

KIOST-TR-2022-0164

MATLAB을 사용한 수치모델에서의 태풍 경로
추출 알고리즘

The Typhoon Tracking Algorithm from numerical model
results using MATLAB



2022. 09. 15.

한 국 해 양 과 학 기 술 원

요 약

제 목	국문	MATLAB을 사용한 수치모델에서의 태풍 경로 추출 알고리즘		
	영문	The Typhoon Tracking Algorithm from numerical model results using MATLAB		
관련과제명	해양수치모델링과 지능정보기술을 활용한 해양예측 정확도 향상 연구 (PM62850)			
저 자 명	정 상 훈			
작성일자	2022년 09월 15일	공개	공개(O), 비공개()	
유형	기술동향자료(), 기술업무메뉴얼(O), 특허허지적재산권(), 결함보고서()			
주제분야	해양물리(O), 해양화학(), 해양생물(), 해양지질(), 해양공학() 해양정책(), 융복합()			
Keyword	typhoon, cyclone tracking, cyclone identification, MPAS-A, algorithm			
초록	<p>MATLAB을 사용하여 해양기상 수치모델로부터 태풍 경로를 자동 추출하는 알고리즘을 제시하였다. Mean Sea Level Pressure(MSLP)와 850 hPa 지위고도 Relative Vorticity(RV)를 활용하여 태풍의 중심 특성을 정의하였다. 등압선이 폐곡선 형태를 보이는 기압 패턴 중 RV가 $3 \times 10^{-5} \text{ s}^{-1}$ 이상인 영역이 20% 이상 포함되는 패턴을 저기압 혹은 태풍으로 정의할 수 있다. 노이즈 제거를 위하여 MSLP 및 RV에 대한 평활화가 필요하며 추적 목적에 부합하는 window size로 $6.75^\circ \times 6.75^\circ$ 를 제안하였다. 저기압 추적을 위해 필요한 저기압의 크기 산정 기준으로 면적 $3.14 \times 10^6 \text{ km}^2$(=반지름 약 1,000 km)를 제안하였다. 저기압의 추적은 저기압 영향 범위의 중첩 및 병합 여부, 6시간 이내 이동거리 500 km 미만인 조건을 고려하였다. 알고리즘 활용의 예시로 KOOS (Korea Operational Oceanographic System)의 전지구 해양기상 예측시스템으로부터 2021년 14호 태풍 찬투의 진로를 추출, best track과 비교하였다.</p>			

목 차

1. 서론	1
2. 재료 및 방법	2
3. 결과 및 토의	21
4. 결론	23
5. 참고문헌	24

약어 및 기호 정의

KOOS	Korea Operational Oceanographic System
MPAS	Model for Prediction Across Scales
JTWC	Joint Typhoon Warning Center
NCEP	National Centers for Environmental Prediction
MSLP	Mean Sea Level Pressure
RV	Relative Vorticity

1. 서론

태풍은 우리나라에서 발생하는 자연재해 중 가장 큰 비중을 차지하는 기상 현상 중 하나이며, 2019년에는 강풍, 호우, 폭풍해일 등에 의한 직간접적인 재산 피해는 약 98%를 차지하였다. (국민안전처, 2020) 따라서 태풍에 의한 인명 및 재산피해를 방지하고 대비하기 위해 태풍의 진로 및 강도 예측이 필수적이며 이에 대한 연구는 꾸준히 진행되고 있다.

한국해양과학기술원 해양재난재해연구센터에서 개발 및 개선하고 있는 운용해양예측시스템 (Korea Operational Oceanographic System, KOOS)에서는 해양기상 및 해수유동 예측시스템을 통해 기상, 유속, 파랑, 해수면, 수온, 염분 등의 기상 및 해양요소에 대한 실시간 예측정보를 생산하고 있다. 또한 태풍 발생 시 한국 연안에 대한 폭풍해일 및 파랑 예측결과를 유관 기관에 제공하고 있다.

태풍 관측 및 예측에서 중점적으로 사용되는 인자는 진로, 최대풍속, 중심기압, 강풍반경 등이며 태풍 예측정확도를 평가하는 기준도 이와 동일하다. 이들 중 태풍 진로를 결정하는 것에 필요한 태풍의 중심은 보통 기압의 중심, 또는 구름 시스템의 순환 중심으로 결정한다. 1970년대 Dvorak (1975) 기법의 제안과 함께 원격탐사 자료를 활용한 태풍 특성 도출에 대한 연구는 활발히 진행되어 왔다. 해상에서 발생 및 이동하는 동안에는 지상 관측자료를 활용할 수 없으므로 정지궤도 위성에서 수집되는 가시, 적외, 합성 영상, 그리고 층별 바람자료로부터 하층의 상세 바람자료를 획득하여 중심위치를 결정한다. 태풍이 해상에서 육상에 근접함에 따라 해상 및 육상 관측자료, 레이더 등이 태풍 중심위치 결정에 활용될 수 있다.

수치모델에서는 해면기압 예측결과를 통해 기압의 중심위치를 판별할 수 있다. 태풍을 포함한 저기압 추적 기법은 기후 연구 분야에서 널리 활용되고 있으며 그 종류도 다양하다. 저기압 추적 기법 크게 두 가지 기능을 포함하는데, 첫번째는 저기압을 매 시간 간격에서 정의(identification)하는 것, 두번째는 정의된 저기압 요소들을 연결하여 추적(tracking)하는 것이다. (Lu, 2017) 넓은 영역, 장기간의 데이터로부터 저기압 정보를 획득하기 위한 시간적 제약에 따라 일련의 과정들은 프로그래밍 된 알고리즘이 활용된다.

해양기상 모델의 태풍 시 예측정확도를 정량적으로 산출하는 것은 수치모델의 정확도 개선 정도를 추적하고, 이를 입력자료로 활용하여 산출되는 파랑과 폭풍해일 예측정확도의 검증을 위해 필수적이다. 이 기술보고서에는 태풍 best track과의 경로 오차 비교를 위한 선행연구로서, 해양기상 수치모델에서 태풍 경로를 MATLAB code를 활용하여 자동 추출하는 기법을 제시하고자 한다.

2. 재료 및 방법

가. 수치모델 및 관측자료

이 매뉴얼에서는 태풍 경로 알고리즘의 예시로 KOOS의 전지구 해양기상 예측모델인 MPAS-A (Model for Prediction Across Scales-Atmosphere) 예측결과와 우리나라 기상청 및 미국 JTWC (Joint Typhoon Warning Center)의 태풍 통보문을 관측자료로 사용하였다. 대상 태풍은 2021년 14호 태풍인 찬투(CHANTHU)이다.

MPAS-A는 그림 1과 같이 육각형 형태의 가변격자 체계를 채택하고 있다. 한반도를 포함한 북서태평양 영역에서는 약 15 km, 전구 영역에서 약 60 km 크기의 수평 해상도를 가지며 총 535554개 grid로 구성되어 있다. 수치모델은 매 3시간 간격, 7일 예측결과를 저장하며, 초기 및 경계조건으로 미국 NCEP(National Centers for Environmental Prediction)의 GFS(Global Forecast Model) 예측결과를 사용한다.

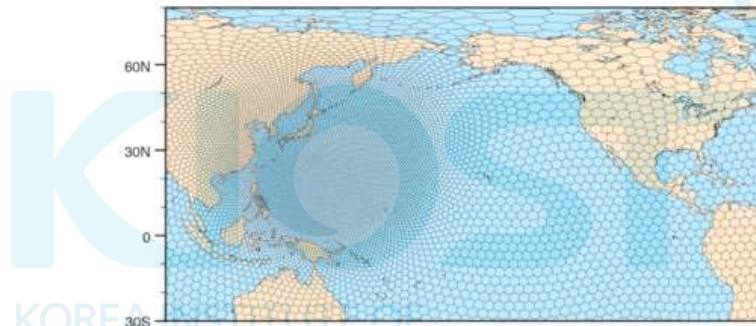


그림 1. KOOS MPAS-A의 수평 격자 체계

2021년 14호 태풍 찬투는 9월 7일 15.7° N, 136.6° E에서 발달하여 9월 17일 06시 경 제주 남부 해역을 통과하였다. (그림 2)

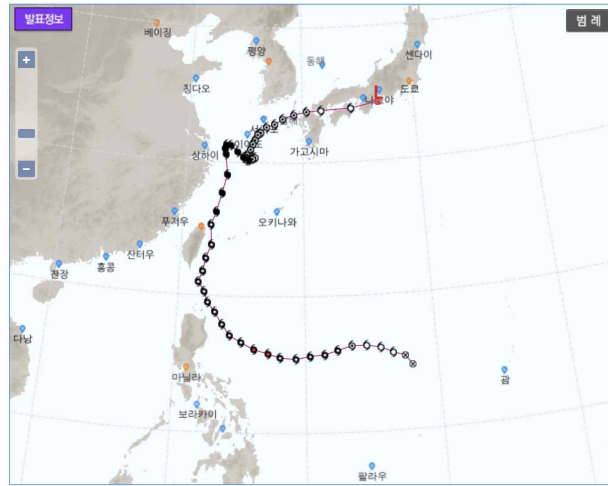


그림 2. 2021년 14호 태풍 찬투(CHANTHU)의 진로 (기상청)

나. Cyclone Identification

격자형 수치모델 자료로부터 저기압을 정의하기 위해서는 저기압성 회전, 즉 북반구에서의 반시계방향 회전의 특징을 검색하는 것이 선행된다. 기상학적 요소들 중 Mean Sea Level Pressure(MSLP), geopotential, meridional wind, temperature, vertical wind, vorticity 등이 활용되며, 요소의 선택은 추적하는 저기압의 규모 및 방법에 따라 서로 상이하다.

이 연구에서는 태풍의 중심에 대한 정의에 맞추어 MSLP를 사용하였으며 저기압성 회전을 정의하기 위하여 850 hPa 지위고도에서의 Relative Vorticity (RV)를 사용하였다. 태풍을 포함한 저기압에서는 MSLP가 폐곡선의 형태로 나타날 것을 예상할 수 있다. 그러나 MSLP의 폐곡선은 고기압성 회전에 의해서도 나타날 수 있다. 이를 구분하는 것으로 폐곡선 내부에서의 RV가 특정 임계치를 만족하는지를 통해 저기압성 회전을 구분한다. 일부 방법은 기존 선행연구를 수정하여 사용하였다 (Flaounas et al., 2014, Lu, 2017)

첫번째 단계로, 수치모델의 RV 및 MSLP를 평활화하는 단계를 거친다. 평활화는 검색 시 불필요한 small scale의 저기압을 제거하기 위해 필요하며, 평활화하고자 하는 window size는 대상 저기압의 규모에 따라 조절될 수 있다. 이 연구에서는 $6.75^\circ \times 6.75^\circ$ 영역의 window size를 제안한다. 만일 수치모델 해상도가 0.5° 라면, 13×13 grid가 적정 window size가 된다.

2D dimension에서의 moving average function은 다음과 같다. 공개된 open source를 활용하였다.

```

function matrixOut = smooth2a(matrixIn,Nr)
% MatrixIn : dataNr : rectangle of size (2*Nr+1)-by-(2*Nr+1)
N(1) = Nr; N(2) = Nr;= size(matrixIn);
eL = spdiags(ones(row,2*N(1)+1),(-N(1):N(1)),row,row);
eR = spdiags(ones(col,2*N(2)+1),(-N(2):N(2)),col,col);= isnan(matrixIn);
matrixIn(A) = 0;= eL*(~A)*eR;
nrmlize(A) = NaN;= eL*matrixIn*eR;
matrixOut = matrixOut./nrmlize;

```

알고리즘 코드에서는 다음과 같이 입력한다

```

sdata = smooth2a(rslp,ism);
rvdata = smooth2a(rv_grid,ism);

```

여기서 rslp는 수치모델의 MSLP, rv_grid는 850 hPa RV field이며 ism은 이 연구에서 6으로 설정하였다. 즉 13 x 13 window size에서 평활화된다. 이 연구에서 사용한 MPAS-A 모델은 가변격자체계를 사용하므로, 0.5° 간격의 직각격자로 내삽하는 과정을 거쳤다. 평활화된 MSLP 및 RV field는 각각 sdata, rvdata에 저장된다.

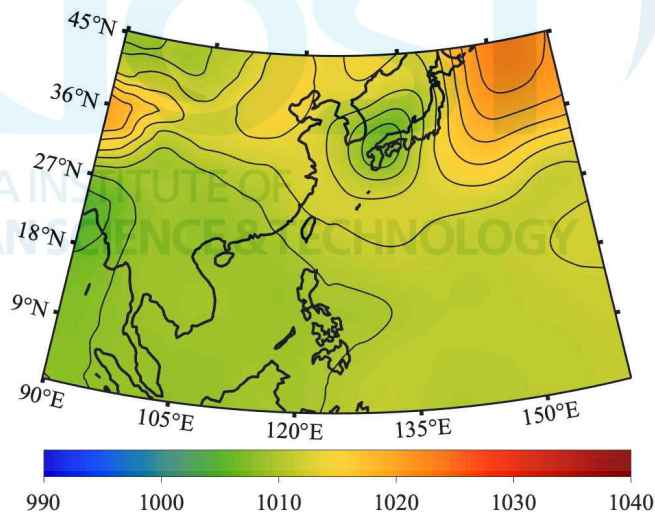


그림 3. 평활화된 수치모델 MSLP의 예시. 등압선은 2 hPa 간격으로 작성되었음.

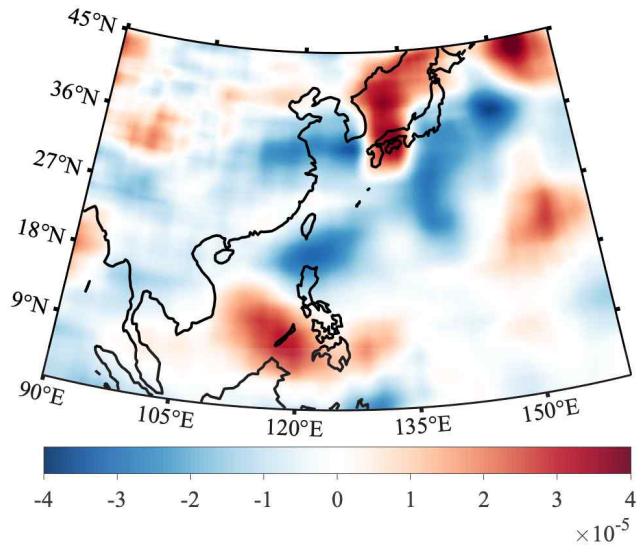


그림 4. 평활화된 수치모델 RV의 예시

두번째 단계로 MSLP field에 형성되는 폐곡선 형태의 등압선을 검색한다. 폐곡선을 추출하는 목적은 저기압성 회전에 의해 형성된 등압선을 검색하는 것이다. 다만 폐곡선은 고기압성 회전이나 특정 기상 현상과 관련 없이 영역 내에서 형성될 수 있다. 또한 저기압 시스템은 다수의 폐곡선 집합으로 나타날 수 있다. 폐곡선은 아래의 방법으로 검출한다.

```
[cf2 hf2]=contour(X,Y,sdata,[800:2:1200]);
ick2 = find(cf2(1,:)>800);
for i=1:length(ick2)
    x2{i} = cf2(1,ick2(i)+1:ick2(i)+1+cf2(2,ick2(i))-1);
    y2{i} = cf2(2,ick2(i)+1:ick2(i)+1+cf2(2,ick2(i))-1);
    p2(i) = cf2(1,ick2(i));
end
ipoly=[];
for i=1:length(ick2)
    if x2{i}(1)==x2{i}(end) && y2{i}(1)==y2{i}(end)
        ipoly=[ipoly;i];
    end
end
```

X, Y는 각각 수치모델의 longitude, latitude 격자정보이며 contour 함수의 결과로 출력되는 cf2에는 등압선의 기압, 선을 구성하는 점의 개수 및 위치정보를 포함한다. x2,

y2, p2 변수에는 분할된 각 등압선의 경도, 위도 및 기압 정보가 나누어 저장된다. ipoly 변수에는 시작점과 끝점이 동일한 등압선, 즉 폐곡선의 조건을 만족하는 등압선의 번호가 저장된다.

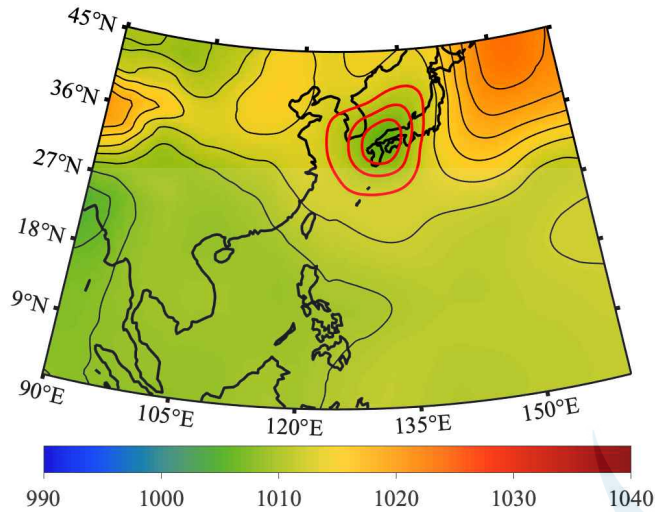


그림 5. 평활화된 MSLP 및 등압선. 폐곡선은 적색으로 표시하였음

폐곡선들 중 특정 기상 현상과 관련 없이 나타나는 요소들을 제거한다. 이 알고리즘에서는 경도방향, 위도방향 길이가 1,500 km 이상이거나 100 km 이하인 폐곡선을 제거하였다. 1,500 km는 저기압 시스템을 초과하는 크기를 고려하여 선정되었다. 장·단축 방향의 길이를 모두 확인하는 또 다른 이유는 장축이 단축보다 유의하게 큰 타원 형태의 폐곡선을 제거하기 위해서이다. 이러한 형태는 또한 폐곡선 내부의 중심기압 및 RV를 통한 저기압 여부를 판별할 수 있는 격자가 포함될 수 있도록 100 km 이하의 폐곡선을 제거하였다. 알고리즘에서는 아래와 같이 입력한다.

```
earthradius = almanac('earth','radius');
clear ick
ick=[];
for i=1:length(ipoly)
    [dx1,dn1]=max(x2{ipoly(i)});
    dy1 = y2{ipoly(i)}(dn1);
    [dx2,dn2]=min(x2{ipoly(i)});
    dy2 = y2{ipoly(i)}(dn2);
```

```

dist1 = distance(dy1,dx1,dy2,dx2,earthradius);
[dy3,dn3]=max(y2{ipoly(i)});
dx3 = x2{ipoly(i)}(dn3);
[dy4,dn4]=min(y2{ipoly(i)});
dx4 = x2{ipoly(i)}(dn4);
dist2 = distance(dy3,dx3,dy4,dx4,earthradius);
if dist1>1500
    ick(i)=0;
    continue
elseif dist2>1500
    ick(i)=0;
    continue
elseif dist1<100
    ick(i)=0;
    continue
elseif dist2<100
    ick(i)=0;
    continue
else
    ick(i)=1;
end
end

if isempty(ipoly)==0
    ipoly(ick==0)=[];
end

```

distance 함수는 위경도 단위의 좌표를 길이로 환산하기 위하여 사용되었다. 임계값을 만족하지 않는 폐곡선들은 ick 변수에 기록되며 폐곡선 집합의 목록인 ipoly에서 삭제된다.

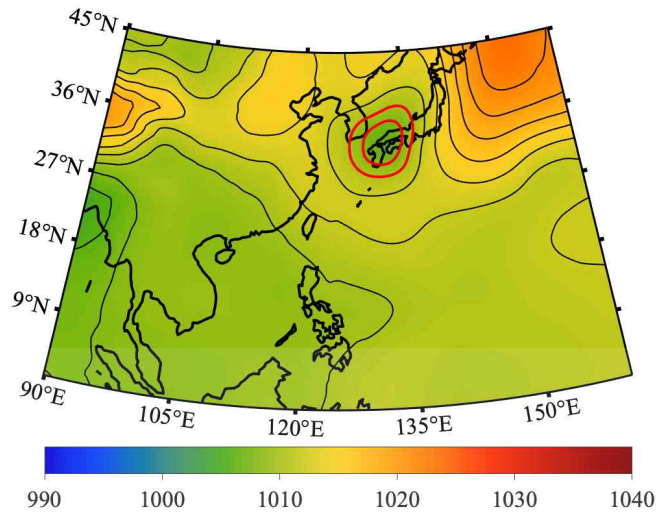


그림 6. 평활화된 MSLP 및 등압선. 폐곡선은 적색으로 표시하였음.

다음 단계로 하나의 저기압 시스템에 다수의 폐곡선이 존재하는지의 여부를 검색한다. 폐곡선의 목록인 ipoly에는 폐곡선의 번호만 저장된 상태이며, 알고리즘은 이 폐곡선들이 서로 다른 시스템인지 아닌지를 판별하지 못한다. 다수의 폐곡선으로 구성되어 있다면, 이를 하나의 시스템으로 간주하고 내측의 폐곡선만을 사용한다. 알고리즘에는 아래와 같이 입력한다.

```

clear ipoly2 ipoly3
ipoly2=ipoly;ipoly3=ipoly;
clear n2
for i=1:length(ipoly)
    for i2 = 1:length(ipoly)
        clear in on ick1 ick2
        [in on]= inpolygon(x2{ipoly(i)},y2{ipoly(i)},x2{ipoly(i2)},y2{ipoly(i2)});
        ick1=numel(x2{ipoly(i)}(in));
        ick2=numel(x2{ipoly(i)}(on));
        if ick1>0 && ick2==0
            n2(i2)=1;
        else
            n2(i2)=NaN;
        end
    end
end
end

```

```

    ipoly2(isnan(n2)==0)=NaN;
end
ipoly(isnan(ipoly2)==1)=[];
clear C
for i=1:length(ipoly)
    C{i}.ic_lon = x2{ipoly(i)};
    C{i}.ic_lat = y2{ipoly(i)};
    C{i}.ic_pres = p2(ipoly(i));
end

```

inpolygon 함수는 다각형이 다른 다각형의 내부 혹은 모서리에 존재하는 점의 개수를 반환한다. 출력 결과인 in은 내부 혹은 모서리에 존재하는 점, on은 모서리에 존재하는 점을 나타낸다. 폐곡선들을 서로 1:1로 대조하여 내부에 또 다른 폐곡선이 존재하면 순차적으로 삭제해 나가면 가장 내부의 폐곡선만 남게 되고, 이를 C 변수에 저장한다. C.ic_lon, C.ic_lat, C.ic_pres는 각각 폐곡선의 경도, 위도 및 등압선의 기압을 의미한다.

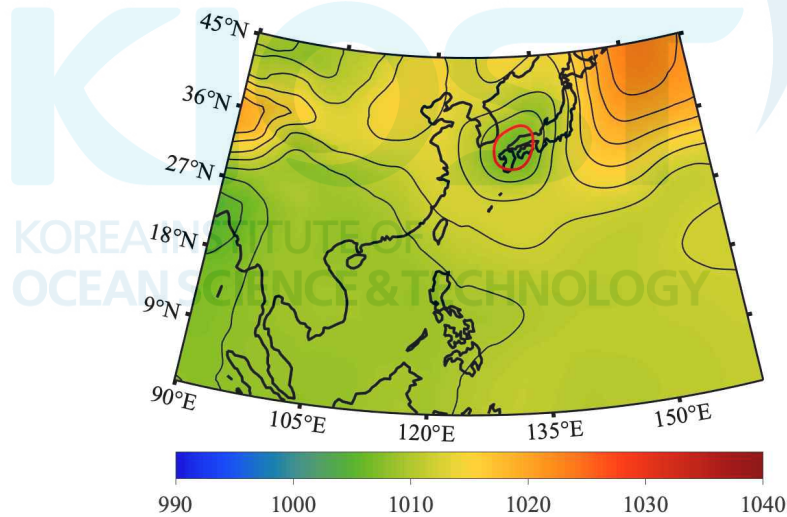


그림 7. 평활화된 MSLP 및 등압선. 적색 폐곡선은 저기압 시스템의 내측 중심을 의미함.

저기압의 외측 경계는 아래와 같이 검색 및 저장한다. 외측 경계는 내측 등압선을 둘러싸는 폐곡선들 중 면적이 $3.14 \times 10^6 \text{ km}^2$ 이하인 폐곡선을 선정한다. 이는 반지름이 1,000 km인 원의 면적을 가정한다.

```

arealim = 3.14e6;
for i=1:length(ipoly)
    clear ick ick_len ick_area
    ick=[];
    for i2=1:length(ipoly3)
        if ipoly(i)==ipoly3(i2);continue;end
        in = inpolygon(x2{ipoly(i)},y2{ipoly(i)},x2{ipoly3(i2)},y2{ipoly3(i2)});
        ick1=numel(x2{ipoly(i)}(in));
        if ick1>0
            ick=[ick;ipoly3(i2)];
        end
    end
    if isempty(ick)==1
        C{i}.oc_lon = x2{ipoly(i)};
        C{i}.oc_lat = y2{ipoly(i)};
        C{i}.oc_pres = p2(ipoly(i));
    else
        for i2=1:length(ick)
            ick_len(i2)=length(x2{ick(i2)});
            ick_area(i2) = areaint(y2{ipoly3(i2)},x2{ipoly3(i2)},earthradius);
        end
        ick_area(ick_area>arealim)=NaN;
        oick=find(ick_len==nanmax(isfinite(ick_len.*ick_area).*ick_len));
        if isempty(oick)==1
            C{i}.oc_lon = x2{ipoly(i)};
            C{i}.oc_lat = y2{ipoly(i)};
            C{i}.oc_pres = p2(ipoly(i));
        else
            C{i}.oc_lon = x2{ick(oick)};
            C{i}.oc_lat = y2{ick(oick)};
            C{i}.oc_pres = p2(ick(oick));
        end
    end
end
end

```

각각의 서로 다른 시스템으로 구성된 C 에서 저기압성 회전의 특성을 만족하는 시스템을 검출한다. 이 연구에서는 RV의 임계값을 $3 \times 10^{-5} \cdot s^{-1}$ 으로 제안하며, 이는

Flaounas et al. (2017)의 연구에서 제시된 것과 동일하다. 폐곡선 내부의 격자점에서 임계치를 만족하는 격자점이 최소 20% 이상 포함되면 저기압성 회전에 의해 형성된 폐곡 등압선으로 판별한다.

```

rvlim = 3e-5;
delc = [];
for i=1:length(C)
    in = inpolygon(X,Y,C{i}.ic_lon,C{i}.ic_lat);
    nck=find(in==1);
    pdata = rvdata(nck);
    nck2 = find(pdata>=rvlim);
    if length(nck2)<length(pdata).*0.2
        delc=[delc i];
    end
end
C(delc)=[];

```

그림 8의 흑색 점선은 RV filed에서 임계값을 만족하는 영역이다. 남중국해, 동해 및 캄차카 반도 남해역에서 임계값을 만족하는 영역이 나타나며, 이 중 폐곡선과 중첩되는 지역은 동해이다.

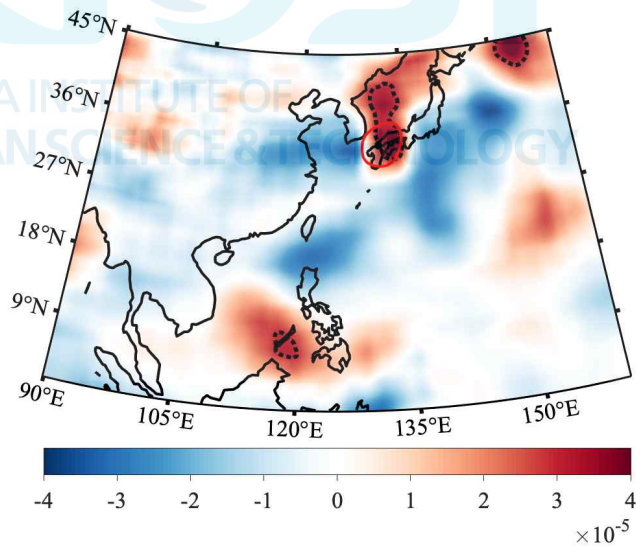


그림 8. 평활화된 RV. 임계치($3 \times 10^{-5} \text{ s}^{-1}$) 이상인 영역은 흑색 점선, 저기압 중심은 적색으로 표시하였음.

저기압으로 판별된 폐곡선들은 평활화하기 이전 MSLP 필드로부터 폐곡선 내부의 해면기압 최솟값을 검색하고, 그 격자를 태풍의 중심 위치 및 중심 기압으로 정의한다.

```

if isempty(ipoly)~=1
    for i=1:length(C)
        xm = find(X(:,1)>=min(C{i}.ic_lon) & X(:,1)<=max(C{i}.ic_lon));
        ym = find(Y(1,:)>=min(C{i}.ic_lat) & Y(1,:)<=max(C{i}.ic_lat));
        pdata = rslp(xm,ym);
        [n1 n2]=find(pdata==min(min(pdata)));
        C{i}.ilon=X(xm(n1),1);
        C{i}.ilat=Y(1,ym(n2));
        C{i}.ipres = pdata(n1,n2);
        C{i}.tim = tim;
    end
else
    C=[];
end
end

```

그림 9에는 영역에서 최종 산출된 저기압이 제시되어 있다. 이 사례에서는 1개의 저기압이 검출되었으며, 저기압의 외측 경계는 적색, 내측 경계는 청색, 그리고 저기압 중심에서의 최소기압은 997.8 hPa로 나타났다.

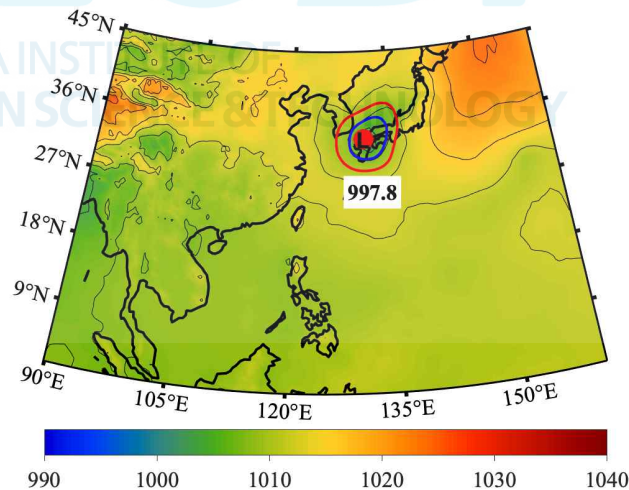


그림 9. 평활화되지 않은 수치모델 MSLP, 추출된 저기압의 내측 중심(청색), 외곽 경계(적색), 중심 위치 및 중심기압

저기압 정의 프로세스는 최소 6시간 간격으로 처리, 저장한다. 저기압 추적 프로세스에는 이동 거리가 고려되는데 이를 위해 최소한의 시간 간격이 요구된다.

다. Cyclone Tracking

저기압의 추적 단계에서는 매 6시간 간격으로 정의된 저기압들의 연결성을 확인한다. 이 연구에서는 6시간 전후로 저기압의 외측 경계가 겹쳐지는지, 겹쳐지지 않더라도 중심점 사이의 거리가 500 km 이하인지를 판별하여 이를 만족한다면 동일한 저기압으로 판별한다. 500 km는 저기압의 이동속도를 고려한 임계값이며 약 80 km/h 이하의 속도를 가정한다. 만일 두개 이상의 저기압이 이를 만족한다면, 거리차, 기압차를 통해 저기압이 병합(merging) 혹은 분할(splitting)을 판정한다. 저기압의 생애주기가 12시간 이하이면 일시적인 폐곡선으로 판정하여 추적하지 않는다.

추적하고자 하는 기간의 첫 번째 time에서 저기압들의 시작점을 정의한다

```
RES = cell(100,1);
CONT = cell(20000,1);
for i=1:length(C)
    RES{i}.n=1;
    RES{i}.lon=C{i}.ilon(end);
    RES{i}.lat=C{i}.ilat(end);
    RES{i}.ic_lon=C{i}.ic_lon;
    RES{i}.ic_lat=C{i}.ic_lat;
    RES{i}.oc_lon=C{i}.oc_lon;
    RES{i}.oc_lat=C{i}.oc_lat;
    RES{i}.pres=C{i}.ipres(end);
    RES{i}.tim=C{i}.tim;
end
```

시작점으로 활용할 C 저기압들은 RES cell에 임시 저장한다. RES에 저장되는 저기압은 매 time step 동안 저기압의 연결성을 판별하기 위해 활용된다. CONT는 추적이 완료된 저기압의 경로를 최종 저장할 때 사용한다.

저기압 추적의 시간 및 정확도를 향상시키기 위해 추적할 영역을 미리 정의하는 방법을 제안한다. 이 알고리즘은 목표로 하는 태풍을 포함하여 연구지역 내 모든 저기압을 정의 및 추적하기 때문에 사후에 원하는 태풍 경로를 지정해주는 과정이 필요하다. 영역을 제한하면 이 과정을 좀 더 원활히 수행할 수 있다.

```
dxlim = [90 160];
dylim = [5 45];
tn = [];
```

```

for i=1:length(C)
    if C{i}.ilon<dxlim(1) || C{i}.ilon>dxlim(2)
        tn=[tn i];
    end
end
C(tn)=[];
tn = [];
for i=1:length(C)
    if C{i}.ilat<dylim(1) || C{i}.ilat>dylim(2)
        tn=[tn i];
    end
end
C(tn)=[];

```

이 연구에서는 검색 범위를 $90^{\circ} E \sim 160^{\circ} E$, $5^{\circ} N \sim 45^{\circ} N$ 로 한정하고 중심위치가 이 범위를 벗어나는 저기압을 제외하였다.

저기압 추적의 우선 조건은 외측 경계의 중첩 여부이다. RES에 저장된 이전 time step의 저기압들과 load 된 현재 time step의 저기압들을 순차 비교하여 중첩 여부를 확인한다.

```

nck=zeros(100,1);
for j=1:length(nck)
    nn=isempty(RES{j});
    nck(j)=nn;
end
ick=find(nck==0);
c_code = zeros(length(C),1);
m=0;
cinf = [];
rinf = [];
for j=1:length(ick)
    for k=1:length(C)
        in = inpolygon(RES{ick(j)}.oc_lon,RES{ick(j)}.oc_lat,C{k}.oc_lon,C{k}.oc_lat);
        in2 = inpolygon(C{k}.oc_lon,C{k}.oc_lat,RES{ick(j)}.oc_lon,RES{ick(j)}.oc_lat);
        ick1 = numel(RES{ick(j)}.oc_lon(in));
        ickn = numel(C{k}.oc_lon(in2));
        if ick1>0 || ickn>0

```

```

        c_code(k) = j;
    end
end
end

```

ick에는 RES 변수 중 이전 time step에서 저장된 저기압들의 목록이 저장되고, 이를 모든 C와 중첩여부를 판별한다. C 저기압 중 k번째 저기압이 RES 중 j번째 저기압과 중첩된다면, c_code에 연결 정보를 저장한다. 검색이 종료되고 중첩과 관련한 연결성이 모두 확인되면, 중첩된 저기압을 연결한다.

```

for j=1:length(ick)
    if isempty(find(c_code==j, 1));continue;end
    if length(find(c_code==j))==1
        k=find(c_code==j);
        RES{ick(j)}.n=RES{ick(j)}.n +1;
        RES{ick(j)}.lon=[RES{ick(j)}.lon;C{k}.ilon(end)];
        RES{ick(j)}.lat=[RES{ick(j)}.lat;C{k}.ilat(end)];
        RES{ick(j)}.ic_lon=C{k}.ic_lon;
        RES{ick(j)}.ic_lat=C{k}.ic_lat;
        RES{ick(j)}.oc_lon=C{k}.oc_lon;
        RES{ick(j)}.oc_lat=C{k}.oc_lat;
        RES{ick(j)}.pres=[RES{ick(j)}.pres;C{k}.ipres(end)];
        RES{ick(j)}.tim=[RES{ick(j)}.tim;C{k}.tim];
        cinf = [cinf k];
        rinf = [rinf j];
    elseif length(find(c_code==j))>1
        k=find(c_code==j);
        clear dis pdis plon
        for ii=1:length(k)
            dis(ii) = distance(RES{ick(j)}.lon(end),RES{ick(j)}.lat(end),...
                C{k(ii)}.ilon(end),C{k(ii)}.ilat(end),earthradius);
            pdis(ii) = abs(RES{ick(j)}.pres(end)-C{k(ii)}.ipres(end));
            plon(ii) = C{k(ii)}.ilon(end)-RES{ick(j)}.lon(end);
        end
        tck=find(dis>500);k(tck)=NaN;
        tck=find(dis==min(dis));k=tck;
    if length(k)==1

```

```

RES{ick(j)}.n=RES{ick(j)}.n +1;
RES{ick(j)}.lon=[RES{ick(j)}.lon;C{k}.ilon(end)];
RES{ick(j)}.lat=[RES{ick(j)}.lat;C{k}.ilat(end)];
RES{ick(j)}.ic_lon=C{k}.ic_lon;
RES{ick(j)}.ic_lat=C{k}.ic_lat;
RES{ick(j)}.oc_lon=C{k}.oc_lon;
RES{ick(j)}.oc_lat=C{k}.oc_lat;
RES{ick(j)}.pres=[RES{ick(j)}.pres;C{k}.ipres(end)];
RES{ick(j)}.tim=[RES{ick(j)}.tim;C{k}.tim];
cinf = [cinf k];
rinf = [rinf j];

end
end
end

```

중첩관계가 확인된 저기압들은 RES의 각 변수에 중심위치, 기압, 시각 등의 정보를 연결하여 저장한다. 중첩되는 저기압이 2개 이상이라면, 중심점 사이의 거리를 통해 병합 여부를 판단한다. 중첩되지만, 이동속도의 임계값을 초과하는 저기압을 제외하고, 가장 가까운 저기압을 중첩으로 판단, 연결한다. cinf, rinf 변수에는 연결 및 병합에 사용된 저기압의 목록이 저장된다. 활용된 저기압들은 목록에서 삭제한다.

```

nvec = 1:length(C);
[a b c]=intersect(nvec,cinf);
nvec(b)=[];
rvec = 1:length(ick);
[a b c]=intersect(rvec,rinf);
rvec(b)=[];

```

중첩관계가 확인되지 않는 저기압들은 중심점 사이의 거리 임계치를 이어서 검증한다. 임계치를 수용하는 저기압들은 동일한 방법으로 RES 변수에 연결, 저장한다.

```

for j=1:length(rvec)
    for k=1:length(nvec)
        dis = distance(RES{ick(rvec(j))}.lon(end),RES{ick(rvec(j))}.lat(end),...
            C{nvec(k)}.ilon(end),C{nvec(k)}.ilat(end),earthradius);
        if dis<500 && RES{ick(rvec(j))}.lon(end)<C{nvec(k)}.ilon(end)
            RES{ick(rvec(j))}.n=RES{ick(rvec(j))}.n +1;

```

```

RES{ick(rvec(j))}.lon=[RES{ick(rvec(j))}.lon;C{nvec(k)}.ilon(end)];
RES{ick(rvec(j))}.lat=[RES{ick(rvec(j))}.lat;C{nvec(k)}.ilat(end)];
RES{ick(rvec(j))}.ic_lon=C{nvec(k)}.ic_lon;
RES{ick(rvec(j))}.ic_lat=C{nvec(k)}.ic_lat;
RES{ick(rvec(j))}.oc_lon=C{nvec(k)}.oc_lon;
RES{ick(rvec(j))}.oc_lat=C{nvec(k)}.oc_lat;
RES{ick(rvec(j))}.pres=[RES{ick(rvec(j))}.pres;C{nvec(k)}.ipres(end)];
RES{ick(rvec(j))}.tim=[RES{ick(rvec(j))}.tim;C{nvec(k)}.tim];
cinf = [cinf k];
rinf = [rinf j];
end
end
end
[a b c]=intersect(nvec,cinf);
nvec(b)=[];
[a b c]=intersect(rvec,rinf);
rvec(b)=[];

```

2단계의 연결성 검증 이후 남겨진 저기압들은 신규 생성된 저기압으로 판단, RES 변수 목록 중 빈자리에 저장, 다음 time step에서 연결성 검증 대상이 된다.

```

for j=1:length(nvec)
m = m+1;
RES{ick2(m)}.n=1;
RES{ick2(m)}.lon=C{nvec(j)}.ilon(end);
RES{ick2(m)}.lat=C{nvec(j)}.ilat(end);
RES{ick2(m)}.ic_lon=C{nvec(j)}.ic_lon;
RES{ick2(m)}.ic_lat=C{nvec(j)}.ic_lat;
RES{ick2(m)}.oc_lon=C{nvec(j)}.oc_lon;
RES{ick2(m)}.oc_lat=C{nvec(j)}.oc_lat;
RES{ick2(m)}.pres=C{nvec(j)}.ipres(end);
RES{ick2(m)}.tim=C{nvec(j)}.tim;
end
[a b c]=intersect(rvec,rinf);
rvec(b)=[];

```

현재 time step에서 연결 대상 저기압이 없는 경우, 소멸한 것으로 판단한다. 저기압 생애주기가 12시간, 즉 2 time step 이상인 저기압은 최종 결과물인 CONT 변수에 저장

하고, 저장이 완료된 저기압은 RES 변수 목록에서 삭제한다.

```
for j=1:length(rvec)
    if RES{ick(rvec(j))}.n>2
        ires = ires+1;
        CONT{ires}=RES{ick(rvec(j))};
    end
    RES{ick(rvec(j))}=[];
end
```

연결성 검증은 다음 time step에서 연결될 수 있는 저기압이 존재하지 않아야 종료된다. 그러므로 저기압 tracking이 연구기간동안 모두 검색된 이후에 여전히 RES 변수에 연결성 검토가 끝나지 않은 저기압이 남겨질 수 있다. 이러한 저기압이 언제 소멸할지는 알 수 없으나, 이를 최종적으로 저장하고 알고리즘을 종료하고 싶다면 시간 반복문 서두에 다음과 같은 코드를 추가한다.

```
fname = (filename of saved cyclones);
if exist(fname,'file')==0
    nck=zeros(100,1);
    for j=1:length(nck)
        nn=isempty(RES{j});
        nck(j)=nn;
    end
    ick=find(nck==0);
    rvec = 1:length(ick);
    for j=1:length(rvec)
        if RES{ick(rvec(j))}.n>2
            ires = ires+1;
            CONT{ires}=RES{ick(rvec(j))};
        end
        RES{ick(rvec(j))}=[];
    end
    continue
end
```

최종 결과물인 CONT 변수를 저장한 후 프로그램을 종료한다.

라. Eye Check

CONT 변수에는 기간 내 추적한 모든 저기압이 포함되어 있으므로, 이 중 원하는 태풍이 무엇인지는 직접 확인해야 한다. 그림 10에는 2021년 9월 5일 12시(UTC) 기준 7일 예측자료에 대한 저기압 추적 결과가 제시되어 있다. 적색 실선은 추출된 저기압, 각 실선에 표기된 번호는 CONT 변수에 저장된 순서이다. 발생 위치 및 기관별 예측 정보를 토대로 14번이 추적하고자 하는 저기압임을 알 수 있다. 8번 저기압과 같이 비정상적인 패턴으로 연결되는 저기압은 저기압 이동속도 임계치로 설정한 약 80 km/h를 만족하기 때문이다. 이 임계치는 태풍 이동 사례중 극치를 적용한 것이다. 추적하고자 하는 태풍이 일반적으로 관측되는 속도 범위를 가진다면, 임계치를 조정하여 노이즈를 제거할 수도 있다. 그림 11에는 6시간당 이동거리 150 km 미만을 임계치로 적용한 결과가 제시되어 있다. 여기에서는 10번 저기압이 추적하는 태풍으로 표기된다. 7번과 12번과 같이 연결이 의심되는 사례는 CONT 변수에 저장된 중심위치 시각 기록을 확인한다. 저기압의 발달 및 이동과정에서 6시간 사이에 유의미한 폐곡선이나 RV가 확인되지 않을 수 있다. 각 저기압의 시작시각과 종료 시각이 12시간이면 연결된 저기압으로 간주한다.

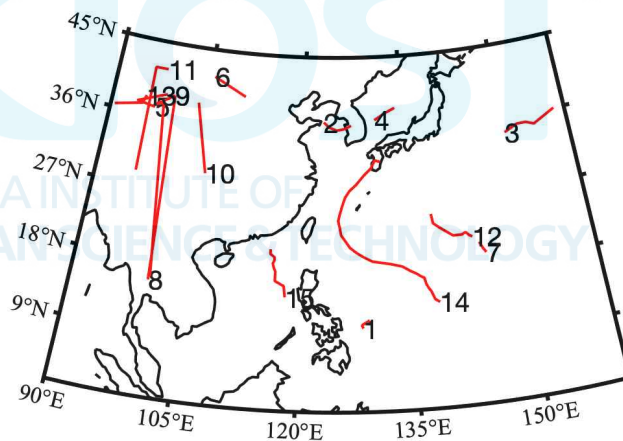


그림 10. 2021년 9월 5일 예측 기준 저기압 추적 결과. 태풍은 14번 저기압이며, 추적거리 임계값은 6시간당 650 km임

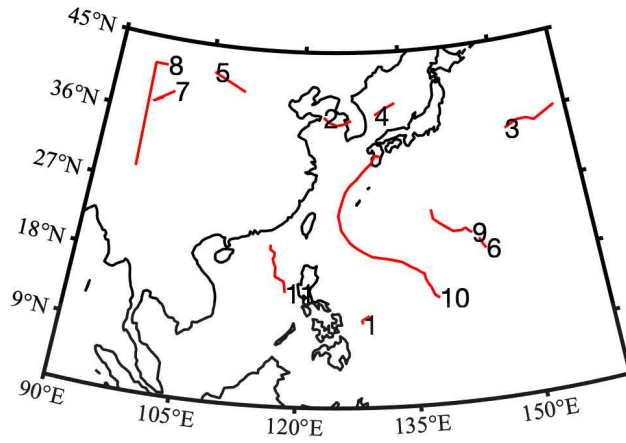


그림 11. 2021년 9월 5일 예측 기준 저기압 추적 결과. 태풍은 10번 저기압이며, 추적거리 임계값은 6시간당 150 km임

3. 결과 및 토의

가. 태풍 찬투 추출 결과 및 비교

그림 12에는 KOOS의 태풍 진로 예측결과와 KMA, JTWC의 관측 및 예측 진로가 제시되어 있다. KOOS의 경우 채워진 마커는 일별 예측결과 중 첫 번째 시각이며, 비워진 마커는 2021년 9월 16일 12시 기준 6시간 간격 예측결과이다. KMA, JTWC의 경우 채워진 마커는 매 태풍 통보문의 첫 번째 시각을 표기하였으며, 비워진 마커는 9월 16일 기준 예측결과이다. 태풍 찬투의 경우 9월 15일경 위도 약 30°, 경도 약 124° 부근에서 정체되는 특성이 나타났다.

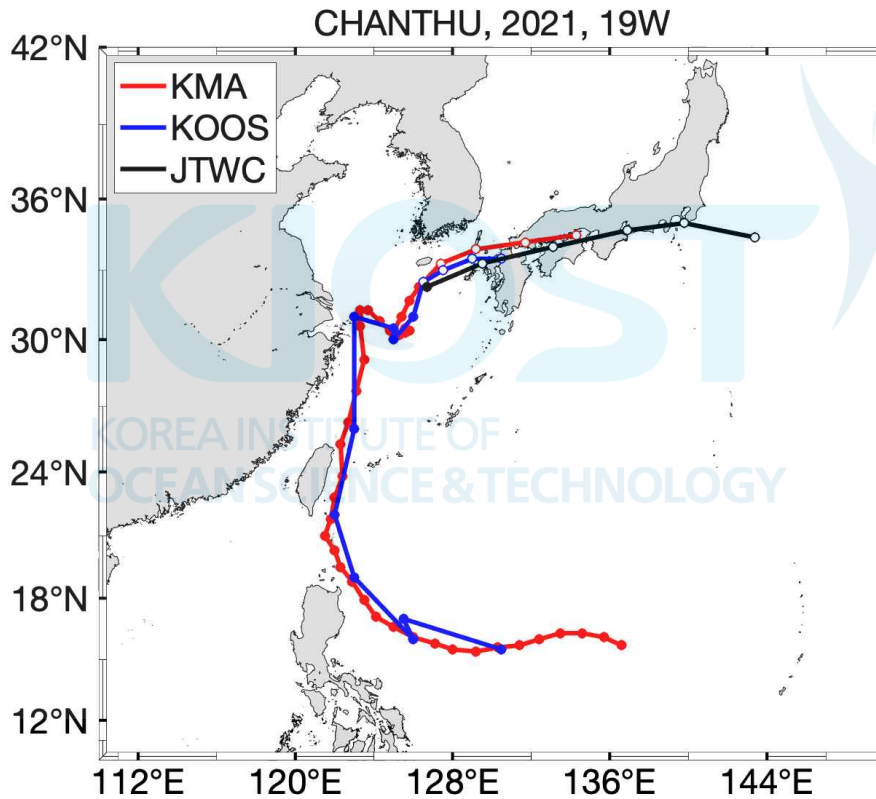


그림 12. 태풍 찬투 태풍 진로 예측 결과 (북서태평양 지역)

그림 13에는 한국 주변 상세 진로가 제시되어 있다. 태풍 정체지역에서의 위치가 수치모델에서도 유의하게 예측되는 것으로 보인다.

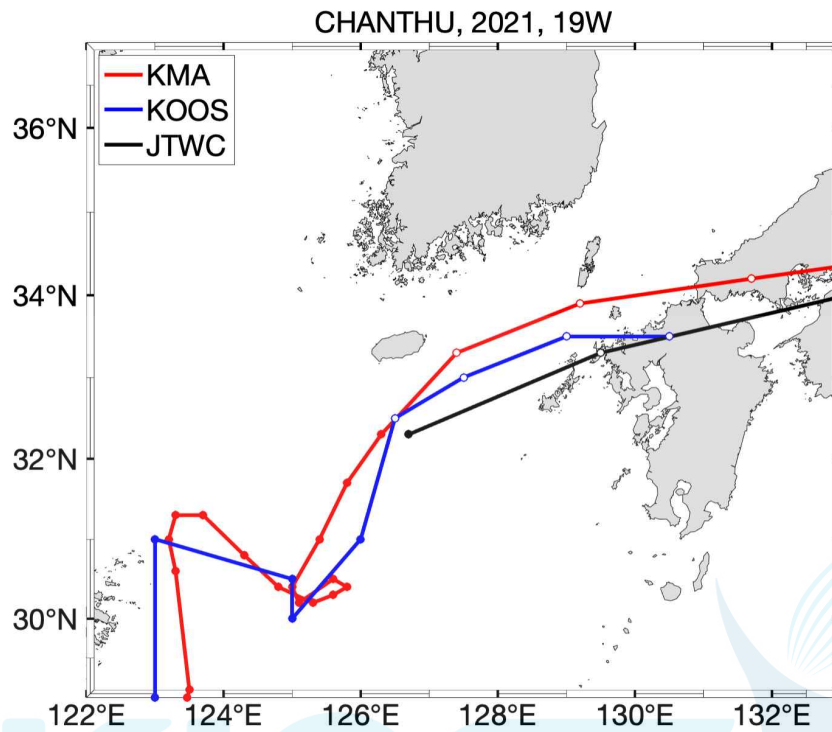


그림 13. 태풍 찬투 태풍 진로 예측 결과 (한반도 주변)

KIOST
KOREA INSTITUTE OF
OCEAN SCIENCE & TECHNOLOGY

4. 결론

이 기술업무 매뉴얼에서는 MATLAB을 사용하여 해양기상 수치모델로부터 태풍 경로를 자동 추출하는 알고리즘을 제시하였다. MSLP와 850 hPa 지위고도 RV를 활용하여 태풍의 중심 특성을 정의하였다. 등압선이 폐곡선 형태를 보이는 기압 패턴 중 RV가 $3 \times 10^{-5} \text{ s}^{-1}$ 이상인 영역이 20% 이상 포함되는 패턴을 저기압 혹은 태풍으로 정의할 수 있다. 노이즈 제거를 위하여 MSLP 및 RV에 대한 평활화가 필요하며 추적 목적에 부합하는 window size로 $6.75^\circ \times 6.75^\circ$ 를 제안하였다. 저기압 추적을 위해 필요한 저기압의 크기 산정 기준으로 면적 $3.14 \times 10^6 \text{ km}^2$ (=반지름 약 1,000 km)를 제안하였다. 저기압의 추적은 저기압 영향 범위의 중첩 및 병합 여부, 6시간 이내 이동거리 500 km 미만인 조건을 고려하였다.

이 알고리즘은 북반구에서 발생하는 태풍을 고려하여 설계되었으나, 일부 coefficient를 조정하면 일반적인 저기압 혹은 목적으로 하는 저기압의 특성에 맞추어 조정하여 활용할 수 있다. 예를 들어 태풍의 크기(외곽 경계)를 필요에 따라 엄밀히 설정할 경우 저기압의 영향 반경, 강풍 반경 등 태풍 연구에 필요한 요소들을 획득할 수 있다. 또한 저기압 특성 검출에 활용되는 요소들을 풍속, 기온 등 다른 성분으로 치환하고자 하는 경우, 일부 코드를 수정하면 작업에 필요한 기본 프로세스를 유지한 상태로 활용할 수 있다.

이 알고리즘에서는 태풍과 같은 대규모, 단일 기상 현상을 고려하기 때문에 저기압의 병합 및 분할을 판별하는 조건문의 정확도가 다소 떨어진다. 기후 등 장기 분석에서 이와 같은 요소가 필요할 경우 알고리즘의 개선이 요구된다. Cyclone tracking 알고리즘은 관련된 다수의 선행연구가 존재하며 그 방법도 다양하다. 다만 알고리즘을 실제 구현한 프로그램은 공개되어 있지 않다. 이 연구에서 제시한 알고리즘은 공통적으로 사용되는 절차가 구현되어 있으며, 관련 연구에 활용될 수 있을 것으로 기대한다.

참고문헌

- 국민안전처 (2020) 재해연보(2020), p 5
- Dvorak, V.F. (1975) Tropical Cyclone Intensity Analysis and Forecasting from Satellite Imagery, Monthly Weather Review, 103:420-430
- Flaounas, E., Kotroni, V., Lagouvardos, K., Flaounas, I. (214) CycloTRACK (v1.0) - tracking winter extratropical cyclones based on relative vorticity: sensitivity to data filtering and other relevant parameters. Geosci Model Dev, 7:1841-1853. doi:10.5194/gmd-7-1841-2014.
- Lu, C.H. (2017) A Modified Algorithm for Identifying and Tracking Extratropical Cyclones. Adv Atmos Sci 2017, 34:909-924. doi:10.1007/s00376-017-6231-2

