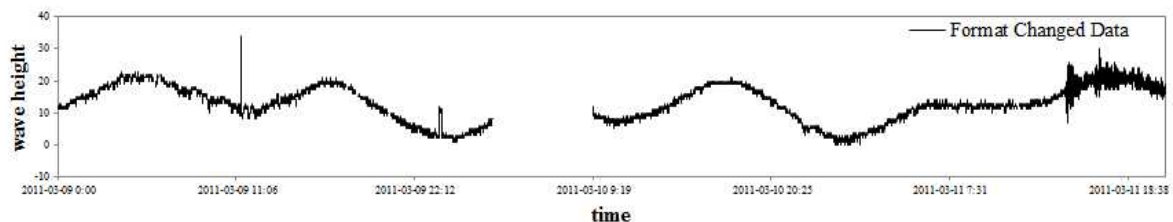


Figure 2.3.3. 포맷 변경 시계열

2.4 Spike 제거

○ 개요

공중발사형 초음파식 파고조위계인 울릉도 해일파고계는 조간대 지점에 설치되어 있어 외해에서 유입되는 파의 회절, 굴절, 반사 등 파의 특성이 변형된 자료가 관측될 수 있다 (윤용훈 등, 2002). 외부에서 관측하는 울릉도 해일파고계 자료에는 짧은 시간 동안 갑자기 파고가 변하는 spike가 존재한다. Spike는 지진해일 검출 알고리즘에 영향을 주어 지진해일이 아닌 경우에도 지진해일로 판단할 수 있으므로 spike를 제거하여야 한다.

○ 원리

KTDA v1.0에 적용한 spike 제거 방법인 Tukey 53H 방법 (Goring and Nikora, 2002)의 원리는 다음과 같다.

- (1) 파고 5개 ($h_{i-2} \sim h_{i+2}$)의 중앙값 ($h_i^{(1)}$)을 구한다.
- (2) 중앙값 3개 ($h_{i-1}^{(1)} \sim h_{i+1}^{(1)}$)의 중앙값 ($h_i^{(2)}$)을 구한다.
- (3) 필터링한 파고 ($h_i^{(3)}$)를 아래 식으로 구한다.

$$h_i^{(3)} = \frac{1}{4}(h_{i-1}^{(2)} + 2h_i^{(2)} + h_{i+1}^{(2)})$$

- (4) 관측값과 필터링 파고의 차이에 대한 절댓값($|h_i - h_i^{(3)}|$)이 한계값 ($thspike$)을 초과하면 spike로 판단하여 제거한다.

(참고) Tukey 53H 방법의 기존 판단 기준은 표준편차에 한계값 ($thspike$)를 곱한 값이나 실시간 관측 자료에 적용 가능하도록 원리 (4)와 같이 수정하였다.

○ 변수 설정

Spike 제거를 수행하기 위하여 컨트롤 파일 (input.dat) 내 아래의 변수들을 설정해야 한다.

Table 2.4.1 Spike 제거 변수

줄번호	변수	내용
05	<i>switch2</i>	Spike 제거 수행 여부 (0: 미수행, 1: 수행, 2: 수행+저장) (예) 2
14	<i>sd</i>	결과 데이터의 유효 소수점 수 (예) 10
23	<i>inputnamesp</i>	<i>switch1</i> 을 0으로 설정 시 불러올 외부 파일명 (예) result10sec.dat
24	<i>thspike</i>	Tukey 53H 방법에 사용되는 spike 판단 한계값 (cm) (예) 1
62	<i>output2</i>	<i>switch2</i> 을 2로 설정 시 결과를 저장할 파일명 (예) despike.dat

○ 수행 결과

- *switch2*를 0으로 설정한 경우

MATLAB 명령 창에 'Without Despiking'을 출력한다.

- *switch2*를 1로 설정한 경우

1열에 MATLAB 시간, 2열에 파고 (spike로 판단된 부분에는 NaN이 입력됨) 가 입력된 *totaldata2*를 생성한다. MATLAB 명령 창에 'Despiking Complete'를 출력한다.

- *switch2*를 2로 설정한 경우

ouput2 이름으로 된 파일에 *totaldata2*를 저장한다. MATLAB 명령 창에 'Despiking Complete'를 출력한다.

- *switch2*를 0, 1, 2 이외의 값으로 설정한 경우

MATLAB 명령 창에 'Check "switch2"'을 출력한다.

734563.0032407407	25.0000000000
734563.0033564814	24.0000000000
734563.0034722223	23.0000000000
734563.0035879629	NaN
734563.0037037038	NaN
734563.0038194445	23.0000000000

Figure 2.4.1. Spike 제거 결과 파일 데이터

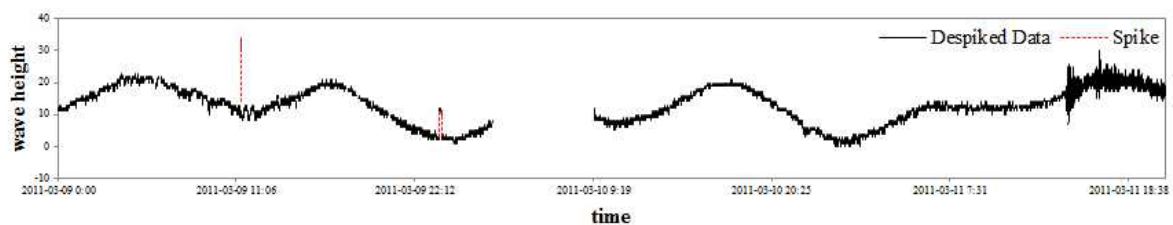


Figure 2.4.2. Spike 제거 시계열

2.5 결측 구간 파고 예측

○ 개요

DART 알고리즘, 경사 알고리즘 그리고 detide 알고리즘은 과거 데이터를 사용한다. 그로 인해 결측 발생 시, 지진해일 검출 알고리즘의 경우 관측을 다시 시작하여도 일정 시간이 경과하여야 적용이 가능하며 detide 알고리즘의 경우 정확도가 낮아진다. 그러므로 지진해일 검출 알고리즘의 복구 속도와 정확도를 향상시키기 위하여 결측 구간 파고를 예측하여야 한다.

○ 원리

- (1) 파고 결측 구간의 길이가 1칸인 경우, 결측 발생 직전 파고값을 사용한다.
- (2) 파고 결측 구간의 길이가 2칸 이상 기준 칸 (*gapnumber*) 미만인 경우, 결측 구간 앞 데이터 (*numberinter*개)와 결측 구간 뒤 데이터 (2개)를 선형 보간한 파고를 사용한다.
- (3) 파고 결측 구간의 길이가 기준 칸 (*gapnumber*) 이상인 경우 EP (End-Point fixing method)로 계산한 파고를 사용한다.

(참고) EP의 원리

장시간의 결측 구간은 불규칙한 해수면 변화를 포함하고 있어 선형 보간법을 이용해 결측 구간의 파고를 예측하는 것은 부적합하다. 본 연구에서는 불규칙한 해수면 변화 예측이 가능한 끝점 연결 기법 (EP)을 고안하여 장시간의 결측 구간 파고 예측에 적용하였다. Figure 2.5.1은 EP 개념도이며 과거 파고 중 결측 구간 주변 파고와 가장 유사한 파고를 채택한 후, 해수면 변화의 연속성을 유지하기 위하여 아래의 식을 이용하여 결측 구간의 각 끝점을 관측값의 각 끝점에 연결하는 것을 나타내었다.

$$H_{EP}(t) = H_{ori}(t) + \frac{b-a}{d}(t-c) + a \quad \text{for } c \leq t \leq c+d$$

여기서 $H_{EP}(t)$ 는 끝점 연결 후 파고, $H_{ori}(t)$ 는 끝점 연결 전 파고, a 는 $h_1 - h_2$, b 는 $h_4 - h_3$, c 는 t_1 , d 는 $t_2 - t_1$ 이다. Figure 2.5.2는 EP 흐름도이다. 먼저 결측의 크기 (gap size)에 $window_size$ 를 곱한 크기의 target window와 동일 크기의 search window를 설정한다. 그리고 결측 구간을 제외한 것을 각각 target data와 search data라 정의한다. Search data는 $npastdata$ 와 결측의 크기 (gap size)를 곱한 크기만큼의 과거 데이터까지 이동하며 target data와의 평균 절대오차 (mae)를 계산한다. 그 중 평균절대오차가 최소가 되는 search data를 결측 구간 데이터로 채택한다. Search window의 결측 구간 데이터와 앞뒤 1개씩의 데이터로 구성된 SW data를 target window의 결측 구간 앞뒤의 양 끝점을 기준으로 연결한다. 끝점이 연결된 SWEP data를 결측 구간 파고로 사용한다.

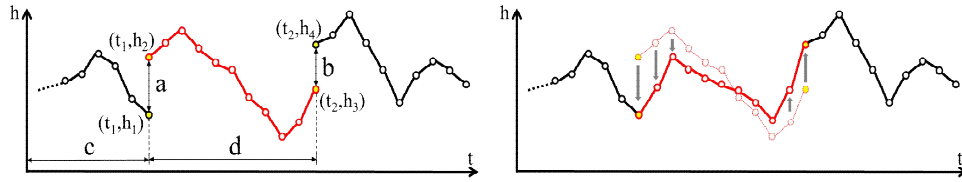


Figure 2.5.1. EP 개념도

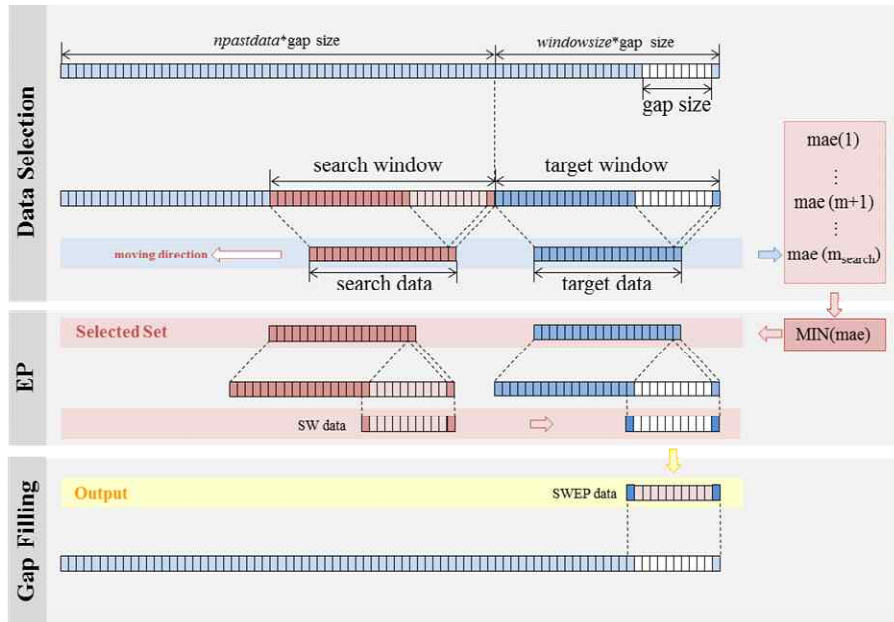


Figure 2.5.2. EP 흐름도

○ 변수 설정

결측 구간 파고 예측을 수행하기 위하여 컨트롤 파일 (input.dat) 내 아래의 변수들을 설정해야 한다.

Table 2.5.1 결측 구간 파고 예측 변수

줄번호	변수	내용
06	<i>switch3</i>	결측구간파고예측 수행 여부 (0: 미수행, 1: 수행, 2: 수행+ 저장) (예) 2
14	<i>sd</i>	결과 데이터의 유효 소수점 수 (예) 10
28	<i>inputnamegap</i>	<i>switch1</i> , <i>switch2</i> 를 0으로 설정 시 불러올 외부 파일명 (예) despikes.dat
29	<i>numberinter</i>	선형 보간에 사용되는 결측 구간 앞 데이터 수 (개) (예) 5
30	<i>gapnumber</i>	EP로 결측 구간 파고를 예측할 최소 결측 구간 수 (개) (예) 30
31	<i>window size</i>	EP를 수행할 창 크기 결정 계수 (예) 2
32	<i>npastdata</i>	EP를 수행할 과거 데이터 수 결정 계수 (예) 10
63	<i>output3</i>	<i>switch3</i> 를 2로 설정 시 결과를 저장할 파일명 (예) gapfill.dat

○ 수행 결과

- *switch3*을 0으로 설정한 경우

MATLAB 명령 창에 'Without Gap Filling'을 출력한다.

- *switch3*을 1로 설정한 경우

1열에 MATLAB 시간, 2열에 결측 구간 파고를 예측한 파고가 입력된 *totaldata3*을 생성한다. MATLAB 명령 창에 'Gap Filling Complete'를 출력한다.

- *switch3*을 2로 설정한 경우

output3 이름으로 된 파일에 *totaldata3*을 저장한다. MATLAB 명령 창에 'Gap Filling Complete'를 출력한다.

- *switch3*을 0, 1, 2 이외의 값으로 설정한 경우

MATLAB 명령 창에 'Check "switch3"'을 출력한다.

734563.0146990741	36.0000000000
734563.0148148148	29.0000000000
734563.0149305556	25.0000000000
734563.0150462963	25.3333333333
734563.0151620370	25.6666666667
734563.0152777778	26.0000000000

Figure 2.5.3. 결측 구간 파고 예측 결과 파일 데이터

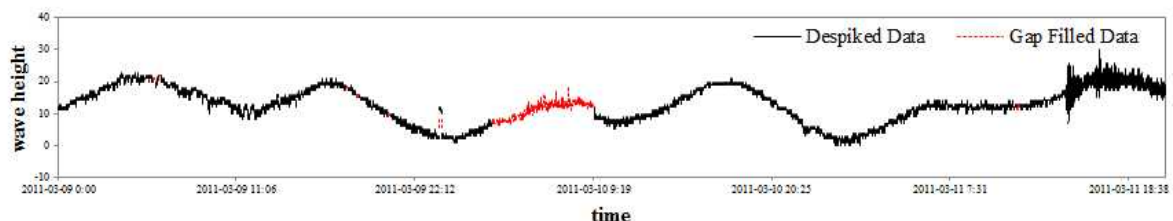


Figure 2.5.4. 결측 구간 파고 예측 시계열

2.6 DART 알고리즘

○ 개요

DART는 Deep-ocean Assessment and Reporting of Tsunamis의 약자로 전세계 해양에 설치된 부이를 이용한 실시간 지진해일 모니터링 시스템이다. Mofjeld (1997)가 제안하고 NOAA의 DART 부이에 사용 중인 지진해일 검출 알고리즘은 1시간 간격의 과거 파고 4개를 보간하여 예측한 파고와 실제 파고의 차이가 기준치를 초과하면 지진해일로 판단한다. Mofjeld (1997)의 지진해일 검출 알고리즘은 데이터를 15초 간격으로 저장하는 DART 부이에 적합하도록 계수가 고정되어 있다. KTDA v1.0에는 Mofjeld (1997)의 지진해일 검출 알고리즘과 동일한 원리를 사용하되, 데이터를 10초 간격으로 저장하는 울릉도 해일파고계뿐만 아니라 모든 간격에 적용 가능하도록 수정한 DART 알고리즘을 개발하여 적용하였다.

○ 원리

- (1) 과거 3시간 동안의 파고로부터 한 시간 간격의 n 분 평균 파고 ($\bar{\zeta}$)를 구한다.
- (2) 평균을 취하는 시간 (n)과 예측 시점 (x)으로부터 다항식의 계수 (ω)를 구한다.
- (3) 아래의 식으로 구한 예측 파고 ($\hat{\zeta}$)와 실제 파고의 차이가 지진해일 검출 한계값 (*threshold*) 이상이면 지진해일로 판단한다.

$$\hat{\zeta}(t') = \sum_{i=0}^3 \omega_i \bar{\zeta}(t'' - i\Delta t)$$

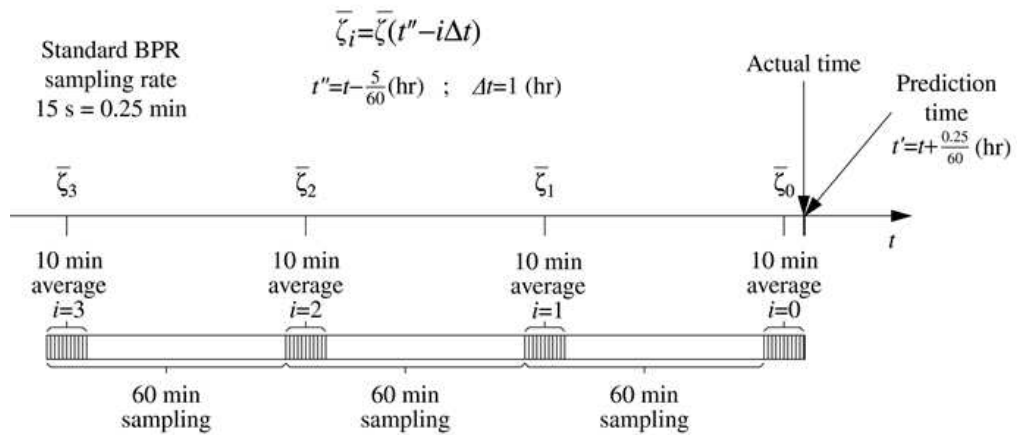


Figure 2.6.1. DART 알고리즘 개념도 (Beltrami, 2008, Figure 1)

○ 변수 설정

DART 알고리즘을 수행하기 위하여 컨트롤 파일 (input.dat) 내 아래의 변수들을 설정해야 한다.

Table 2.6.1 DART 알고리즘 변수

줄번호	변수	내용
07	<i>switchdart</i>	DART 알고리즘 수행 여부 (0: 미수행, 1: 수행, 2: 수행+ 저장) (예) 2
14	<i>sd</i>	결과 데이터의 유효 소수점 수 (예) 10
36	<i>inputnamedart</i>	<i>switch1</i> , <i>switch2</i> , <i>switch3</i> 을 0으로 설정 시 불러올 외부 파일명 (예) gapfill.dat
37	<i>n</i>	원리 (1) 단계의 평균을 취하는 시간 (min) (예) 10
38	<i>x</i>	DART 알고리즘으로 예측한 파고의 시점 (min) (예) 10/60
39	<i>threshold</i>	지진해일 검출 한계값 (cm) (예) 3
64	<i>outputdart</i>	<i>switchdart</i> 를 2로 설정 시 결과를 저장할 파일명 (예) dart.dat

○ 수행 결과

- *switchdart*를 0으로 설정한 경우

MATLAB 명령 창에 'Without Dart Algorithm'을 출력한다.

- *switchdart*를 1로 설정한 경우

Figure 2.6.2와 같은 그림을 출력한다. 그림의 위 칸은 관측값 (청록색)과 DART 알고리즘의 예측 파고 (빨간색) 시계열을 나타낸다. 그림의 아래 칸은 관측값과 예측 파고의 차이에 대한 절댓값 (파란색)과 지진해일 검출 한계값 (빨간색)을 나타낸다. 1열에 MATLAB 시간, 2열에 파고, 3열에 지진해일 검출정보 (검출: 1, 미검출: 2, 알고리즘 적용 불가: NaN)가 입력된 *totaldatadart*를 생성한다. MATLAB 명령 창에 'DART Algorithm Complete'를 출력한다.

- *switchdart*를 2로 설정한 경우

Figure 2.6.2와 같은 그림을 출력한다. *outputdart* 이름으로 된 파일에 *totaldatadart*를 저장한다. MATLAB 명령 창에 'DART Algorithm Complete'를 출력한다.

- *switchdart*를 0, 1, 2 이외의 값으로 설정한 경우

MATLAB 명령 창에 'Check "switchdart"'를 출력한다.

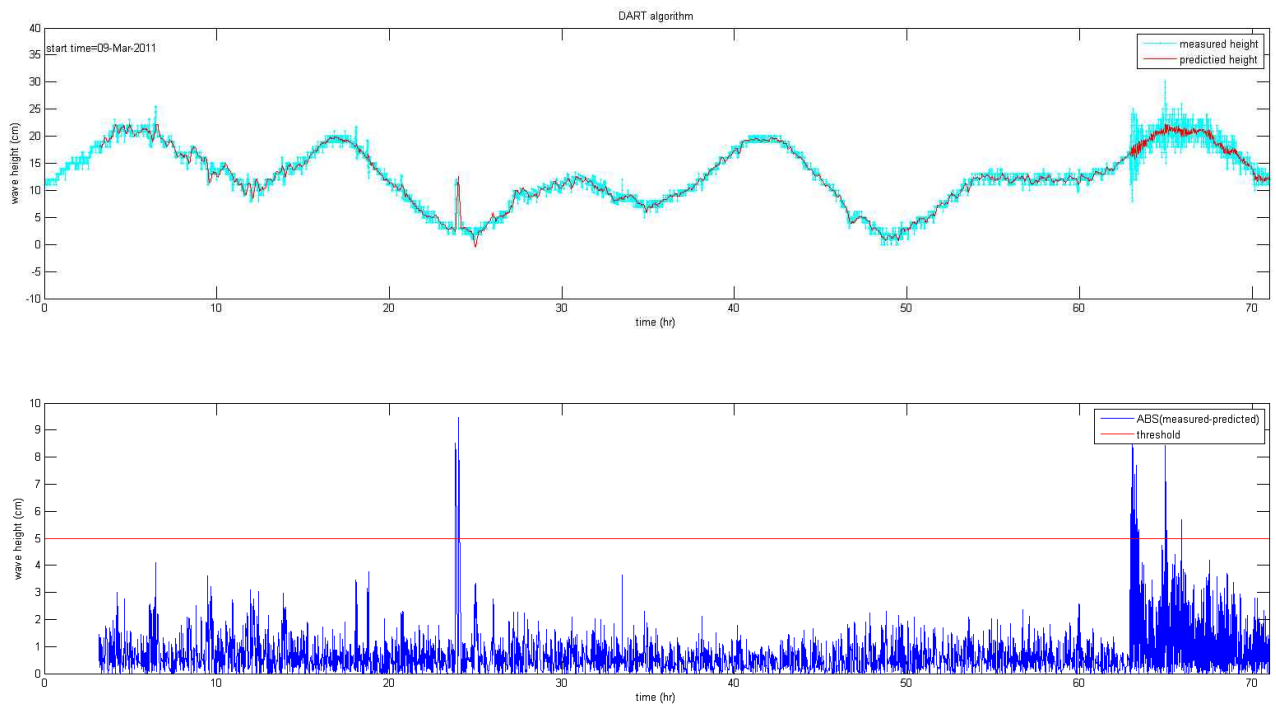


Figure 2.6.2. DART 알고리즘 수행 결과

2.7 경사 알고리즘

○ 개요

Bressan and Tinti (2011)가 제안한 지진해일 검출 알고리즘으로 단시간 파고 경사 또는 단시간 파고 경사를 장시간 파고 경사로 나눈 값이 기준치를 초과하면 지진해일로 판단한다. KTDA v1.0에는 Bressan and Tinti (2011)의 지진해일 검출 알고리즘과 동일한 원리를 사용하되, 데이터를 10초 간격으로 저장하는 울릉도 해일파고계뿐만 아니라 모든 간격에 적용 가능하도록 수정한 경사 알고리즘을 개발하여 적용하였다.

○ 원리

- (1) 과거 기준 시간 (t_{IS})으로부터 현재 시간 (t)까지 ($I_{IS}(t)$)의 파고로부터 파고 경사 ($IS_T(t)$)를 구한다.
- (2) 현재 시간으로부터 기준 간격 (t_{GTide})만큼 떨어진 일정 시간 (t_{Tide}) 동안 ($I_{Tide}(t)$)의 파고 경사 ($IS_T(t)$)를 평균하여 조석 경사 ($Tide_{uns}(t)$)를 구한다.
- (3) 조석 경사 ($Tide_{uns}(t)$)를 짧은 시간 (t_{sm}) 동안 평균한 평균 조석 경사 ($Tide(t)$)를 구한다.
- (4) 파고 경사($IS_T(t)$)와 평균 조석 경사 ($Tide(t)$)의 차인 순간 경사 ($IS(t)$)를 구한다.
- (5) 현재 시간으로부터 기준 간격 (t_G)만큼 떨어진 일정 시간 (t_{BS}) 동안 ($I_{BS}(t)$)의 순간 경사 ($IS(t)$)를 이용하여 배경 경사 ($BS(t)$)를 구한다.
- (6) 순간 경사 ($IS(t)$)의 절댓값을 배경 경사 ($BS(t)$)로 나눈 조절 함수 ($CF(t)$)를 구한다.
- (7) 순간 경사 ($IS(t)$)의 절댓값이 순간 경사 기준치 (λ_{IS}) 이상이거나, 조절 함수 ($CF(t)$)가 조절 함수 기준치 (λ_{CF}) 이상이면 지진해일로 판단한다.

(참고) 배경 경사 ($BS(t)$) 계산 방법

Bressan and Tinti (2011)는 아래와 같이 3가지 방법의 배경 경사를 제안하고 있다. BS1은 순간경사 최댓값과 최솟값 차의 절반이며, BS2는 순간경사의 표준편차에 2의 제곱근을 곱한 값이며, BS3는 순간경사 절댓값 중 최댓값이다.

$$BS1(t) = [\max(IS(t')) - \min(IS(t'))]/2; \quad t' \in I_{BS}(t)$$

$$BS2(t) = standard\ deviation\ of\ IS(t') \cdot \sqrt{2};\ t' \in I_{BS}(t)$$

$$BS3(t) = \max(|IS(t')|); \ t' \in I_{BS}(t)$$

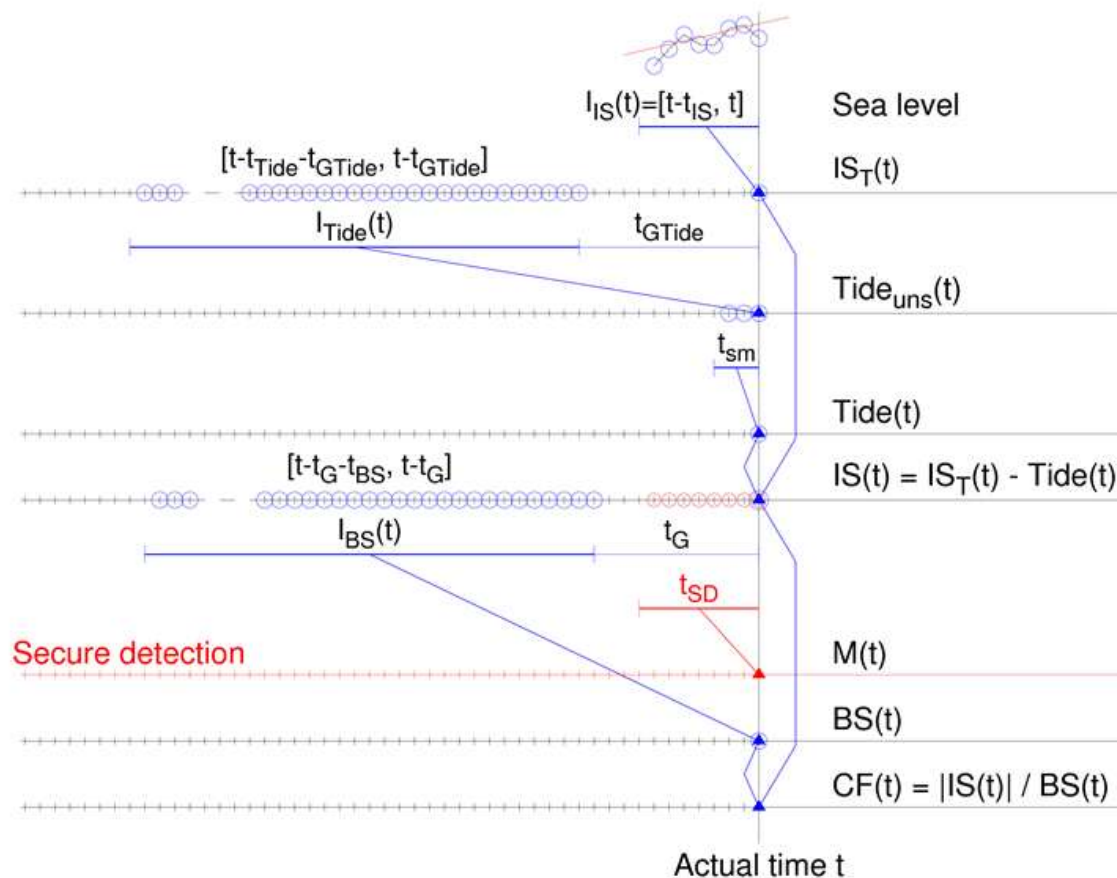


Figure 2.7.1. 경사 알고리즘 개념도 (Bressan and Tinti, 2011, Figure 1)

○ 변수 설정

경사 알고리즘을 수행하기 위하여 컨트롤 파일 (input.dat) 내 아래의 변수들을 설정해야 한다.

Table 2.7.1 경사 알고리즘 변수

줄번호	변수	내용
08	<i>switchslope</i>	경사 알고리즘 수행 여부 (0: 미수행, 1: 수행, 2: 수행+ 저장) (예) 2
14	<i>sd</i>	결과 데이터의 유효 소수점 수 (예) 10
43	<i>inputnameslope</i>	<i>switch1</i> , <i>switch2</i> , <i>switch3</i> 을 0으로 설정 시 불러올 외부 파일명 (예) gapfill.dat
44	<i>tIS</i>	파고 경사 ($IS_T(t)$) 계산에 사용하는 시간 ($\times 10$ sec) (예) 10*6
45	<i>tG</i>	배경 경사 ($BS(t)$) 계산에 사용하는 지연 시간 ($\times 10$ sec) (예) 15*6
46	<i>tGTide</i>	조석 경사 ($Tide_{uns}(t)$) 계산에 사용하는 지연 시간 ($\times 10$ sec) (예) 15*6+ 1*6
47	<i>tTide</i>	조석 경사 ($Tide_{uns}(t)$) 계산에 사용하는 시간 간격 ($\times 10$ sec) (예) 60*6
48	<i>tsm</i>	평균 조석 경사 ($Tide(t)$) 계산에 사용하는 시간 간격 ($\times 10$ sec) (예) 6*6
49	<i>tBS</i>	배경 경사 ($BS(t)$) 계산에 사용하는 시간 간격 ($\times 10$ sec) (예) 60*6
50	<i>ISthreshold</i>	지진해일 검출 순간 경사 한계값 (cm/sec) (예) 1
51	<i>CFthreshold</i>	지진해일 검출 조절 함수 한계값 (예) 2.15
65	<i>outputslope</i>	<i>switchslope</i> 을 2로 설정 시 결과를 저장할 파일명 (예) slope.dat

○ 수행 결과

- *switchslope*을 0으로 설정한 경우

MATLAB 명령 창에 'Without Slope Algorithm'을 출력한다.

- *switchslope*을 1로 설정한 경우

Figure 2.7.2와 같은 그림을 출력한다. 그림의 1행은 관측값 (청록색) 시계열을 나타내며 2행은 순간 경사 (파란색)와 순간 경사 한계값 (빨간색)을 나타낸다. 3행은 배경 경사1 (검은색), 배경 경사2 (녹색), 배경 경사3 (파란색)을 나타내며 4행은 조절 함수1 (검은색), 조절 함수2 (녹색), 조절 함수3 (파란색)과 조절 함수 한계값 (빨간색)을 나타낸다. 1열에 MATLAB 시간, 2열에 파고, 3열에 순간경사에 의한 지진해일 검출정보, 4열에 조절 함수1에 의한 지진해일 검출정보, 5열에 조절 함수2에 의한 지진해일 검출정보, 6열에 조절 함수3에 의한 지진해일 검출정보 (검출: 1, 미검출: 2, 알고리즘 적용 불가: NaN)가 입력된 *totaldataslope*을 생성한다. MATLAB 명령 창에 사용한 변수값과 함께 'Slope Algorithm Complete'를 출력한다.

- *switchslope*을 2로 설정한 경우

Figure 2.7.2와 같은 그림을 출력한다. *outputslope* 이름으로 된 파일에 *totaldataslope*을 저장한다. MATLAB 명령 창에 사용한 변수값과 함께 'Slope Algorithm Complete'를 출력한다.

- *switchslope*을 0, 1, 2 이외의 값으로 설정한 경우

MATLAB 명령 창에 'Check "switchslope"'을 출력한다.

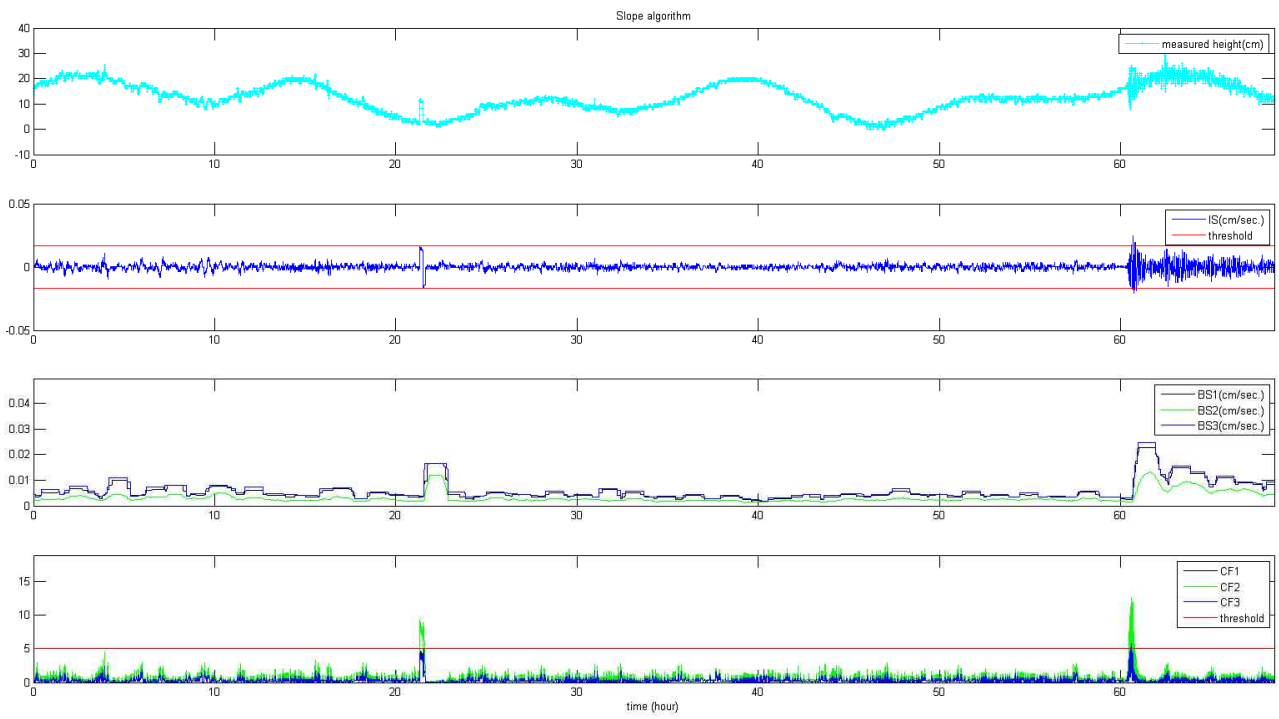


Figure 2.7.2. 경사 알고리즘 수행 결과

2.8 Detide 알고리즘

○ 개요

울릉도 해일파고계 데이터로부터 지진해일의 파고를 알기 위해서는 조석 성분을 제거해야 한다. Pawlowicz *et al.* (2002)이 개발한 조화분해 코드인 T_TIDE를 통해 해수면 변화 자료로부터 조석 성분을 구할 수 있다. 조화분해에는 장기간의 시계열 데이터를 사용해야 하는데 10초 간격인 울릉도 해일파고계 데이터를 적용하면 조석 성분을 구하는데 많은 시간이 소요된다. KTDA v1.0에는 재 샘플링을 통해 간격 조정된 데이터로 조화분해를 수행한 후 원래 간격으로 출력하도록 수정한 detide 알고리즘을 개발하여 적용하였다.

○ 원리

- (1) 시계열 데이터를 설정한 간격 (*inter*)으로 재 샘플링을 수행한다.
- (2) 재 샘플링한 데이터로 위도 (*lat*)를 고려한 조화분해를 수행하여 조화상수를 계산한다.
- (3) 조화상수를 바탕으로 재 샘플링 수행 전 데이터 간격의 조석 성분을 구한다.
- (4) 시계열 데이터에서 조석 성분을 제거한다.

○ 변수 설정

Detide 알고리즘을 수행하기 위하여 컨트롤 파일 (input.dat) 내 아래의 변수들을 설정해야 한다.

Table 2.8.1 Detide 알고리즘 변수

줄번호	변수	내용
09	<i>switchtide</i>	Detide 알고리즘 수행 여부 (0: 미수행, 1: 수행, 2: 수행+ 저장) (예) 2
14	<i>sd</i>	결과 데이터의 유효 소수점 수 (예) 10
55	<i>inputnametide</i>	<i>switch1</i> , <i>switch2</i> , <i>switch3</i> 을 0으로 설정 시 불러올 외부 파일명 (예) gapfill.dat
56	<i>inter</i>	조화분해에 사용할 데이터 간격 (hr) (예) 1/60
57	<i>lat</i>	관측소의 위도 (deg) (예) 37+ 29/60
66	<i>outputtide</i>	<i>switchtide</i> 를 2로 설정 시 결과를 저장할 파일명 (예) ttide.dat

○ 수행 결과

- *switchtide*를 0으로 설정한 경우

MATLAB 명령 창에 'Without Detide Algorithm'을 출력한다.

- *switchtide*를 1로 설정한 경우

Figure 2.8.1과 같은 그림을 출력한다. 그림은 관측값 (파란색), 예측 조석 (녹색), 관측값과 예측 조석의 차이 (빨간색) 시계열을 나타낸다. MATLAB 명령 창에 조화상수 주기, 진폭, 상 등의 정보와 함께 'Detide Algorithm Complete'를 출력한다.

- *switchtide*를 2로 설정한 경우

Figure 2.8.1과 같은 그림을 출력한다. *outputtide* 이름의 파일에 1열은 MATLAB 시간, 2열은 예측 조석의 파고 (*xout*)가 입력된 행렬을 저장한다. MATLAB 명령 창에 조화상수 주기, 진폭, 상 등의 정보와 함께 'Detide Algorithm Complete'를 출력한다.

- *switchtide*를 0, 1, 2 이외의 값으로 설정한 경우

MATLAB 명령 창에 'Check "switchtide"'를 출력한다.

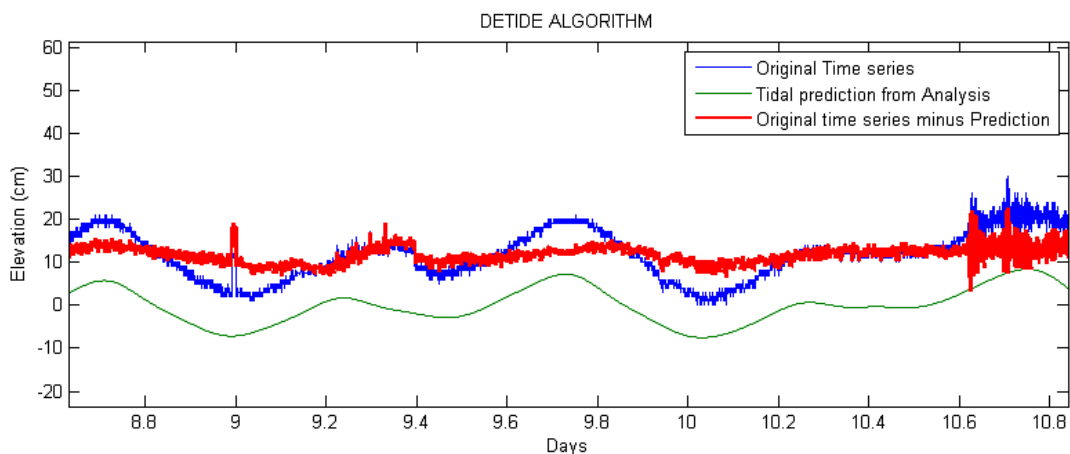


Figure 2.8.1. Detide 알고리즘 수행 결과

3. KTDA.m 코드

본 절에서는 KTDA.m 코드에 대한 상세한 내용을 기술하였다.

001	%-----%	%
002	% KTDA v1.0	%
003	%-----%	%
004	% J.-W. LEE (2014)	%
005	% Contact: junwhan89@korea.kr	%
006	%-----%	%
007	% <Required functions>	%
008	% (1) saveascii.m: http://www.mathworks.com/matlabcentral/fileexchange/	%
009	% 10243-saveascii-m-v6-2-oct-2009-	%
010	% (2) vlookup.m: http://www.mathworks.com/matlabcentral/fileexchange/	%
011	% 29233-vlookup-similar-to-ms-excel-function	%
012	% (3) gfit2.m: http://www.mathworks.com/matlabcentral/fileexchange/	%
013	% 22020-goodness-of-fit-modified-	%
014	% (4) T_Tide package: http://www.eos.ubc.ca/~rich/#T_Tide	%
015	%-----%	%
016	clear all close all fclose all clc;	
017	%-----%	%
018	% Loading data	%
019	%-----%	%
020	inputfilesheet=fopen('input.dat');	
021	inputfileinside=textscan(inputfilesheet,'%s%s%s','delimiter','=');	
022	switch1=str2double(cell2mat(inputfileinside{2}(4)));	
023	switch2=str2double(cell2mat(inputfileinside{2}(5)));	
024	switch3=str2double(cell2mat(inputfileinside{2}(6)));	
025	switchdart=str2double(cell2mat(inputfileinside{2}(7)));	
026	switchslope=str2double(cell2mat(inputfileinside{2}(8)));	
027	switchtide=str2double(cell2mat(inputfileinside{2}(9)));	
028	dt=str2num(cell2mat(inputfileinside{2}(13)));	
029	sd=str2num(cell2mat(inputfileinside{2}(14)));	
030	starttime=datenum(str2num(cell2mat(inputfileinside{2}(18))));	
031	endtime=datenum(str2num(cell2mat(inputfileinside{2}(19))));	
032	inputnamesp=cell2mat(inputfileinside{2}(23));	
033	thspike=str2num(cell2mat(inputfileinside{2}(24)));	
034	inputnamegap=cell2mat(inputfileinside{2}(28));	
035	numberinter=str2num(cell2mat(inputfileinside{2}(29)));	
036	gapnumber=str2num(cell2mat(inputfileinside{2}(30)));	

	<p>KTDA v1.0 코드를 이해하는데 도움이 될 수 있는 MATLAB 관련 내용을 기술하였다. 왼쪽 칸의 번호는 관련 내용이 시작하는 KTDA v1.0 코드의 줄번호이다.</p> <p>MATLAB 내장 함수에 대한 상세한 설명은 MATLAB 명령 창에 'help 내장함수명'을 입력하여 찾을 수 있다.</p> <p>MATLAB 코드의 녹색은 코멘트로 코드 실행에 영향이 없다. :: 코드 마지막에 세미클론을 붙이면 명령 창에 계산 결과를 출력하지 않는다.</p>
007	KTDA.m을 구동하기 위하여 필요한 mfile 등은 명시된 사이트에서 다운로드 가능하다.
014	(주의) KTDA.m은 사이트에서 받은 코드 중 t_tide.m이 수정된 것을 사용한다.
016	기존에 저장된 변수, 열린 파일, 열린 함수, 명령 창에 기입된 내용을 모두 지우고 달는다.
020	input.dat 파일을 열어 내용을 <i>inputfilesheet</i> 에 셀로 저장한다.
021	구분 기호 (delimiter)인 =와 :을 기준으로 셀을 3칸의 문자열 (%s%s%s)로 나눈다.
022	<p>str2double: 문자열 변수를 배열밀도의 실수/복소수로 변경한다.</p> <p>cell2mat: 셀 내용을 독자의 행렬로 변경한다.</p> <p>(예) switch1=str2double(cell2mat(inputfileinside{2}(4)));</p> <p>구분 기호에 의해 3개로 나뉜 셀 중 2 번째 셀의 4 번째 값을 읽어 실수 변수인 switch1에 저장한다.</p>
028	<p>str2num: 문자열 변수를 수열로 변경한다. str2double보다 연산 속도가 느리나 연산 기호가 포함되어도 변경할 수 있다는 장점이 있다.</p>

```

037 windowsize=str2num(cell2mat(inputfileinside{2}(31)));
038 npastdata=str2num(cell2mat(inputfileinside{2}(32)));
039 inputnamedart=cell2mat(inputfileinside{2}(36));
040 n=str2num(cell2mat(inputfileinside{2}(37)));
041 x=str2num(cell2mat(inputfileinside{2}(38)));
042 threshold=str2num(cell2mat(inputfileinside{2}(39)));
043 inputnameslope=cell2mat(inputfileinside{2}(43));
044 tIS=str2num(cell2mat(inputfileinside{2}(44)));
045 tG=str2num(cell2mat(inputfileinside{2}(45)));
046 tGTide=str2num(cell2mat(inputfileinside{2}(46)));
047 tTide=str2num(cell2mat(inputfileinside{2}(47)));
048 tsm=str2num(cell2mat(inputfileinside{2}(48)));
049 tBS=str2num(cell2mat(inputfileinside{2}(49)));
050 lSthreshold=str2num(cell2mat(inputfileinside{2}(50)));
051 CFthreshold=str2num(cell2mat(inputfileinside{2}(51)));
052 inputnametide=cell2mat(inputfileinside{2}(55));
053 inter=str2num(cell2mat(inputfileinside{2}(56)));
054 lat=str2num(cell2mat(inputfileinside{2}(57)));
055 output=cell2mat(inputfileinside{2}(61));
056 output2=cell2mat(inputfileinside{2}(62));
057 output3=cell2mat(inputfileinside{2}(63));
058 outputdart=cell2mat(inputfileinside{2}(64));
059 outputslope=cell2mat(inputfileinside{2}(65));
060 outputtide=cell2mat(inputfileinside{2}(66));
061 disp('Input Data Loading Complete');
062 %------%
063 % Format Change Algorithm %
064 %------%
065 if (switch1==1)||(switch1==2)
066     !ls *.kan >list.dat
067     ttt=(endtime-starttime)*(24*60*60/dt);
068     for tt=1:ttt
069         totaltime(tt)=starttime+dt/(24*60*60)*(tt-1);
070         totalheight(tt)=NaN;
071         timedata=datestr(totaltime(tt),'dd mm yyyy HH:MM:SS');
072         dateref=str2double(timedata(1:2));
073         monthref=str2double(timedata(4:5));
074         yearref=str2double(timedata(7:10));
075         hourref=str2double(timedata(12:13));

```

061	disp: 명령 창에 작은따옴표 안 문자열을 출력한다.
065	만약 <i>switch1</i> 이 1 또는 2이면 아래의 명령어를 수행한다.
066	!: 쉘 이탈 명령어로 뒷부분을 운영체제에 대한 명령어로 인식한다. ls: 파일명을 출력하는 리눅스 명령어이다. (예) <code>lls *.kan >list.dat</code> .kan으로 끝나는 파일명을 모두 찾아 list.dat 파일에 저장한다. (주의) 리눅스 명령어이므로 윈도우 운영체제에서는 오류로 인식한다.
067	<i>ttt</i> : 포맷 변경의 원리 (1)을 수행하기 위한 전체 시간 개수
070	NaN: Not a Number의 약자로 값이 없다는 의미의 국제전기전자기술자협회 (IEEE)에서 공인한 표현이다.
071	datestr: MATLAB 시간을 원하는 형식의 문자열로 전환한다. (예) <code>timedata=datestr(totaltime(tt),'dd mm yyyy HH:MM:SS');</code> <i>totaltime</i> 에 저장된 MATLAB 시간을 '일 월 연 시:분:초' 형식의 문자열로 변경하여 <i>timedata</i> 에 저장한다.
072	<i>dateref</i> : <i>timedata</i> 의 1 번째와 2 번째 문자열만 읽어 배열길이의 실수로 변경한 일 정보

```

076         minref=str2double(timedata(15:16));
077         secondref=str2double(timedata(18:19));
078         totaltimeref(tt)=yearref*10^10+monthref*10^8+dateref*10^6+...
079         hourref*10^4+minref*100+secondref;
080     end
081     totaltimeref=[totaltimeref' totalheight'];
082     totaldata=[totaltime' totalheight'];
083     filename=textread('list.dat','%s');
084     nfile=length(filename);
085     for ii=1:1:nfile
086         file=char(filename(ii));
087         fid=fopen(file);
088         [aa,bb]=strtok(file, '.');
089         yy(ii)=str2num(aa(1:4));
090         mm(ii)=str2num(aa(5:6));
091         dd(ii)=str2num(aa(7:8));
092         aline=fgetl(fid);
093         aline=fgetl(fid);
094         while 1
095             aline=fgetl(fid);
096             [a,b]=strtok(aline, ',');
097             [c,d]=strtok(b, ',');
098             [e,f]=strtok(a, ':');
099             [g,h]=strtok(f, ':');
100             [i,j]=strtok(h, ':');
101             if aline ==-1
102                 break
103             end
104             hour=str2num(e);
105             minute=str2num(g);
106             second=str2num(i);
107             datat=yy(ii)*10^10+mm(ii)*10^8+dd(ii)*10^6+hour*10^4+...
108             minute*100+second;
109             datah=str2num(c);
110             [content,index] = vlookup(totaltimeref, datat, 1, 1);
111             totaldata(index,2)=datah;
112         end
113         fclose(fid);
114         disp(file);

```

078	<i>totaltimeref</i> 에 예를 들어 20110301000000과 같이 연월일시분초 순으로 저장한다. ...: MATLAB 코드의 문장이 긴 경우 마침표 3개를 적어 다음 줄로 넘길 수 있다.
081	': 행렬의 행을 열로, 열을 행으로 변경한다. [: 대괄호 안에 변수를 하나로 합쳐 행렬을 구성한다.
083	list.dat 파일 안의 문자열을 한 줄에 1개씩 (%s) 읽어 <i>filename</i> 에 저장한다.
084	length: 괄호 안 변수의 길이를 구한다.
086	<i>filename</i> 의 <i>ii</i> 번째 내용을 읽어 <i>file</i> 에 문자열로 저장한다.
088	strtok: 문자열에서 작은따옴표 안의 표시 (token)을 찾아 나눈다. (예) [aa,bb]=strtok(file, '.'); <i>file</i> 에 마침표(.) 표시 앞 문자열을 <i>aa</i> 에 저장하고 표시와 뒤 문자열은 <i>bb</i> 에 저장한다.
094	while문 안의 내용을 참 (1)인 동안 계속 수행한다.
095	fgetl: 파일 내용 한 줄을 읽는다.
101	fgetl로 빈 줄을 읽으면 멈추고 while문을 나간다.
110	<i>totaltimeref</i> 의 첫 행에서 <i>datat</i> 를 찾아 <i>totaltimeref</i> 의 첫 행을 <i>content</i> 에 저장하고 색인 번호를 <i>index</i> 에 저장한다.
113	열었던 파일 <i>fid</i> 를 닫는다.
114	명령 창에 열었던 파일명을 출력한다.

```

115     end
116     if switch1==2
117         saveascii(totaldata,output,sd)
118     end
119     disp('Data Format Change Complete');
120 elseif switch1==0
121     disp('Without Format Change');
122 else
123     disp('Check "switch1"');
124 end
125 %-----%
126 % Despike algorithm %
127 %-----%
128 if (switch2==1)||(switch2==2)
129     if switch1==0
130         aaa=load(inputnamesp);
131     elseif (switch1==1)||(switch1==2)
132         aaa=totaldata;
133     else
134         disp('Check "switch1"');
135     end
136     h1=aaa(:,2);
137     h1ori=h1;
138     t1=aaa(:,1);
139 % Tukey 53H method
140     for i=3:length(h1)-2
141         m11(i-2)=median([h1(i-2) h1(i-1) h1(i) h1(i+1) h1(i+2)]);
142     end
143     for i=2:length(m11)-1
144         m21(i-1)=median([m11(i-1) m11(i) m11(i+1)]);
145     end
146     for i=2:length(m21)-1
147         h11(i-1)=1/4*(m21(i-1)+2*m21(i)+m21(i+1));
148         delta1(i-1)=abs(h1(i+3)-h11(i-1));
149         if delta1(i-1)>thspike
150             h1(i+3)=NaN;
151         end
152     end
153     disp('Despiking Complete');

```

117	<i>totaldata</i> 를 <i>output</i> 이름의 파일에 유효 소수점 <i>sd</i> 까지 ascii로 저장한다.
129	포맷 변경을 수행하지 않으면 <i>inputnamesp</i> 이름의 외부 데이터를 불러온다.
130	load: 입력 함수로 모든 줄의 데이터 형식이 동일하여 행렬로 읽을 수 있는 경우에만 사용 가능하다.
136	<i>aaa</i> 의 2열 전체를 <i>h1</i> 에 저장한다.
140	spike 제거 원리 (1)에 해당한다. <i>m11</i> 은 앞 뒤 데이터 2개까지 사용하므로 <i>i</i> 는 3부터 시작하여 <i>length(h1)-2</i> 에 끝난다.
141	median: 괄호 안 데이터의 중간값을 계산한다.
144	spike 제거 원리 (2)에 해당한다.
147	spike 제거 원리 (3)에 해당한다.
148	spike 제거 원리 (4)에 해당한다. abs: 괄호 안의 값에 절댓값을 계산한다.
149	spike 여부를 판단한다.

```

154     totaldata2=[t1 h1];
155     if switch2==2
156         saveascii(totaldata2,output2,sd)
157     end
158 elseif switch2==0
159     disp('Without Despiking');
160 else
161     disp('Check "switch2"');
162 end
163 %------%
164 % Gap filling algorithm %
165 %------%
166 if (switch3==1)||(switch3==2)
167 % loading data
168     if (switch1==0)&&(switch2==0)
169         aaa=load(inputnamegap);
170     elseif ((switch1==1)||(switch1==2))&&(switch2==0)
171         aaa=totaldata;
172     elseif (switch2==1)||(switch2==2)
173         aaa=totaldata2;
174     else
175         disp('Check "swich1 and switch2"');
176     end
177     h1=aaa(:,2);
178     t1=aaa(:,1);
179     k=1;
180     for i=1:length(h1)
181         if isnan(h1(i))==1
182             stepp(k)=i;
183             k=k+1;
184         end
185     end
186     k=1;
187     p=1;
188     step(1)=0;
189     for i=1:length(stepp)
190         step(i+1)=stepp(i);
191     end
192     step(end+1)=9999999;

```


168	포맷 변경과 spike 제거 모두 수행 안하면 <i>inputnamegap</i> 이름의 외부데이터를 불러온다.
170	포맷 변경만 수행하고 spike 제거를 수행 안한 경우이다.
172	포맷 변경 수행 여부에 상관 없이 spike 제거를 수행한 경우이다.
181	isnan: 괄호 안의 값이 NaN 이면 1, NaN이 아니면 0을 출력한다.
182	<i>h1</i> 을 스캔하여 몇 번째 값이 결측인지를 <i>stepp</i> 에 저장한다.
188	194번 줄에서 결측 구간의 길이로 구분하는 알고리즘을 수행하기 위하여 가상의 결측 지점을 맨 앞에 (<i>step(1)</i>) 설정한다.
192	194번 줄에서 결측 구간의 길이로 구분하는 알고리즘을 수행하기 위하여 가상의 결측 지점을 맨 뒤에 (<i>step(end+1)</i>) 설정한다.

```

193     for i=2:length(step)-1
194         if (step(i)-step(i-1)~=1)&&(step(i+1)-step(i)~=1)
195             step2(k)=step(i);
196             k=k+1;
197         else
198             step3(p)=step(i);
199             p=p+1;
200         end
201     end
202 % One point gap filling algorithm
203     for i=1:length(step2)
204         if step2(i)>=3
205             h1(step2(i))=h1(step2(i)-1);
206         else
207             h1(step2(i))=h1(step2(i)+1);
208         end
209     end
210 % linear interpolation algorithm
211     k=1;
212     p=1;
213     step3(end+1)=9999999;
214     startt(1)=step3(1);
215     for i=2:length(step3)-1
216         if (step3(i)-step3(i-1)~=1)&&(step3(i+1)-step3(i)~=1)
217             startt(k+1)=step3(i);
218             k=k+1;
219         end
220         if (step3(i+1)-step3(i)~=1)||i==length(step3)-1
221             endt(p)=step3(i);
222             p=p+1;
223         end
224     end
225     h1L=h1;
226     for i=1:length(startt)
227         if (startt(i)>numberinter)&&(startt(i)<length(h1)-2)&& ...
228             (endt(i)>numberinter)&&(endt(i)<length(h1)-2)
229             for k=1:numberinter
230                 xlinear1(k)=(startt(i)-numberinter+(k-1));
231                 vlinear1(k)=h1(startt(i)-numberinter+(k-1));

```

194	만약 결측 지점 (<i>step</i>)이 앞 결측 지점과 2칸 이상 차이나고 뒤 결측 지점과도 2칸 이상
195	차이하면, 즉 결측 지점의 길이가 1칸이면 <i>step2</i> 에 저장한다.
198	만약 결측 구간의 길이가 2칸 이상이면 <i>step3</i> 에 저장한다.
205	결측 구간의 길이가 1칸이면 바로 전 값으로 대체한다.
207	만약 첫 칸이 결측이면 2 번째 칸 값으로 대체한다.
213	216, 220번 줄에서 결측 구간의 길이가 2칸 이상인 부분의 시작점과 끝점 정보를 구분하 기 위하여 가상의 결측 지점을 맨 뒤에 (<i>step3(end+1)</i>) 설정한다.
210	2칸 이상의 모든 결측 구간에 대하여 선형 보간을 수행한다.
217	만약 결측 지점 (<i>step3</i>)이 앞 결측 지점과 2칸 이상 차이나고 뒤 결측 지점과는 1칸 차이 나면, 즉 결측 구간의 시작점이면 <i>startt</i> 에 저장한다.
221	만약 결측 지점 (<i>step3</i>)이 뒤 결측 지점과 2칸 이상 차이거나 결측 지점의 마지막 값이 면, 즉 결측 구간의 끝점이면 <i>endt</i> 에 저장한다.
227	결측 구간의 시작점과 끝점이 선형 보간에 사용되는 앞 데이터 수 (<i>numberinter</i>)와 뒤 데 이터 수인 2개 안 범위 안에 있는 경우에만 수행한다.
230	선형 보간에 사용할 앞 데이터 수 (<i>numberinter</i>)만큼의 지점 정보를 <i>xlinear1</i> 에 저장한다.
231	선형 보간에 사용할 앞 데이터 수 (<i>numberinter</i>)만큼의 파고 정보를 <i>vlinear1</i> 에 저장한다.

```

232         end
233         for k=1:2
234             xlinear2(k)=(endtt(i)+k);
235             vlinear2(k)=h1(endtt(i)+k);
236         end
237         xlinear=[xlinear1 xlinear2]';
238         vlinear=[vlinear1 vlinear2]';
239         xlinearq=[starttt(i):1:endtt(i)];
240         vlinearq=interp1(xlinear,vlinear,xlinearq,'linear');
241         for j=1:length(xlinearq)
242             h1L(xlinearq(j))=vlinearq(j);
243         end
244     else
245         xlinearq=[starttt(i):1:endtt(i)];
246         for j=1:length(xlinearq)
247             h1L(xlinearq(j))=h1(xlinearq(1)-1);
248         end
249     end
250 end
251 % end-point fixing algorithm
252 h1LEP=h1L;
253 for i=1:length(starttt)
254     number=endtt(i)-starttt(i)+1;
255     if number>=gapnumber
256         window=(endtt(i)-starttt(i)+1)*windowsize;
257         if starttt(i)-(npastdata*window)>=1
258             pastdata=h1L(starttt(i)-(npastdata*window):starttt(i)-1);
259         else
260             pastdata=h1L(1:starttt(i)-1);
261         end
262         nnn=length(pastdata);
263         targetwindow=pastdata(end-window+1:end);
264         targetwindow(end-number:end-1)=NaN;
265         targetdata=targetwindow;
266         targetdata(isnan(targetdata))=[];
267         [target,settingtarget]=mapminmax(targetdata);
268         for j=1:nnn-window*2-1
269             searchwindow=h1L(end-window*2-(j-1)+1:end-window-(j-1));
270             searchwindow(end-number:end-1)=NaN;

```

234	선형 보간에 사용할 뒤 데이터 수 (2개)만큼의 지점 정보를 <i>xlinear2</i> 에 저장한다.
235	선형 보간에 사용할 뒤 데이터 수 (2개)만큼의 파고 정보를 <i>vlinear2</i> 에 저장한다.
237	선형 보간에 사용할 데이터의 지점 정보를 <i>xlinear</i> 에 저장한다.
238	선형 보간에 사용할 데이터의 파고 정보를 <i>vlinear</i> 에 저장한다.
239	선형 보간으로 파고를 구해야하는 결측 지점 정보를 <i>xlinearq</i> 에 저장한다.
240	interp1: 작은따옴표 안의 방법으로 1차 보간을 수행한다. (예) <i>vlinearq</i> =interp1(<i>xlinear</i> , <i>vlinear</i> , <i>xlinearq</i> , <i>'linear'</i>); <i>xlinear</i> 에 해당하는 <i>vlinear</i> 로 <i>xlinearq</i> 에 해당하는 값을 선형 보간하여 <i>vlinearq</i> 에 저장한다.
247	결측 구간이 2칸 이상이지만 선형 보간 수행에 필요한 데이터 수를 충족하지 못하면 한 칸 전 값으로 대체한다.
254	결측 구간의 길이를 <i>number</i> 에 저장한다.
255	만약 결측 구간의 길이 (<i>number</i>)가 <i>gapnumber</i> 이상이면 끝점 연결 기법을 수행한다.
256	끝점 연결 기법의 창 크기 (<i>window</i>)를 결측 구간의 길이와 <i>window</i> size를 곱한 크기로 설정한다.
257	만약 끝점 연결 기법의 <i>pastdata</i> 가 데이터 범위를 벗어나지 않으면 아래를 수행한다.
258	<i>pastdata</i> 는 창 크기 (<i>window</i>)와 <i>npastdata</i> 를 곱한 크기를 갖고 결측 구간 시작점 (<i>startt</i>) 앞에 위치한다.
260	만약 끝점 연결 기법의 <i>pastdata</i> 가 데이터 범위를 벗어나면 <i>pastdata</i> 는 첫 칸부터 결측 구간 시작점 (<i>startt</i>) 앞까지로 설정한다.
264	선형 보간 되었던 결측 구간의 파고를 다시 NaN으로 변경한다.
266	<i>targetdata</i> 의 NaN 부분을 지운다.
269	<i>targetwindow</i> 앞에 같은 크기의 <i>searchwindow</i> 를 설정한다.

```

271         searchdata=searchwindow;
272         searchdata(isnan(searchdata))=[];
273         [search,settingsearch]=mapminmax(searchdata);
274         cf(j)=gfit2(target,search,'5');
275     end
276     mincf=min(cf);
277     iii=find(cf==mincf);
278     selectsearch=h1L(end-window-(iii-1)+1-(number+1):end- ...
279         window-(iii-1));
280     ystartoriginal=targetdata(end-1);
281     yendoriginal=targetdata(end);
282     ystartcompare=selectsearch(1);
283     yendcompare=selectsearch(end);
284     a=ystartoriginal-ystartcompare;
285     b=yendoriginal-yendcompare;
286     c=number;
287     for ccc=1:c
288         modified(ccc)=(a-b)/(1-c)*(ccc-c)+b;
289         h1LEP(startt(i)+ccc-1)=selectsearch(ccc+1)+modified(ccc);
290     end
291 end
292 end
293 totaldata3=[t1 h1LEP];
294 disp('Gap Filling Complete');
295 if switch3==2
296     saveascii(totaldata3,output3,sd)
297 end
298 elseif switch3==0
299     disp('Without Gap Filling');
300 else
301     disp('Check "switch3"');
302 end
303 %-----%
304 % DART algorithm %
305 %-----%
306 if (switchdart==1)||((switchdart==2)
307     if ((switch1==1)||((switch1==2))&&(switch2==0)&&(switch3==0)
308         data=totaldata;
309     elseif (switch2==1)||((switch2==2))&&(switch3==0)

```

273	mapminmax: 괄호 안 데이터의 열을 기준으로 데이터의 범위를 -1에서 1로 변경한다. (예) [search,settingsearch]=mapminmax(searchdata); searchdata의 열을 기준으로 데이터 범위를 변경하여 search에 저장하고 변경에 사용 한 계수들을 settingsearch에 저장한다.
274	gfit2: 괄호 안 첫 번째 값을 기준으로 2 번째 값에 대한 작은따옴표 내용의 적합도 검정 법을 수행한다. 적합도 검정 법 종류 (all: 모든 방법, 1: 평균 제곱오차, 2: 정규 평균 제곱오차, 3: 평균 제곱근 편차, 4: 정규 평균 제곱근 편차, 5: 평균 절대오차, 6: 평균 절대 상대오차, 7: 상관계수, 8: 결정계수, 9: 효율성 계수, 10: 최대 절대오차, 11: 최대 절대 상대오차) (예) cf(j)=gfit2(target,search,'5'); target과 search의 평균 절대오차를 cf에 저장한다.
276	min: 괄호 안 데이터의 최솟값을 계산한다.
277	find: 괄호 안의 조건을 만족하는 데이터의 지점 정보 (index)를 저장한다.
280	아래의 변수는 Figure 2.5.1.의 기호에 해당한다. ystartoriginal: h1, yendoriginal:h4, ystartcompare:h2, yendcompare:h3
288	EP 원리의 수식으로 ccc의 범위가 1부터 c여서 EP 원리의 수식과 표현 방식이 다르며 EP
289	로 구한 결측 구간 예측 파고가 hILEP에 저장된다.

```

310         data=totaldata2;
311     elseif (switch3==1)||(switch3==2)
312         data=totaldata3;
313     elseif (switch1==0)&&(switch2==0)&&(switch3==0)
314         data=load (inputnamedart);
315     else
316         disp('Check "swich1, switch2 and switch3"');
317     end
318     tori=data(:,1);
319     dtotal=data(:,2);
320     maxd=max(dtotal);
321     mind=min(dtotal);
322     runtime=length(dtotal)-((180+n)/x+1)+1;
323     for k=1:length(tori)
324         checkdart(k)=NaN;
325     end
326     W1=x/60+n/120+1;
327     W2=x/60+n/120+2;
328     W3=x/60+n/120+3;
329     w(1)=(W3*W2*W1)/6;
330     w(2)=(W3*W2)/2-(W3*W2*W1)/2;
331     w(3)=W3-(W3*W2)+(W3*W2*W1)/2;
332     w(4)=1-W3+(W3*W2)/2-(W3*W2*W1)/6;
333     for i=1:length(tori)
334         tttotal(i)=(tori(i)-tori(1))*24;
335     end
336     for ii=1:runtime
337         sectime=tttotal(1+(ii-1):((180+n)/x+1)+(ii-1));
338         secdata=dtotal(1+(ii-1):((180+n)/x+1)+(ii-1));
339         sum=0;
340         for i=1:1:4
341             mindata(5-i)=mean(secdata(1+360*(i-1):60+360*(i-1)));
342         end
343         for i=1:1:4
344             sum=sum+w(i)*mindata(i);
345         end
346         secdatapred(ii)=sum;
347         diff(ii)=abs(secdata(end)-secdatapred(ii));
348     % Alarm if the diff. is larger than threshold

```


320	max: 괄호 안 데이터의 최대값을 계산한다.																														
322	DART 알고리즘은 (180+n)분의 연속 데이터가 있어야 수행하므로 총 수행 횟수 (<i>runtime</i>)은 전체 데이터 길이 (<i>length(dttotal)</i>)에서 (180+n)분에 해당하는 데이터 길이를 뺀 값이다.																														
324	DART 알고리즘을 수행하지 못하는 구간에는 NaN을 출력하기 위해 저장한다.																														
326	<p>아래의 계산에 의해 DART 알고리즘의 계수를 구한다.</p> <table> <tr> <th>x</th><th>y</th><th>f'</th><th>f''</th><th>f'''</th><th></th></tr> <tr> <td>$t - \frac{n}{120} - 3 = x_0$</td><td>$\zeta\left(t - \frac{n}{120} - 3\right) = y_0$</td><td></td><td></td><td></td><td></td></tr> <tr> <td>$t - \frac{n}{120} - 2 = x_1$</td><td>$\zeta\left(t - \frac{n}{120} - 2\right) = y_1$</td><td>$\frac{y_1 - y_0}{x_1 - x_0} = y_1 - y_0$</td><td></td><td></td><td></td></tr> <tr> <td>$t - \frac{n}{120} - 1 = x_2$</td><td>$\zeta\left(t - \frac{n}{120} - 1\right) = y_2$</td><td>$\frac{y_2 - y_1}{x_2 - x_1} = y_2 - y_1$</td><td>$\frac{y_2 - 2y_1 + y_0}{2}$</td><td></td><td></td></tr> <tr> <td>$t - \frac{n}{120} = x_3$</td><td>$\zeta\left(t - \frac{n}{120}\right) = y_3$</td><td>$\frac{y_3 - y_2}{x_3 - x_2} = y_3 - y_2$</td><td>$\frac{y_3 - 2y_2 + y_1}{2}$</td><td>$\frac{y_3 - 3y_2 + 3y_1 - y_0}{6}$</td><td>단위</td></tr> </table> <p> $f(x) = y_0 + (x - x_0)f' + (x - x_0)(x - x_1)f'' + (x - x_0)(x - x_1)(x - x_2)f''' \leftarrow$ </p> <p> $W_3 = \left(\frac{X}{60} + \frac{n}{120} + 3\right) \quad W_2 = \left(\frac{X}{60} + \frac{n}{120} + 2\right) \quad W_1 = \left(\frac{X}{60} + \frac{n}{120} + 1\right) \quad \text{이라 하면,}$ </p> <p> $f\left(t + \frac{X}{60}\right) = y_0 + W_3(y_1 - y_0) + W_3W_2\frac{y_2 - 2y_1 + y_0}{2} + W_3W_2W_1\frac{y_3 - 3y_2 + 3y_1 - y_0}{6}$ $= \left\{1 - W_3 + \frac{W_3}{W_2} - \frac{W_3W_2W_1}{6}\right\}y_0 + \left\{W_3 - W_3W_2 + \frac{W_3W_2W_1}{2}\right\}y_1 + \left\{W_3W_2 - \frac{W_3W_2W_1}{2}\right\}y_2 + \left\{\frac{W_3W_2W_1}{6}\right\}y_3$ </p> <p> $\therefore \omega_0 = 1 - W_3 + \frac{W_3}{W_2} - \frac{W_3W_2W_1}{6} \quad \omega_1 = W_3 - W_3W_2 + \frac{W_3W_2W_1}{2} \quad \omega_2 = W_3W_2 - \frac{W_3W_2W_1}{2} \quad \omega_3 = \frac{W_3W_2W_1}{6}$ </p>	x	y	f'	f''	f'''		$t - \frac{n}{120} - 3 = x_0$	$\zeta\left(t - \frac{n}{120} - 3\right) = y_0$					$t - \frac{n}{120} - 2 = x_1$	$\zeta\left(t - \frac{n}{120} - 2\right) = y_1$	$\frac{y_1 - y_0}{x_1 - x_0} = y_1 - y_0$				$t - \frac{n}{120} - 1 = x_2$	$\zeta\left(t - \frac{n}{120} - 1\right) = y_2$	$\frac{y_2 - y_1}{x_2 - x_1} = y_2 - y_1$	$\frac{y_2 - 2y_1 + y_0}{2}$			$t - \frac{n}{120} = x_3$	$\zeta\left(t - \frac{n}{120}\right) = y_3$	$\frac{y_3 - y_2}{x_3 - x_2} = y_3 - y_2$	$\frac{y_3 - 2y_2 + y_1}{2}$	$\frac{y_3 - 3y_2 + 3y_1 - y_0}{6}$	단위
x	y	f'	f''	f'''																											
$t - \frac{n}{120} - 3 = x_0$	$\zeta\left(t - \frac{n}{120} - 3\right) = y_0$																														
$t - \frac{n}{120} - 2 = x_1$	$\zeta\left(t - \frac{n}{120} - 2\right) = y_1$	$\frac{y_1 - y_0}{x_1 - x_0} = y_1 - y_0$																													
$t - \frac{n}{120} - 1 = x_2$	$\zeta\left(t - \frac{n}{120} - 1\right) = y_2$	$\frac{y_2 - y_1}{x_2 - x_1} = y_2 - y_1$	$\frac{y_2 - 2y_1 + y_0}{2}$																												
$t - \frac{n}{120} = x_3$	$\zeta\left(t - \frac{n}{120}\right) = y_3$	$\frac{y_3 - y_2}{x_3 - x_2} = y_3 - y_2$	$\frac{y_3 - 2y_2 + y_1}{2}$	$\frac{y_3 - 3y_2 + 3y_1 - y_0}{6}$	단위																										
334	시간을 0부터 시작하고 (<i>-troi(1)</i>) 단위를 일에서 시간으로 변경 (*24)한다.																														
344	DART 알고리즘 원리 (3)의 수식에 해당한다.																														
347	관측값 (<i>secdata</i>)과 예측값 (<i>secdatapred</i>) 차이의 절댓값을 <i>diff</i> 에 저장한다.																														

```

349         if diff(ii)>=threshold
350             checkdart(ii+(180+n)/x)=1;
351             sprintf(' %s %s','alarm:',datestr(sectime(end)/24+tori(1)))
352         else
353             checkdart(ii+(180+n)/x)=0;
354         end
355     end
356     if switchdart==2
357         totaldatadart=[data checkdart'];
358         saveascii(totaldatadart,outputdart,sd)
359     end
360 % Plotting the results
361     figure(1)
362     subplot(2,1,1)
363     plot(tttotal(1:((180+n)/x+1)+runtime-1),dttotal(1:((180+n)/x+1)+runtime-1),...
364         '-co','MarkerSize',2)
365     hold on
366     plot(tttotal(((180+n)/x+1):((180+n)/x+1)+runtime-1),secdatapred(1:end),'r')
367     axis([min(tttotal(1:((180+n)/x+1)+runtime-1)) ...
368         max(tttotal(1:((180+n)/x+1)+runtime-1)) mind-10 maxd+10])
369     xlabel('time (hr)')
370     ylabel('wave height (cm)')
371     title('DART algorithm')
372     legend('measured height','predicted height')
373     text(0.1,maxd+5,['start time=',datestr(tori(1))], ...
374         'verticalalignment','bottom');
375     subplot(2,1,2)
376     plot(tttotal(((180+n)/x+1):((180+n)/x+1)+runtime-1),diff(1:end),'b')
377     hold on
378     plot(tttotal(1:((180+n)/x+1)+runtime-1),threshold,'r-')
379     axis([min(tttotal(1:((180+n)/x+1)+runtime-1)) ...
380         max(tttotal(1:((180+n)/x+1)+runtime-1)) 0 10])
381     xlabel('time (hr)')
382     ylabel('wave height (cm)')
383     legend('ABS(measured-predicted)','threshold')
384     disp('DART Algorithm Complete');
385 elseif switchdart==0
386     disp('Without DART Algorithm');
387 else

```

349	만약 <i>diff</i> 가 지진해일 검출 한계값 (<i>threshold</i>) 이상이면 아래를 수행한다.
350	지진해일이 검출 된 시점을 <i>checkdart</i> 에 1로 저장한다.
351	sprintf: 작은따옴표 안 형식에 맞춰 문자열을 명령 창에 출력한다.
	<div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p>Command Window</p> <p>ans =</p> <p>(예) alarm: 11-Mar-2011 15:10:30</p> </div>
353	지진해일이 검출 안된 시점을 <i>checkdart</i> 에 0으로 저장한다.
361	figure: 괄호 안의 번호에 해당하는 그림 창을 팝업 한다.
362	subplot(a,b,c): 그림을 a행 b열로 나누었을 때 c 번째 그림을 선택한다.
363	x축에 <i>ttotal</i> , y축에 <i>dtotal</i> 인 그래프를 청록색 선과 크기 2 (MarkerSize)의 원 기호 (-co)로 그린다.
365	hold on: 위에서 수행한 결과를 지우지 말고 유지한다. (↔ hold off)
366	예측 파고 (<i>secdatapred</i>)를 빨간색 선 (r)로 그린다.
367	axis([a b c d]): 그림의 x축을 a에서 b로, y축을 c에서 d로 한정한다.
369	xlabel: 괄호 안의 값으로 x축 제목을 쓴다.
370	ylabel: 괄호 안의 값으로 y축 제목을 쓴다.
371	title: 괄호 안의 값으로 그림의 제목을 쓴다.
372	legend: 괄호 안 순서대로 사용한 범례의 이름으로 쓴다.
373	text: 지정한 그림의 일부분에 글자를 쓴다.
	<p>(예) text(0.1,maxd+5,['start time=',datestr(tori(1))],'verticalalignment','bottom');</p> <p>x축이 0.1이고 y축이 <i>maxd</i>+5인 지점에 [] 안의 글을 바닥 (bottom)기준으로 쓴다.</p>

```

388     disp('Check "switchdart");
389 end
390 %------%
391 % Slope algorithm %
392 %------%
393 if (switchslope==1)|| (switchslope==2)
394     if ((switch1==1)|| (switch1==2))&&(switch2==0)&&(switch3==0)
395         dataslope=totaldata;
396     elseif (switch2==1)|| ((switch2==2))&&(switch3==0)
397         dataslope=totaldata2;
398     elseif (switch3==1)|| (switch3==2)
399         dataslope=totaldata3;
400     elseif (switch1==0)&&(switch2==0)&&(switch3==0)
401         dataslope=load (inputnameslope);
402     else
403         disp('Check "swich1, switch2 and switch3");
404     end
405     ttotalslope=dataslope(:,1);
406     dtotalslope=dataslope(:,2);
407     for k=1:length(ttotalslope)
408         alarmIS(k)=NaN;
409         alarmCF1(k)=NaN;
410         alarmCF2(k)=NaN;
411         alarmCF3(k)=NaN;
412     end
413     for ii=1:length(dtotalslope)-(tGTide+tTide+tG+tBS)
414         usingd=dtotalslope(ii:tGTide+tTide+tG+tBS+ii)';
415         usingt=ttotalslope(ii:tGTide+tTide+tG+tBS+ii)';
416         for i=1:length(usingd)-tIS
417             IST(i)=(usingd(i+tIS)-usingd(i))/(tIS*dt);
418         end
419         for i=1:length(IST)-tTide
420             Tideuns(i)=mean(IST(i:i+tTide));
421         end
422         for i=1:length(Tideuns)-tsm
423             Tide(i)=mean(Tideuns(i:i+tsm));
424         end
425         for i=1:length(Tide)
426             ISori(i)=IST(i+length(IST)-length(Tide))-Tide(i);

```

393	만약 switchslope이 1 또는 2로 설정되어 있으면 경사 알고리즘을 수행한다.
394	포맷 변경만 수행한 경우 포맷 변경 시계열인 <i>totaldata</i> 를 불러온다.
396	Spike 제거를 수행한 경우 spike 제거된 시계열인 <i>totaldata2</i> 를 불러온다.
398	결측 구간 파고 예측을 수행한 경우 결측 구간 파고가 예측된 <i>totaldata3</i> 를 불러온다.
400	포맷 변경, spike 제거, 결측 구간 파고 예측을 모두 수행 안한 경우 <i>inputnametslope</i> 이름의 외부 데이터를 불러온다.
407	경사 알고리즘을 수행하지 못하는 구간에는 NaN을 출력하기 위해 미리 작성한다.
417	경사 알고리즘 원리 (1)에 해당한다.
420	경사 알고리즘 원리 (2)에 해당한다.
423	경사 알고리즘 원리 (3)에 해당한다.

```

427     end
428     time(ii)=usingt(end);
429     IS(ii)=ISori(end);
430     BS1(ii)=(max(ISori(2:tBS+1))-min(ISori(2:tBS+1)))/2;
431     BS2(ii)=std(ISori(2:tBS+1))*(2^0.5);
432     BS3(ii)=max(abs(ISori(2:tBS+1)));
433     CF1(ii)=abs(IS(ii))/BS1(ii);
434     CF2(ii)=abs(IS(ii))/BS2(ii);
435     CF3(ii)=abs(IS(ii))/BS3(ii);
436     % Alarm if abs(IS) and CF are larger than threshold
437     if abs(IS(ii))>=ISthreshold
438         sprintf('%s%s','IS alarm at:',datestr(usingt(end)))
439         alarmIS((tGTide+tTide+tG+tBS)+ii)=1;
440     else
441         alarmIS((tGTide+tTide+tG+tBS)+ii)=0;
442     end
443     if CF1(ii)>=CFthreshold
444         sprintf('%s%s','CF1 alarm at:',datestr(usingt(end)))
445         alarmCF1((tGTide+tTide+tG+tBS)+ii)=1;
446     else
447         alarmCF1((tGTide+tTide+tG+tBS)+ii)=0;
448     end
449     if CF2(ii)>=CFthreshold
450         sprintf('%s%s','CF2 alarm at:',datestr(usingt(end)))
451         alarmCF2((tGTide+tTide+tG+tBS)+ii)=1;
452     else
453         alarmCF2((tGTide+tTide+tG+tBS)+ii)=0;
454     end
455     if CF3(ii)>=CFthreshold
456         sprintf('%s%s','CF3 alarm at:',datestr(usingt(end)))
457         alarmCF3((tGTide+tTide+tG+tBS)+ii)=1;
458     else
459         alarmCF3((tGTide+tTide+tG+tBS)+ii)=0;
460     end
461 end
462 if switchslope==2
463     totaldataslope=[dataslope alarmIS' alarmCF1' alarmCF2' alarmCF3'];
464     saveascii(totaldataslope,outputslope,sd)
465 end

```

429	경사 알고리즘 원리 (4)에 해당한다.
430	배경 경사 계산 방법 중 BS1에 해당한다.
431	배경 경사 계산 방법 중 BS2에 해당한다.
432	배경 경사 계산 방법 중 BS3에 해당한다.
433	경사 알고리즘 원리 (6)에 해당한다.
437	경사 알고리즘 원리 (7)에 해당한다. 만약 순간 경사 (IS)의 절댓값이 순간 경사 한계값 ($IS_{threshold}$) 이상이면 아래를 수행한다.
438	<div> <div>Command Window</div> <div>ans =</div> <div>CF2 alarm at :01-Mar-2011 03:28:40</div> </div> <p>(예)</p>
439	순간 경사 한계값에 의해 지진해일이 검출 된 시점을 $alarmIS$ 에 1로 저장한다.
441	순간 경사 한계값에 의해 지진해일이 검출 안 된 시점을 $alarmIS$ 에 0으로 저장한다.
443	$CF1$, $CF2$, $CF3$ 에 대해서도 위의 IS 와 동일하게 수행한다.
463	1열에 MATLAB 시간 ($dataslope(:,1)$), 2열에 파고 ($dataslope(:,2)$), 3열에 순간 경사에 의한 지진해일 검출 정보 ($alarmIS$), 4열에 조절 함수1에 의한 지진해일 검출 정보 ($alarmCF1$), 5열에 조절 함수2에 의한 지진해일 검출 정보 ($alarmCF2$), 6열에 조절 함수3에 의한 지진해일 검출 정보 ($alarmCF3$)의 행렬을 $totaldataslope$ 에 저장한다.

```

466 % Plotting the results
467     xxaxis=ttotalslope(length(dtotalslope)-length(IS)+1:end)*24-...
468         ttotalslope(length(dtotalslope)-length(IS)+1)*24;
469     xstart=xxaxis(1);
470     xend=xxaxis(end);
471     figure(2)
472     subplot(4,1,1)
473     plot(xxaxis,dtotalslope(length(dtotalslope)-length(IS)+1:end),...
474         '-co','MarkerSize',2)
475     maxslope=max(dtotalslope(length(dtotalslope)-length(IS)+1:end));
476     minslope=min(dtotalslope(length(dtotalslope)-length(IS)+1:end));
477     title('Slope algorithm')
478     legend('measured height(cm)')
479     axis([xstart xend minslope-10 maxslope+10])
480     subplot(4,1,2)
481     plot(xxaxis,IS,'b')
482     hold on
483     plot(xxaxis,ISthreshold,'r-')
484     hold on
485     plot(xxaxis,ISthreshold*(-1),'r-')
486     hold on
487     legend('IS(cm/sec.)','threshold')
488     axis([xstart xend ISthreshold*(-1)*3 ISthreshold*3])
489     subplot(4,1,3)
490     plot(xxaxis,BS1,'k')
491     hold on
492     plot(xxaxis,BS2,'g')
493     hold on
494     plot(xxaxis,BS3,'b')
495     legend('BS1(cm/sec.)','BS2(cm/sec.)','BS3(cm/sec.)')
496     axis([xstart xend 0 max(BS3)*2])
497     subplot(4,1,4)
498     plot(xxaxis,CF1,'k')
499     hold on
500     plot(xxaxis,CF2,'g')
501     hold on
502     plot(xxaxis,CF3,'b')
503     hold on
504     plot(xxaxis,CFthreshold,'r-')

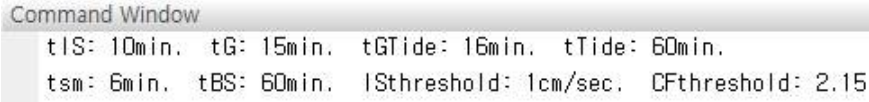
```


471	Slope 알고리즘 결과 그림은 DART 알고리즘 결과 그림 (1번)과 겹치지 않는 2번으로 연다.
472	4등분 한 그림 중 맨 위 그림을 선택한다.
473	관측값을 청록색 선과 크기 2의 원으로 그린다.
479	y축 범위를 관측값의 최솟값보다 10cm 작고, 최댓값보다 10cm 큰 값까지로 설정한다.
480	4등분 한 그림 중 위에서 2번 째 그림을 선택한다.
488	y축 범위를 순간 경사 한계값의 -3배에서 3배까지로 설정한다.
489	4등분 한 그림 중 위에서 3번 째 그림을 선택한다.
490	k는 검은색, g는 녹색, b는 파란색을 의미한다.
497	4등분 한 그림 중 맨 아래 그림을 선택한다.

```

505 xlabel('time (hour)')
506 legend('CF1','CF2','CF3','threshold')
507 axis([xstart xend 0 max([max(CF1),max(CF2),max(CF3)])*2])
508 disp(['tIS: ',num2str(tIS/6),'min.',' ','tG: ',num2str(tG/6), ...
509     'min.',' ','tGTide: ',num2str(tGTide/6),'min.',' ',' ...
510     'tTide: ',num2str(tTide/6),'min.']);
511 disp(['tsm: ',num2str(tsm/6),'min.',' ','tBS: ',num2str(tBS/6), ...
512     'min.',' ','ISthreshold: ',num2str(ISthreshold),'cm/sec.',...
513     ' ','CFthreshold: ',num2str(CFthreshold)]);
514 disp('Slope Algorithm Complete');
515 elseif switchslope==0
516     disp('Without Slope Algorithm');
517 else
518     disp('Check "switchslope"');
519 end
520 %------%
521 % Detide algorithm %
522 %------%
523 if (switchtide==1)|| (switchtide==2)
524     if ((switch1==1)|| (switch1==2))&&(switch2==0)&&(switch3==0)
525         datatide=totaldata;
526     elseif (switch2==1)|| ((switch2==2))&&(switch3==0)
527         datatide=totaldata2;
528     elseif (switch3==1)|| (switch3==2)
529         datatide=totaldata3;
530     elseif (switch1==0)&&(switch2==0)&&(switch3==0)
531         datatide=load (inputnametide);
532     else
533         disp('Check "swich1, switch2 and switch3"');
534     end
535     timetideori=datatide(:,1);
536     elevori=datatide(:,2);
537     totaldayori=length(timetideori);
538     q=1;
539     for p=1:totaldayori
540         if mod(p,(inter*60*60)/dt)==1
541             timetide(q)=timetideori(p);
542             elev(q)=elevori(p);
543             q=q+1;

```

508	MATLAB 명령 창에 경사 알고리즘에 사용한 변수값을 출력한다.
	
523	만약 switchtide가 1 또는 2로 설정되어 있으면 detide 알고리즘을 수행한다.
524	포맷 변경만 수행한 경우 포맷 변경 시계열인 <i>totaldata</i> 를 불러온다.
526	Spike 제거를 수행한 경우 spike 제거된 시계열인 <i>totaldata2</i> 를 불러온다.
528	결측 구간 파고 예측을 수행한 경우 결측 구간 파고가 예측된 <i>totaldata3</i> 를 불러온다.
530	포맷 변경, spike 제거, 결측 구간 파고 예측을 모두 수행 안한 경우 <i>inputnametide</i> 이름의 외부 데이터를 불러온다.
539	Detide에 사용할 간격으로 재 샘플링하는 과정이다.
540	<p>$\text{mod}(a,b)$: a를 b로 나누고 남은 나머지 값을 계산한다.</p> <p>조화분해에 사용할 데이터 간격 (<i>inter</i>)은 시간 단위이므로 $*60*60$으로 초 단위로 변경하고 원본 시계열 데이터 간격 (<i>dt</i>)로 나누어 나눌 개수를 구한다. 예를 들어 <i>inter</i>가 1/60, <i>dt</i>가 10이면 $(\text{inter}*60*60)/dt$는 6이다. 따라서 if $\text{mod}(p,(\text{inter}*60*60)/dt)=1$은 p를 6으로 나누어 나머지가 1인 경우에 아래를 수행한다. 즉, p가 1, 7, 13, ...인 경우를 채택한다.</p>

```

544         end
545     end
546     timetide=timetide';
547     elev=elev';
548     totalday=length(elev);
549     [nameu,fu,tidecon,pout,synth,ltype,stime]=t_tide(elev,'interval',...
550         inter,'start',timetide(1),'latitude',lat, 'synthesis',0);
551 % Change the output data interval
552     xout=t_predic(stime+[0:totaldayori-1]*dt/3600/24.0,...
553         nameu,fu,tidecon,'lat',lat,'synth',synth,'anal',ltype);
554     if switchtide==2
555         outtide=fopen(outputtide,'w');
556         for i=1:length(xout)
557             fprintf(outtide,'%f %f \n',timetideori(i),xout(i));
558         end
559     end
560 % Plot the results
561     figure(3)
562     plot(timetideori-timetideori(1),[elevori xout]);
563     hold on
564     plot(timetideori-timetideori(1),elevori-xout,'linewi',2,'color','r');
565     xlabel('Days');
566     ylabel('Elevation (cm)');
567     legend('Original Time series','Tidal prediction from Analysis',...
568         'Original time series minus Prediction');
569     title('DETIDE ALGORITHM');
570     disp('Detide Algorithm Complete');
571 elseif switchtide==0
572     disp('Without Detide Algorithm');
573 else
574     disp('Check "switchtide"');
575 end

```

549	원본 <code>t_tide.m</code> 함수는 <code>[nameu,fu,tidecon,xout]=t_tide(xin,varargin);</code> 형태이다. 즉 원본은 등호 왼쪽에 4가지만 출력되는데 KTDA v1.0에서는 재 샘플링을 통해 조화상수를 빠르게 계산하고 다시 원래 샘플링 간격으로 조석 성분을 예측하기 위하여 등호 왼쪽에 <i>synth</i> , <i>ltype</i> , <i>stime</i> 도 출력되도록 수정하였다. 참고로 <i>pout</i> 은 재 샘플링 된 간격으로 예측된 조석 성분 파고이다.
552	<code>t_predic.m</code> 함수는 조화 상수 정보를 바탕으로 조석 성분 파고를 예측하는 함수이다. MATLAB 함수 안의 변수는 함수 내에서만 저장되고 메인 코드에는 반영이 안 된다. 따라서 등호 오른쪽에 있는 <i>synth</i> , <i>ltype</i> , <i>stime</i> 이 549번 줄에서 출력되어 메인 코드에 변수 값이 저장되도록 수정하였다.
555	<code>fopen</code> : 저급 파일 입출력 함수로 <code>load</code> 로 입출력이 불가능한 경우에 사용한다. <code>saveascii.m</code> 이 없는 경우 이 함수를 이용하여 결과를 파일로 저장할 수 있다. (예) <code>outtide=fopen(outputtide,'w');</code> <code>outputtide</code> 이름의 파일을 쓰기용 (w)으로 연 것을 <i>outtide</i> 라 명한다.
557	<code>fprintf(a,','b)</code> : 작은따옴표 안 형식으로 b의 변수를 a에 입력한다. (예) <code>fprintf(outtide,'%f %f \n',timetideori(i),xout(i));</code> <code>outtide</code> 에 2개의 실수 (%f %f) 후 줄 변경 (\n) 형식으로 <i>timetideori</i> 와 <i>xout</i> 을 입력한다.
561	<code>Detide</code> 알고리즘 결과 그림은 <code>DART</code> 알고리즘 결과 그림, 경사 알고리즘 결과 그림과 겹치지 않는 3번으로 연다.

부 록

1. 컨트롤 파일: input.dat

001	%-----%
002	% SWITCH (0; No, 1; Yes, 2; Yes and save data) %
003	%-----%
004	switch1 = 0 : Format change
005	switch2 = 0 : Despike
006	switch3 = 0 : Fill gap
007	switchdart = 0 : DART algorithm
008	switchslope = 2 : Slope algorithm
009	switchtide = 0 : Detide algorithm
010	%-----%
011	% BASIC INFORMATION %
012	%-----%
013	dt = 10 : Time interval of data (sec)
014	sd = 10 : Significant decimals
015	%-----%
016	% FORMAT CHANGE (year,month,day,hour,minute,second) %
017	%-----%
018	starttime = 2011,3,1,0,0,0 : Start time
019	endtime = 2011,3,31,23,0,0 : End time
020	%-----%
021	% DESPIKE %
022	%-----%
023	inputnamesp = result10sec.dat : Use it if you already have timeseries data
024	thspike = 1/7.603 : Predetermined threshold to detect spikes
025	%-----%
026	% GAP FILLING %
027	%-----%
028	inputnamegap = despikes.dat : Use it if you already have timeseries data
029	numberinter = 5 : Number of data for interpolation
030	gapnumber = 30 : Minimum gap size for EP
031	window size = 2 : Total window size (*gap size)
032	npastdata = 10 : Total data for search (*gap size)
033	%-----%
034	% DART ALGORITHM %
035	%-----%

036	inputnamedart = gapfill.dat	: Use it if you already have timeseries data
037	n = 10	: Sampling time (min)
038	x = 10/60	: Predicting time (min)
039	threshold = 3	: Threshold to alarm tsunami (cm)
040	%-----%	
041	% SLOPE ALGORITHM %	
042	%-----%	
043	inputnameslope= gapfill.dat	: Use it if you already have timeseries data
044	tIS = 10*6	: Time interval for the average slope (*10 sec)
045	tG = 15*6	: Delay gap time for IBS (*10 sec)
046	tGTide = 15*6+1*6	: Delay gap time for detiding (*10 sec)
047	tTide = 60*6	: Time interval for detiding (*10 sec)
048	tsm = 6*6	: Time interval for long period wave effect (*10 sec)
049	tBS = 60*6	: Time interval for the background slope (*10 sec)
050	ISthreshold = 1	: ABS(IS) threshold (cm/sec)
051	CFthreshold = 2.15	: CF threshold (no dimension)
052	%-----%	
053	% DETIDE ALGORITHM %	
054	%-----%	
055	inputnametide= gapfill.dat	: Use it if you already have timeseries data
056	inter = 1/60	: Input data interval (hr)
057	lat = 37+29/60	: Sation latitude (degree)
058	%-----%	
059	% SAVE DATA %	
060	%-----%	
061	output = result10sec.dat	: Format changed data name
062	output2 = despiked.dat	: Despiked data name
063	output3 = gapfill.dat	: Gap filled data name
064	outputdart = dart.dat	: DART alarm data name
065	outputslope = slope.dat	: Slope alarm data name
066	outputtide = ttide.dat	: Detided data name
067	%-----%	

참고문헌

- Beltrami, G.M. (2008). An ANN algorithm for automatic, real-time tsunami detection in deep-sea level measurements. *Ocean Engineering*, 35(5), 572-587.
- Bressan, L. and Tinti, S. (2011). Structure and performance of a real-time algorithm to detect tsunami or tsunami-like alert conditions based on sea-level records analysis. *Natural Hazards and Earth System Science*, 11(5), 1499-1521.
- Goring, D.G. and Nikora, V.I. (2002). Despiking acoustic Doppler velocimeter data. *Journal of Hydraulic Engineering*, 128(1), 117-126.
- Mofjeld, H.O. (1997). Tsunami detection algorithm, unpublished paper. <http://nctr.pmel.noaa.gov/tda_documentation.html>
- Pawlowicz, R., Beardsley, B. and Lentz, S. (2002). Classical tidal harmonic analysis including error estimates in MATLAB using T_TIDE. *Computers & Geosciences*, 28(8), 929-937.
- 윤용훈, 서장원, 이덕기, 이호만, 오석훈, 김태희, 서태건, 김영아, 장유순, 이경아, 남재철, 황승언, 박원우, 조진대, 정성권, 서은진, 임교순 (2002). 해양기상관측 및 조사 연구 -서해 중부 해상 파랑 관측 결과 및 파고계 유지관리-. *기상연구소 사업보고서 (11-1360055-000151-14)*, 기상연구소 해양기상지진연구실, 6-9.

