



## 저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

理學碩士學位論文

밀집쌍성계의 나선형  
공전운동으로부터  
방출된 중력파에 대한 통계적  
데이터 분석

仁濟大學校 大學院

컴퓨터應用科學科 理論物理學 專攻

金 正 初

指導教授 李 亨 源

밀집쌍성계의 나선형  
공전운동으로부터  
방출된 중력파에 대한 통계적  
데이터 분석

仁濟大學校 大學院

컴퓨터應用科學科 理論物理學 專攻

金 正 初

이 論文을 理學碩士論文으로 提出함

指導教授 李 亨 源

2015年 6月

金正初 의 理學碩士論文을 認定함.

委員長 명 연 수 印

委 員 이 창 환 印

委 員 이 형 원 印

仁濟大學校 大學院

2015年 6月

# 목차

• 국문초록 .....	i
• abstract .....	iii
I. 서론 .....	1
II. 연구목적 .....	6
III. 중력파 및 중력파 검출	
1. 중력파 및 중력파원 .....	7
2. 중력파형 .....	14
3. 중력파 검출 .....	18
IV. 밀집쌍성계의 중력파 모델	
1. 포스트 뉴토니안(post-Newtonian, pN) 방법론 .....	21
2. TaylorF2 모델 .....	24
3. 진폭향이 추가된 TaylorF2모델 .....	26
V. 마르코프 체인 몬테카를로(Markov Chain Monte Carlo) 모수추정	
1. 베이지언(Bayesian) 추론 .....	29
2. MCMC 모수추정 .....	31
VI. 연구결과 .....	33
VII. 결론 .....	53
VIII. 인사말 .....	55
IX. 참고문헌 .....	56
• 부록 .....	58

## 국문초록

### 밀집쌍성계의 나선형 공전운동으로부터 방출된 중력파에 대한 통계적 데이터 분석

김 정 초

인제대학교 대학원 컴퓨터응용과학과

중력파는 질량을 가진 물질이 가속할 때 발생하는 시공간의 일렁임이 파동으로 전파되는 것을 말한다. 아인슈타인의 일반 상대성 이론을 통하여 중력파의 존재는 예측되었지만 오늘날 까지 직접적인 중력파의 관측은 이루어지지 않았다. 중력파의 직접 검출을 위해 미국의 LIGO(Laser Interferometer Gravitational-wave Observatory)와 프랑스-이탈리아의 Virgo가 건설되어 가동되고 있고, 일본의 KAGRA(Kamioka Gravitational Wave Detector)가 건설 중에 있다. 이들은 모두 장거리 레이저 간섭계를 기반으로 하고 있으며, 수 십 헤르츠(Hz)에서 수 백 헤르츠 대역의 진동수를 갖는 중력파 검출에 최적화 되도록 설계되었다. 300Hz 부근에서 가장 민감하여 주로 중성자별이나 태양정도의 질량을 가지는 블랙홀에 의한 중력파를 측정하는데 적합하기 때문에 밀집쌍성계의 충돌(Compact Binaries Coalescence :CBC)에서 발생하는 중력파의 관측에 중점을 두고 있다. 밀집쌍성계는 별의 공전에 의한 병합 과정에 따라 크게 나선회전(inspiral)-병합(merge)-안정(ringdown)의 과정으로 구분된

다. 쌍성계의 나선형공전운동으로부터 발생하는 중력파형은 이론적으로 포스트-뉴토니언(post-Newtonian, pN) 방법으로 비교적 자세하게 기술될 수 있으며 이를 통해 정확한 파형공식을 유도할 수 있다. 검출된 중력파형에 포스트-뉴토니언(pN) 모델을 적용하여, 통계적으로 쌍성계의 성질을 특정 지을 수 있는 물리량을 측정한다.

이 논문에서는 PN 방법을 기반으로 중력파형 모델인 TaylorF2를 이용하여 중성자별-블랙홀(NS-BH)과 블랙홀-블랙홀(BH-BH) 쌍성계로부터 방출된 중력파를 컴퓨터로 시뮬레이션하여, 베이지언(Bayesian) 추론을 적용한 중력파 데이터 분석을 수행하였다. 특히 PN항에 의한 중력파형의 진폭변화가 물리적 매개변수의 모수추정에 미치는 효과를 분석하였다. 본 논문에서는 진폭의 효과를 2.5PN까지 고려하였으며, 가상의 중력파를 사용하여 그 효과를 분석하였다. 연구에서 사용한 소프트웨어는 기존의 라이고 과학협력단(LIGO Scientific Collaboration)이 작성한 중력파 데이터 분석용 소프트웨어인 LAL(LIGO Algorithmic Library)을 근간으로 하여 본 연구에 맞게 수정하였으며, 한국과학기술정보연구원(Korea Institute of Science and Technology Information, KISTI)의 슈퍼컴퓨터를 사용하였다.

진폭에 PN효과를 고려하여 모수추정을 하는 경우에 밀집쌍성계의 질량의 비가 1보다 큰 경우에는 중력파의 궤도 위상과 편광각을 PN효과를 고려하지 않은 것에 비해 명확하게 결정할 수 있으며, 중력파원의 천구상의 위치와 거리의 결정에는 큰 영향을 미치지 않음을 알 수 있었다. 따라서 본 논문에서 사용한 모수추정 방법으로 중력파원의 물리적 매개변수를 결정하는 경우에 목표로 하는 물리적 매개변수에 따라 올바른 중력파형 모델을 선택하는 것이 효과적이라는 것을 확인 하였다.

---

중심단어:중력파, 밀집쌍성계, 모수추정, MCMC, Bayesian통계, TaylorF2

## ABSTRCT

### Data Analysis for Compact Binary Coalescence Inspirial Gravitational Wave.

Kim Jeong Cho  
(Advisor : Prof. HyungWon Lee)  
Department of Computer Aided Science  
Graduate School, Inje University.

Gravitational-waves are ripples in the curvature of spacetime. These ripples are caused by the accelerated mass. The existence of gravitational-wave was predicted by the Einstein's general relativity. However, gravitational wave is not observed directly yet. In order to observe gravitational-wave directly, laser interferometer detectors such as the U.S Laser Interferometer Gravitational-wave Observatory(LIGO), the French-Italian Virgo were constructed and the Japanese KAGRA is under construction. It based on long-distance laser interferometer, designed to



optimize the gravitational wave detection with a frequency of several ten Hz to several hundred Hz range. The most sensitive in the near 300Hz is suitable mainly for gravitational waves by measuring neutron star or a black hole with the mass of the sun, which focuses on detection of gravitational waves generated in the Compact Binary Coalescence. Compact Binary is divided in three steps of inspiral-merge-ringdown according to the merge process of revolution of the star. Gravitational waveforms generated from inspiral step are theoretically described as post-Newtonian. Detected gravitational wave could be used to determine the characteristics of binary system through statistical analysis.

In this work, a simulated gravitational waves emitted from Neutron Star(NS) - Black Hole(BH) and Black Hole(BH) - Black Hole(BH) by using post-Newtonian formalism obtained gravitational waveform model TaylorF2 was used. Using a simulated gravitational wave, data analysis was performed applying Bayesian inference. Using the newly developed F2Amp includes amplitude corrections up to 2.5pN, the effect of amplitude corrections were analyzed. The analysis software is a modified version of LAL(LIGO Algorithm Library) package including amplitude correction effects and all calculations were performed on the dedicated KISTI(Korea Institute of Science and Technology Information) cluster.

Amplitude corrections allow us to constrain orbital phase and polarization. For more interesting parameters such as mass or

distance, amplitude corrections are unfortunately not very helpful. So depending on the physical quantity to the target when determining the physical quantity of the gravitational wave source by parameter estimation used in this paper, it is effective to select the correct waveform model.

---

keyword : gravitational wave, Compact binary,

Parameter Estimation, MCMC, Bayesian, TaylorF2

## 그림 목차

그림1. ....	2
중력파와 전자기파가 미치는 영역	
<a href="http://en.wikipedia.org/wiki/Gravitational_wave">http://en.wikipedia.org/wiki/Gravitational_wave</a>	
그림2. ....	3
중력파 존재를 간접적으로 증명	
<a href="http://en.wikipedia.org/wiki/PSR_B1913%2B16">http://en.wikipedia.org/wiki/PSR_B1913%2B16</a>	
그림3. ....	13
나선회전(Inspiral)-병합(Merge)-안정화(Ringdown) 과정	
NASA/CXC/GSFC/T.Strohmayer	
그림4. ....	17
그림 2 : 나선회전 단계에서 중력파 소스의 예	
Image: A. Stuver/LIGO	
그림5. ....	19
레이저 간섭계의 개략도	
<a href="http://en.wikipedia.org/wiki/Gravitational-wave_observatory">http://en.wikipedia.org/wiki/Gravitational-wave_observatory</a>	
그림6. ....	32
LAL에서 모수추정이 사용되는 과정을 도식화해 보여준다.	
그림7. ....	34
질량이 10Msun-1.4Msun인 블랙홀-중성자별의 파형	
그림8. ....	35
질량이 10.1Msun-9.9Msun인 블랙홀-블랙홀의 파형	
그림9. ....	37
블랙홀-중성자별의 TaylorF2의 모수추정 결과 - 1D	
그림10. ....	38
블랙홀-중성자별의 TaylorF2의 로그라이클리후드	

그림11.	.....	38
	블랙홀-중성자별의 TaylorF2의 스카이맵	
그림12.	.....	40
	블랙홀-중성자별의 TaylorF2Amp의 모수추정 결과 - 1D	
그림13.	.....	41
	블랙홀-중성자별의 TaylorF2Amp의 로그라이클리후드	
그림14.	.....	41
	블랙홀-중성자별의 TaylorF2Amp의 스카이맵	
그림15.	.....	43
	블랙홀-블랙홀의 TaylorF2의 모수추정 결과 - 1D	
그림16.	.....	44
	블랙홀-블랙홀의 TaylorF2의 로그라이클리후드	
그림17.	.....	44
	블랙홀-블랙홀의 TaylorF2의 스카이맵	
그림18.	.....	46
	블랙홀-블랙홀의 TaylorF2Amp의 모수추정 결과 - 1D	
그림19.	.....	47
	블랙홀-블랙홀의 TaylorF2Amp의 로그라이클리후드	
그림20.	.....	47
	블랙홀-블랙홀의 TaylorF2Amp의 스카이맵	
그림21.	.....	49
	블랙홀-중성자별의 진폭향에 따른 물리량의 상관관계	
그림22.	.....	51
	블랙홀-블랙홀의 진폭향에 따른 물리량의 상관관계	

## 표 목차

표1. 주입한 물리량 .....	33
관측된 중력파가 존재하지 않기 때문에 모수추정을 하기위해 시뮬레이션한 중력파를 나타내는 데 사용한 9개의 물리량	
표2. 정확도 계산값 .....	52
위향상을 동일하게 두고 진폭향의 뉴토니안인 경우와 진폭향이 포스트뉴토니안의 경우일 때 모수추정을 통해 결정한 물리량의 정확도	

## 영문용어 목차

GW	Graviational Wave
TT	Transverse-Traceless
LIGO	Laser Interferometer Gravitational-wave Observatory
LISA	Laser Interferometer Space Antenna
LSC	LIGO Scientific Collaboration
LAL	LSC Algorithm Library
CBC	Compact Binary Coalescence
BH	Black Hole
NS	Neutron Star
PN	Post-Newtonian
NR	Numerical Relativity
IMR	Inspiral Merge Ringdown
SPA	Stationary-Phase Approximation
ISCO	Innermost Stable Circular Orbit
PSD	Power Spectral Density
MCMC	Markov-Chain Monte-Carlo
SNR	Singnal to Noise Ratio

## I. 서론

중력파는 가속하는 질량에서 발생하는 시공간의 섭동으로, 아인슈타인의 상대성이론으로부터 처음 예측되었다. 아인슈타인의 상대성 이론에 의하면 전자가 진동할 때 전자기파가 발생하는 것과 비슷하게 중력파는 중력파원으로부터 빛의 속도로 퍼져나간다. 그러나 중력파는 전자기파와는 다르게 서로 45도의 각도 차이를 가진 편광을 가지고 있다. 또한 중력파는 전자기파와는 달리 세기가 매우 약하고, 질량을 통해서만 상호작용을 하기에 전자기파가 갈수 없는 우주 먼 곳까지 관측이 가능할 것이다 (그림 1). 이런 중력파를 관측하고자 하는 중요한 이유는 1905년 헤르츠에 의해 전자기파가 발견되어 전파망원경을 통해 이전에는 볼 수 없었던 우주를 보는 창이 넓어진 것처럼 중력파를 통해 전파망원경을 통해 볼 수 없는 우주를 볼 수 있는 새로운 창을 여는 것이기 때문이다.

중력파를 관측하기 위해 최초로 고안된 웨버바(Weber Bar)는 1960년대, 실험물리학자인 웨버가 사중편광의 성질을 이용하여 막대 검출기를 고안하였다. 하지만 실제 웨버바의 검출기 민감도는 중력파의 최소 주파수에 미치지 못하였다. 직접적으로 관측된 적이 없는 중력파의 존재는 1974년 조셉테일러(J. H. Taylor)와 러셀헨스(R. A. Hulse)가 발견한 PSR B1913+16(적경 : 19시 13분, 적위:+16도)라는 쌍성 맥동성(pulsar)으로부터 간접적으로 증명되었다. 이들은 쌍성 맥동성(pulsar)을 약 30년간 지속적으로 관측한 결과 궤도 변화를 발견하였고, 그 변화는 중력파에 의한 에너지와 각운동량의 감소율에 의한 것과 정확히 일치하는 것을 확인하였다 (그림 2). 이를 통해 중력파의 존재를 간접적으로 증명하였다. 하지만 중력파는 매우 미세하기 때문에 측정이 어려워 아직까지 직접적인 검출을 하지 못하고 있다.

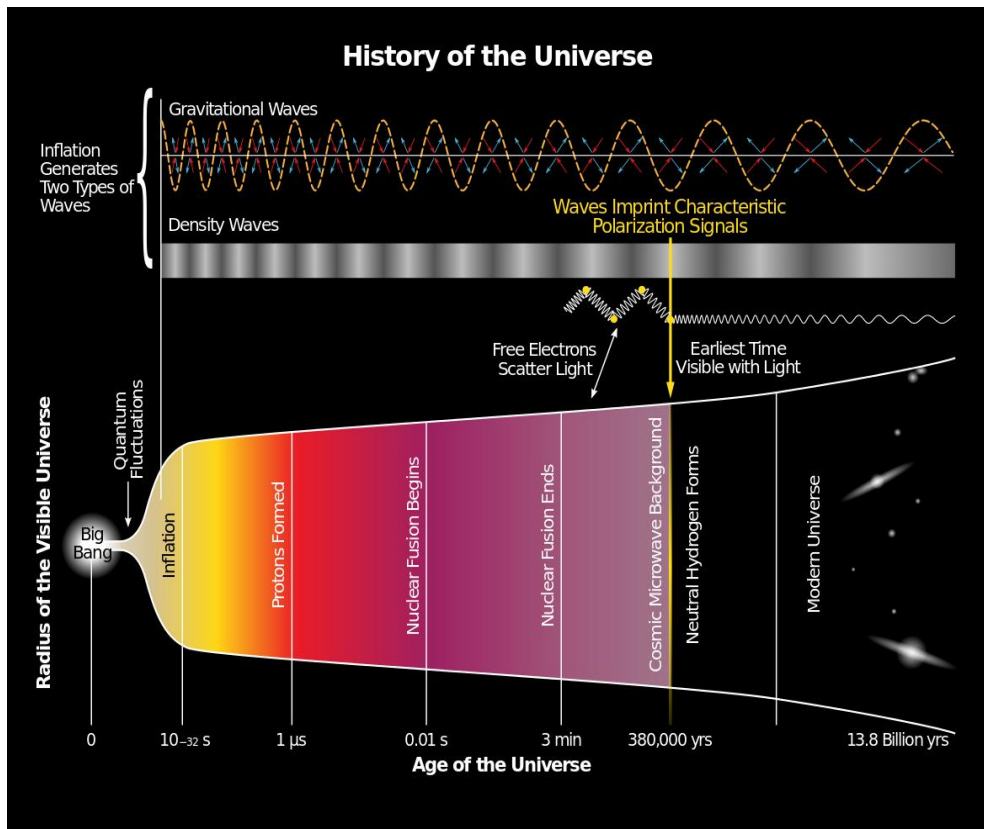


그림 1 : 중력파와 전자기파가 미치는 영역

[http://en.wikipedia.org/wiki/Gravitational\\_wave](http://en.wikipedia.org/wiki/Gravitational_wave)



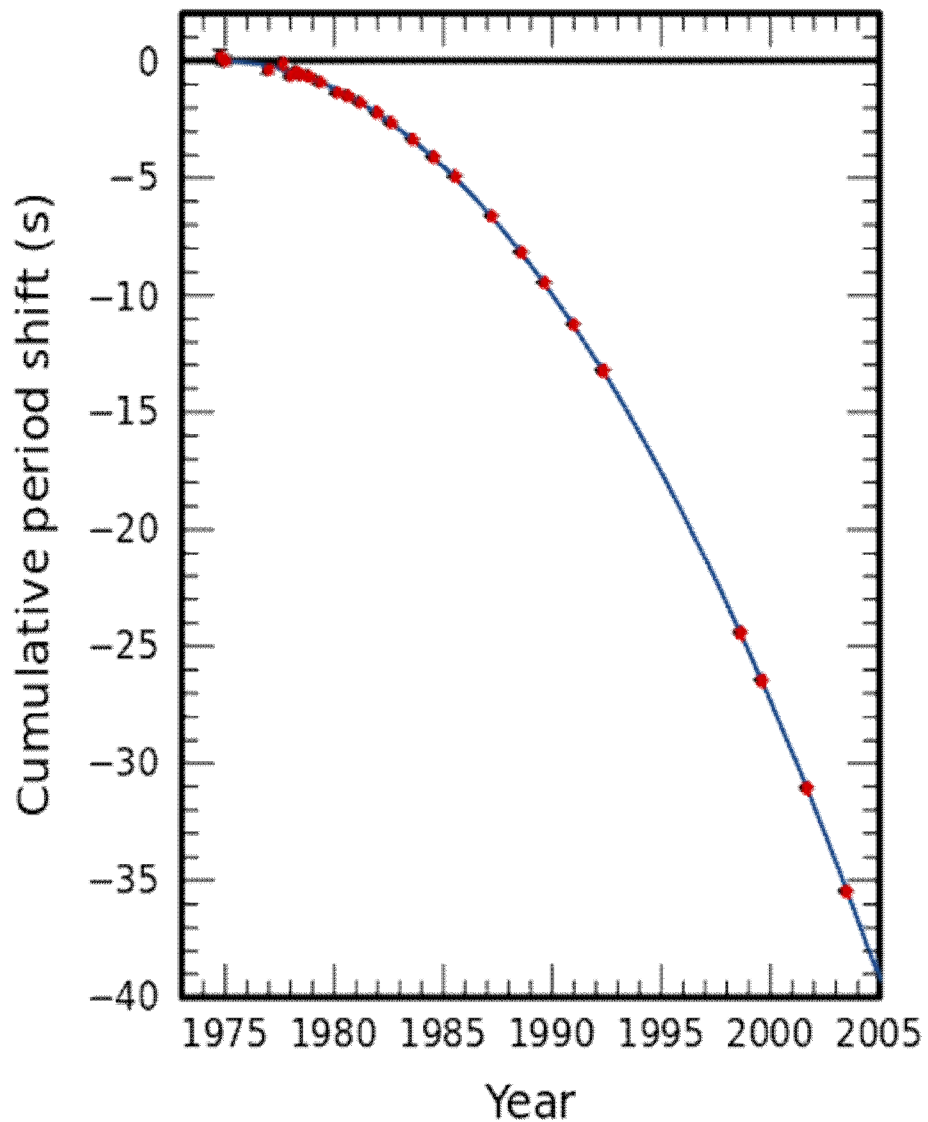


그림 2 : 포물선이 일반 상대성 이론에 따라 이론적으로 예상되는 변화를 나타내는 것(실선)과 PSR B1913+16의 궤도붕괴를 관찰하여 나타낸 것(점)으로 두 개가 일치함을 보임으로써 중력파의 존재를 간접적으로 증명함.

[http://en.wikipedia.org/wiki/PSR\\_B1913%2B16](http://en.wikipedia.org/wiki/PSR_B1913%2B16)

1990년대 캘리포니아 공대에 킵 쏜(Kip Thorne)교수가 새로운 개념의 검출기를 제안하였고 캘리포니아 공대의 로널드 드레버(Ronald Drever)와 MIT(Massachusetts Institute of Technology)의 레이너 와이즈(Rainer Weiss)가 구체적인 실험을 계획하였다. 이들이 생각한 검출기는 마이켈슨 간섭계라고 부르는 형태의 레이저 간섭계형 검출기이다. 이들의 제안으로 미국에서는 LIGO(Laser Interferometer Gravitational-wave Obsevatory)라 불리는 중력파 검출기를 미국의 루이지애나 주의 리빙스턴(Livingston)과 워싱턴주의 한포드(Hanford) 두 곳에 각각 양쪽 팔 길이가 4km에 달하는 레이저 간섭계 검출기를 건설하였다. 그 후 이태리와 프랑스가 합작한 레이저 간섭계형 검출기인 Virgo가 이태리의 카시나(Cascina)에 팔길이 3km의 검출기를 건설하였다. 두 개의 라이고(LIGO) 관측소와 버고(Virgo)관측소간의 거리차로 인해 중력파가 휩쓸고 지나갈 경우 시간차가 생기게 되는데, 이 시차를 계산하여 어느 위치에서 중력파가 왔는지를 추정할 수 있다.

중력파는 강한 중력장의 변화에 의해 만들어지기 때문에 유력한 중력파 원으로는 블랙홀(Black Hole)이나 중성자별(Neutron Star)과 같이 밀도가 매우 큰 천체 두 개로 이루어진 쌍성계이다. 쌍성이 내는 중력파의 강도는 궤도 반지름이 작아질수록 더 커진다. 이때 궤도의 변화 역시 궤도 반지름이 작아질수록 더 급하게 일어나 두 별은 충돌하게 된다. 쌍성이 궤도 운동을 하는 과정을 나선회전(inspiral), 중성자별 두 개가 합쳐지는 것을 병합(merge), 그리고 중성자별이 완전히 블랙홀로 만들어진 직후에도 중력장의 진동이 당분간 지속되지만 곧 약해지는 감쇠 단지동의 중력파를 내는 안정화과정(ringdown)이 있다. 병합(merge)과정은 별의 구체적인 구조와 물질의 상태방정식에 따라 달라지고 매우 복잡하기 때문에 수치 상대론적 계산을 통해서만 파형을 구할 수 있지만, 나선회전(inspiral)단계에서 나오는 중력파의 파형은 쌍성을 이루고 있는 별들의 질량만 알면 비교적

쉽고 근사적으로 계산할 수 있기 때문에, 이 논문에서는 나선회전 (inspiral)단계에서의 중력파형을 계산한다.

만약 중력파의 직접적인 관측이 이루어지면, 다음 단계는 가능한 한 많은 중력파원의 물리량을 찾아내는 것이다. 중력파의 물리량에는 쌍성의 질량, 위치 그리고 방향을 포함하고 있는데, 관측된 파형이 일반상대론의 예측에 가장 잘 맞는지에 대한 여부를 확인한다. 중력파가 관측될 때, 중력파형의 물리량추정에 사용될 마르코프 체인 몬테카를로(MCMC)는 이미 개발 되었다. 마르코프 체인 몬테카를로(MCMC)는 변하지 않는 확률분포를 갖는 임의의 상태공간에서 무작위행보를 통하여 얻어진 마르코프 체인을 사용하여 주어진 매개변수의 결정과 오차를 예측할 수 있다. 현재까지는 직접적으로 관측된 중력파의 데이터가 없기 때문에 이론적인 중력파형을 이용하여 시뮬레이션(simulation)한 데이터에 기계의 잡음을 섞어 만든 소프트웨어 주입(injection)을 사용한다. 소프트웨어 주입(injection)을 템플릿(template)파형과 비교하여 통계적으로 수렴할만한 결과가 나올 때까지 수백만에서 수 천만번 정도 반복한다. 이 때 나온 결과를 이용하여 물리량을 예측한다.

이 논문에서 사용하고 있는 중력파형은 라이고과학협력단(LSC)의 검출 파이프라인에서 사용하고 있는 주파수의 함수인 TaylorF2이다. 현실적인 진폭향에 대한 물리적인 효과를 추가하여, 진폭향의 효과가 파형과 모수추정의 물리량 측정에 어떠한 영향을 주는지 비교하였다.

## II. 연구목적

중력파와 중력파가 발생하는 중력파원에 대한 기본적인 이해를 바탕으로 전 세계적으로 중력파검출을 위한 많은 노력을 기울이고 있다. 향후 중력파의 직접적인 검출이 이루어질 경우, 중력파를 내는 파원의 모든 물리량을 데이터로부터 밝혀내기 위해 모수추정을 수행한다. 모수추정 결과를 통해 중력파원의 위치, 어떤 현상으로부터 중력파가 오는지, 그 천체의 물리량들을 통계학적으로 구한다.

본 논문에서는 라이고과학협력단(LSC)에서 사용하는 라이브러리인 LAL(LSC Algorithm Library)의 중력파형 모델중 하나인 TaylorF2 모델을 사용하였다. TaylorF2는 3.5pN의 위상항, 뉴토니안의 진폭항을 포함하는 주파수의 함수로서, 본 논문에서는 TaylorF2모델에 진폭항을 2.5pN까지를 추가하여 만든 TaylorF2Amp라는 모델을 LAL(LSC Algorithm Library)에 추가하여 진폭항의 변화에 따른 파형을 비교하였다. 또한 진폭항이 추가된 TaylorF2Amp 모델을 사용하여 모수추정을 수행하여, 진폭항의 pN효과가, 물리량 결정에 미치는 영향에 대해 조사한다.

중력파형 모델에 여러 고차원의 진폭항, 스핀 등의 물리적인 효과를 추가함으로써, 직접적인 중력파검출이 있을 때 예상한 중력파형과 일치하는지에 대한 검증의 확률이 커질 것이며, 다양한 중력파가 검출된다면 모수추정방법을 통하여 중력파원의 데이터로부터 모든 물리량을 밝혀낼 때, 찾고자 하는 물리량에 따른 중력파형 모델을 선택하는 것이 효과적이라는 것을 확인할 것이다.

### Ⅲ. 중력파 및 중력파 검출

#### 1. 중력파 및 중력파원

중력파는 중력장이 진동할 때 빛의 속도로 전파되는 것으로 아인슈타인의 일반 상대론에 의해 예측되었다. 아인슈타인의 장방정식은 다음과 같다.

$$G_{\mu\nu} \equiv R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R = \frac{8\pi G}{c^4}T_{\mu\nu}. \quad (1)$$

중력장에 의해 시공간이 휘어지는 정도를 나타내는 좌변과 그 시공간의 왜곡을 일으키는 물질간의 관계를 나타내는 방정식으로,  $G_{\mu\nu}$ 는 아인슈타인 곡률 텐서(Einstein curvature tensor),  $g_{\mu\nu}$ 는 4차원의 휘어진 시공간의 거리잡기 텐서(metric tensor),  $R_{\mu\nu}$ 는 리치텐서(Ricci tensor),  $R$ 은 리치스칼라(Ricci scalar),  $G$ 는 중력상수,  $c$ 는 빛의 속도, 그리고  $T_{\mu\nu}$ 는 스트레스-에너지 텐서(stress-energy tensor)이다. 시공간에서 리만 곡률 텐서(Riemann curvature tensor)의 정의는 다음과 같다.

$$R^\mu_{\nu\rho\sigma} = \partial_\rho \Gamma^\mu_{\nu\sigma} - \partial_\sigma \Gamma^\mu_{\nu\rho} + \Gamma^\kappa_{\nu\sigma} \Gamma^\mu_{\kappa\rho} - \Gamma^\kappa_{\nu\rho} \Gamma^\mu_{\kappa\sigma}. \quad (2)$$

여기에서  $\Gamma$  는 아핀연결(affine connection)로 정의는 다음과 같다.

$$\Gamma^\mu_{\nu\rho} = \frac{1}{2}g^{\mu\sigma}(\partial_\rho g_{\sigma\nu} + \partial_\nu g_{\sigma\rho} - \partial_\sigma g_{\nu\rho}). \quad (3)$$

리치텐서(Ricci tensor)인  $R_{\mu\nu}$ 는 리만 텐서(Riemann tensor)의 축약된 형태를 포함한다.

$$R_{\mu\nu} = R^\kappa_{\mu\kappa\nu}. \quad (4)$$

리치스칼라(Ricci scalar)인  $R$ 은 리치텐서(Ricci tensor)와 거리잡기 텐서(metric tensor)의 축약을 포함한다.

$$R = g^{\mu\nu} R_{\mu\nu}. \quad (5)$$

평평한 시공간일 경우,  $R_{\nu\rho\sigma}^\mu$ 는 0이 된다.

중력장이 약할 때, 민코프스키 거리잡기(Minkovski metric)  $\eta_{\mu\nu}$ 는

$$\eta_{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (6)$$

평평한 공간에서 중력에 의한 작은 섭동  $h_{\mu\nu}$ 을 가정 한다면 시공간의 거리잡기 텐서(metric tensor)는 다음과 같다. 이때 섭동의 크기는 매우 작다고 가정한다.

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}, \quad |h_{\mu\nu}| \ll 1. \quad (7)$$

$h^{\mu\nu} \equiv \eta^{\mu\alpha} \eta^{\nu\beta} h_{\alpha\beta}$ 로 두자. 위의 식을 이용하여  $h$ 를 1차로 하는 아핀연결(affine connection)은 다음과 같다.

$$\Gamma_{\nu\rho}^\mu = \frac{1}{2} \eta^{\mu\sigma} (\partial_\rho h_{\sigma\nu} + \partial_\nu h_{\sigma\rho} - \partial_\sigma h_{\nu\rho}). \quad (8)$$

간단하게 나타내기 위해  $h$ 는 1차로 유지한다. 리만 텐서(Riemann tensor), 리치 텐서(Ricci tensor), 리치 스칼라(Ricci scalar)의 선형화된 형태는 다음과 같다.

$$R_{\nu\rho\sigma}^\mu = \frac{1}{2} \eta^{\mu\kappa} (\partial_\rho \partial_\nu h_{\kappa\sigma} - \partial_\rho \partial_\kappa h_{\nu\sigma} - \partial_\sigma \partial_\nu h_{\kappa\rho} + \partial_\sigma \partial_\kappa h_{\nu\rho}), \quad (9)$$

$$R_{\mu\nu} = \frac{1}{2}(\partial_\mu \partial_\rho h_\nu^\rho - \square h_{\mu\nu} - \partial_\mu \partial_\nu h + \partial_\nu \partial_\rho h_\mu^\rho), \quad (10)$$

$$R = \partial_\mu \partial_\nu h^{\mu\nu} - \square h. \quad (11)$$

여기에서  $h \equiv h_\mu^\mu$ 는  $h_\mu^\nu$ 의 흔적(trace) 이고,  $\square \equiv \eta^{\mu\nu} \partial_\mu \partial_\nu = \partial_\mu \partial^\mu$ 는 평평한 시공간에서 달랑베르(d'Alembertian) 연산자이다.

이를 적용하면 선형화된 아인슈타인 텐서(Einstein tensor)  $G_{\mu\nu}$ 는 다음과 같은 형태를 가진다.

$$\begin{aligned} G_{\mu\nu} &= R_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}R. \\ &= \frac{1}{2}(\partial_\mu \partial_\rho h_\nu^\rho - \square h_{\mu\nu} - \partial_\mu \partial_\nu h + \partial_\nu \partial_\rho h_\mu^\rho) \\ &\quad - \frac{1}{2}\eta_{\mu\nu}(\partial_\rho \partial_\sigma h^{\rho\sigma} - \square h). \end{aligned} \quad (12)$$

아인슈타인 텐서(Einstein tensor)를 간단하게 하게 위해 역-흔적(trace-reversed) 거리잡기 섭동(metric perturbation)인  $\bar{h}_{\mu\nu}$ 를 도입한다.  $\bar{h}_{\mu\nu}$ 의 정의는 다음과 같다.

$$\bar{h}_{\mu\nu} \equiv h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h. \quad (13)$$

여기에서  $\eta = \eta_\mu^\mu = 4$ 이기 때문에  $\bar{h} = \eta^{\mu\nu}\bar{h}_{\mu\nu} = -h$  라 한다.

따라서 거리잡기 섭동(metric perturbation)  $h_{\mu\nu}$ 는 다음과 같이 쓸 수 있다.

$$h_{\mu\nu} = \bar{h}_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}\bar{h}. \quad (14)$$

아인슈타인 텐서(Einstein tensor)에서  $h_{\mu\nu}$ ,  $h$ 를 대신하여  $\bar{h}_{\mu\nu}$ ,  $\bar{h}$ 를 사용하여 역-흔적(trace-reversed) 거리잡기 섭동(metric perturbation)을 포

함하는 선형화된 아인슈타인 텐서(Einstein tensor)를 아래와 같이 나타낼 수 있다.

$$G_{\mu\nu} = \frac{1}{2}(\partial_\mu\partial_\rho\bar{h}_\nu^\rho + \partial_\rho\partial_\nu\bar{h}_\mu^\rho - \square\bar{h}_{\mu\nu} - \eta^{\mu\nu}\partial_\rho\partial_\sigma\bar{h}^{\rho\sigma}). \quad (15)$$

선형화된 아인슈타인 방정식은 다음과 같다.

$$\partial_\mu\partial_\rho\bar{h}_\nu^\rho + \partial_\rho\partial_\nu\bar{h}_\mu^\rho - \square\bar{h}_{\mu\nu} - \eta^{\mu\nu}\partial_\rho\partial_\sigma\bar{h}^{\rho\sigma} = \frac{16\pi G}{c^4}T_{\mu\nu}. \quad (16)$$

여기에서  $\partial_\nu\bar{h}^{\mu\nu} = 0$ 과 같은 로렌츠 게이지(Lorentz gauge) 상태를 사용하여 선형화된 아인슈타인 방정식을 간단히 하면 다음과 같이 나타낼 수 있다.

$$\square\bar{h}_{\mu\nu} = -\frac{16\pi G}{c^4}T_{\mu\nu}. \quad (17)$$

이론적으로 중력파의 발생은,  $\square\bar{h}_{\mu\nu} = -16\pi GT_{\mu\nu}$ 에서 시작된다.

여기에서 간단하게 계산하기 위해  $c = G = 1$ 로 두었다.

위 식의 해는  $\bar{h}_{\mu\nu}(x^\rho) = -16\pi \int d^4y G(x^\rho - y^\rho) T_{\mu\nu}(y^\rho)$ 로 주어진다.

이때  $G(x^\rho - y^\rho)$ 는  $\square_x G(x^\rho - y^\rho) = \delta^{(4)}(x^\rho - y^\rho)$ 를 만족하는 그린함수(Green's function)이며, 지연된 그린함수(retarded Green's function)의 해를 택한다.

$$G(x^\rho - y^\rho) = -\frac{1}{4\pi|x-y|}\delta[|x-y|-(x^0-y^0)]\theta(x^0-y^0). \quad (19)$$

여기에서 일반적으로  $|x-y| = [\delta_{ij}(x^i - y^i)(x^j - y^j)]^{1/2}$ 이고,

$x = (x^1, x^2, x^3)$ ,  $y = (y^1, y^2, y^3)$ 는 공간벡터이며, 계단함수(step function)인  $\theta(x^0 - y^0)$ 는  $x^0 > y^0$ 이면 1이고, 그렇지 않으면 0이다.

위의 식을 정리하면 중력파 발생의 해는 다음과 같다.



$$\bar{h}_{\mu\nu}(t, x^i) = 4 \int d^3y \frac{1}{|x-y|} T_{\mu\nu}(t-|x-y|, y^i). \quad (20)$$

이때  $t = x_0, t_r \equiv t - |x-y|$ 이고,  $t_r$ 은 지연시간으로 정의 된다.

아인슈타인의 방정식으로부터 예측 가능한 중력파를 발생시키기 위해서는 중력장의 변화가 있어야 한다. 실제로 천체들은 대부분 다른 천체에 대해 상대 운동을 하고 있기 때문에 중력파는 끊임없이 만들어진다고 볼 수 있다. 그러나 중력파 검출기가 검출할 수 있을 정도의 충분한 크기를 가진 파동은 강력한 중력장이 변할 때 만들어진다. 중력파 검출할 때 고려해야 할 또 다른 사항은 중력파 역시 거리가 멀어질수록 그 크기가 줄어든다는 점이다. 비교적 약한 중력파라도 지구로부터 가까운 곳에서 오는 것이면 검출이 가능하고 아주 먼 곳에서 오는 것은 더 어렵게 된다. 따라서 천체 물리학적으로 중요한 중력파원은 검출 가능한 거리 이내에 충분히 자주 중력파를 낼 수 있어야 한다. 중력파의 진폭은 거리에 반비례 한다. 강력한 중력파를 만들어내는 현상은 대부분 매우 격렬한 변화에 의한 것이다. 한 은하에서 이러한 격렬한 현상은 자주 일어나지 않지만 아주 먼 거리에서 오는 중력파까지 검출할 수 있다면 관측 빈도는 충분히 높을 수 있다. 반면 격렬하지는 않아도 꾸준히 연속적으로 중력파를 내는 현상도 있을 수 있다. 그러나 이러한 중력파는 대개 그 진폭이 작을 것으로 예상된다. 다만 이러한 천체가 가까운 거리에서 발견될 수 있다면 중력파원으로 흥미로운 대상이 될 수 있다.

편의상 중력파원은 크게 연속파원과 폭발파원으로 나눌 수 있다. 연속파원은 오랫동안 일정한 진폭과 진동수를 가지고 중력파를 내는 천체를 말하고 폭발파원은 갑자기 짧은 순간에 막대한 중력파를 내는 것을 말한다. 연속파원으로는 펄사와 초기 우주를 들 수 있고, 폭발 파원으로는 중성자 별 또는 블랙홀 쌍성의 합병이나 초신성, 그리고 감마선 폭발을 들 수 있다.

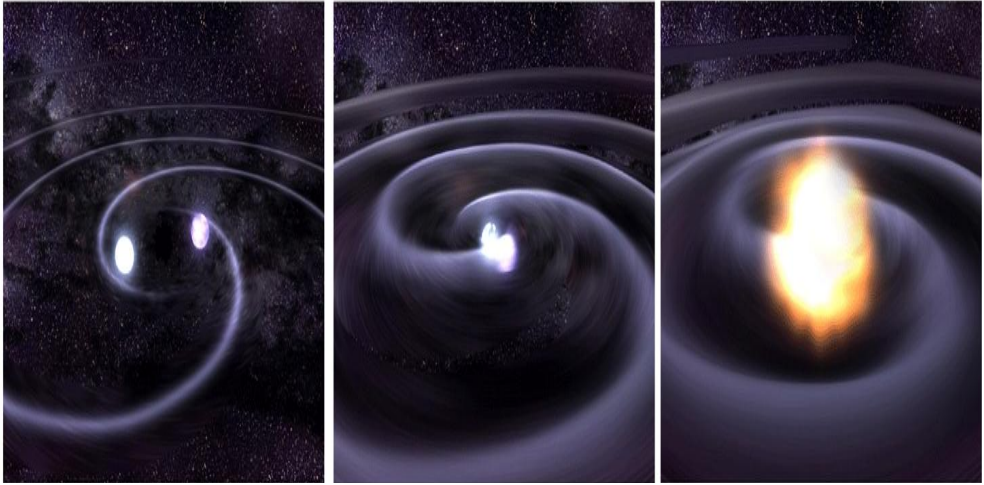


그림 3 : 두 별이 서로를 공전하고 병합하는 과정에서 중력파를 낸다.

NASA/CXC/GSDC/T.Strohmayer

중력파원 가운데 가장 쉽게 이해할 수 있는 것은 고밀도 천체 두 개로 이루어진 쌍성계이다. 이들은 질량 중심에 대한 궤도를 따라 움직이면서 중력장의 진동을 일으키고 결과적으로 중력파를 발생시킨다. 특히 쌍성의 궤도는 매우 간단하게 기술 할 수 있기 때문에 중력파의 모양을 나타내는 파형을 비교적 쉽게 계산할 수 있다. 중력파는 에너지와 각운동량을 가지고 있어 쌍성의 궤도는 중력파에 의해 점점 작아지고 원 궤도 모양으로 변한다. 쌍성이 내는 중력파의 강도는 궤도 반지름이 작아질수록 더 커진다. 따라서 궤도의 변화 역시 궤도 반지름이 작아질수록 더 급하게 일어나고 두 별은 궁극적으로 충돌하게 된다. 만약 중성자별 두 개가 충돌하면 두 별은 합쳐져 하나가 되며 그 질량은 중성자별 질량의 상한을 넘기 때문에 블랙홀로 변한다. 쌍성이 궤도 운동을 하는 과정을 나선회전(inspiral), 합쳐지는 것을 병합(merge), 그리고 두 별이 합쳐져 블랙홀로 만들어진 직후에도 중력장의 진동이 당분간 지속되지만 곧 약해지는 감쇠단기동의 중력파는 내려 이 과정을 안정화(ringdown)이라고 한다(그림 3). 쌍성계의 나선회전(inspiral)단계와 안정화(ringdown)단계에서는 이론적으

로 뉴턴역학과 포스트-뉴토니안(pN) 방법으로 근사적으로 기술될 수 있으며, 이를 통해 근사적인 파형의 공식을 유도할 수 있다. 병합(merge)단계는 구성별의 구체적인 구조와 물질의 상태방정식에 따라 달라지고 매우 복잡하기 때문에 수치 상대론적 계산을 통해서만 파형을 구할 수 있다.

## 2. 중력파형

우주에는 여러 종류의 중력파원이 존재한다. 이러한 중력파원 중에서 가장 먼저 발견될 가능성이 높은 것이 근접 쌍성계이다. 중성자별이나 블랙홀과 같이 아주 작은 천체들이 서로 가까이 접근할 때 현재 가동되고 있는 지상의 중력파 검출기로 측정가능한 양의 중력파를 낼 수 있기 때문이다.

두 별의 거리가  $R$ 만큼 떨어져 있고 지구에서  $r$ 만큼 떨어진 곳에서 기울임 각이  $\iota$ (iota) 이고, 반지름이  $R$ 인 원 궤도를 따라 회전하고 있는 쌍성계를 생각하자. 이때 두 별의 총 질량은  $M = m_1 + m_2$ 이고 환산질량은  $\mu = \frac{m_1 m_2}{M}$ 이다. 쌍성계의 질량분포는 아래와 같은 에너지 밀도로 주어진다.

$$\rho(\vec{x}) = m_1 \delta(x - x_1) \delta(y - y_1) \delta(z) + m_2 \delta(x - x_2) \delta(y - y_2) \delta(z). \quad (21)$$

여기에서 각 값의 해는 다음과 같다.

$$\begin{aligned} x_1 &= \frac{m_2}{M} R \cos \Omega t, y_1 = \frac{m_2}{M} R \sin \Omega t, \\ x_2 &= -\frac{m_1}{M} R \cos \Omega t, y_2 = -\frac{m_1}{M} R \sin \Omega t. \end{aligned} \quad (22)$$

위 식을 중력파의 성질인 사중극자 모멘텀 텐서(momentum tensor)의 정의인  $I_{\mu\nu}(t) \equiv \int y_\mu y_\nu T^{00}(t, \vec{y}) d^3y$  에 대입하면 다음과 같다.

$$\begin{aligned} \ddot{I}_{xx} &= -2\mu \Omega^2 R^2 \cos 2\Omega t, \\ \ddot{I}_{yy} &= 2\mu \Omega^2 R^2 \cos 2\Omega t, \\ \ddot{I}_{xy} &= \ddot{I}_{yx} = -2\mu \Omega^2 R^2 \sin 2\Omega t. \end{aligned} \quad (23)$$

지구에서 이 쌍성계를 관측하는 관찰자 기준 좌표계  $(r, \iota, \phi)$ 을 도입하면 이들에 대한 기본 벡터는

$$\begin{aligned}\hat{e} &= \cos\iota\cos\phi\hat{e}_x + \cos\iota\sin\phi\hat{e}_y - \sin\phi\hat{e}_z, \\ \hat{e}_\phi &= -\sin\phi\hat{e}_x + \cos\phi\hat{e}_y.\end{aligned}\quad (24)$$

가 되고 위의 식은 다음과 같이 주어진다.

$$\begin{aligned}\ddot{I}_\iota &= -2\mu\Omega^2R^2\cos^2\iota\cos 2(\Omega t - \phi), \\ \ddot{I}_{\phi\phi} &= 2\mu\Omega^2R^2\cos 2(\Omega t - \phi), \\ \ddot{I}_{\iota\phi} &= \ddot{I}_{\phi\iota} = -2\mu\Omega^2R^2\cos^2\iota\sin 2(\Omega t - \phi).\end{aligned}\quad (25)$$

따라서  $\bar{h}_{0i} = 0, \bar{h}_\mu^\mu = 0$ 인 TT 게이지 조건(gauge condition)을 이용하여 정리하면 사중극자모멘트의 세기 텐서는

$$\begin{aligned}h_+(t) &= -\frac{2G}{c^4r}\mu(GM\pi f_{GW})^{\frac{2}{3}}(1 + \cos^2\iota)\cos 2(\pi f_{GW}t - \phi), \\ h_\times(t) &= -\frac{2G}{c^4r}\mu(GM\pi f_{GW})^{\frac{2}{3}}\cos\iota\sin 2(\pi f_{GW}t - \phi).\end{aligned}\quad (26)$$

와 같이 된다. 여기서 중력과 진동수는 회전 진동수의 두 배로

$f_{GW} = \frac{\Omega}{\pi}$  주어진다. 만약  $\cos(\iota)$ 가 1이 된다면 검출기에서 관측된 중력파

의 세기는 최대가 되며  $h_0 \equiv \sqrt{h_+^2 + h_\times^2} = \frac{4G}{c^4r}\mu(GM\pi f_{GM})^{\frac{2}{3}}$ 가 된다.

한편, 중력파가 방출하는 에너지 손실은 궤도 에너지의 시간 변화량으로

주어지며 그 값은  $L = \dot{E} = -\frac{32\mu^2G^4M^3}{5c^5R^5}$  으로 주어지며, 두 별 사이의

시간에 따른 궤도 변화는 단순히 위 식에서  $R(t)$ 을  $t$ 에 관해  $t$ 에서부터  $t_c$ 까지 적분하여 다음과 같이 얻어진다.

$$R(t) = \left( \frac{256 G^3 \mu M^2}{5 c^5} \right)^{\frac{1}{4}} (t_c - t)^{\frac{1}{4}}. \quad (27)$$

여기에서  $t_c$ 는 병합이 이루어지는 시간을 의미한다.

중력파는 일반 상대론적 현상이지만 궤도 반지름이 슈바르츠 반지름 보다 훨씬 크다면 궤도 주기는 뉴턴역학의 결과인 케플러 법칙을 이용해 구할 수 있고 그 결과를 궤도 변화식에 적용하면 중력파 세기의 파형은 다음과 같다.

$$\begin{aligned} h_c &= A(t) \cos 2\Omega(t)t, \\ h_s &= A(t) \sin 2\Omega(t)t. \end{aligned} \quad (28)$$

진폭과 위상은 각각 다음과 같다.

$$\begin{aligned} A(t) &= \frac{4G\mu}{D} \Omega^2(t) R^2(t) = \frac{(5c^5 G)^{\frac{1}{4}}}{D} M^{\frac{5}{4}} (t_c - t)^{-\frac{1}{4}}, \\ \Phi(t) &= \int \Omega(t) dt = \Phi_0 - \left( \frac{c^5}{G^{5/3}} \right)^{\frac{3}{8}} \left( \frac{t_c - t}{5M} \right)^{\frac{5}{8}}. \end{aligned} \quad (29)$$

여기에서  $M$ 은 처프질량(chirp mass)로서  $M \equiv \left( \frac{\mu}{M} \right)^{\frac{3}{5}}$ 으로 정의 된다.

중력파에 의한 에너지 손실을 적용함으로써 궤도 반지름이 시간에 대해 변하는 양상을 구하고 파형에 적용하면 파형이 시간에 대해 어떻게 변하는 지 알 수 있다. 시간이 지나면서 궤도 반지름이 작아지기 때문에 궤도 진동수의 두 배에 해당하는 중력파 진동수가 점차 증가하고 진폭도 같이 증가하는 모습을 볼 수 있다(그림 4).

IV장에서는 여기서 구한 파형을 바탕으로 하여 포스트-뉴토니안(pN) 근사를 한 모형을 적용한 중력파형에 대해 알아 볼 것이다.

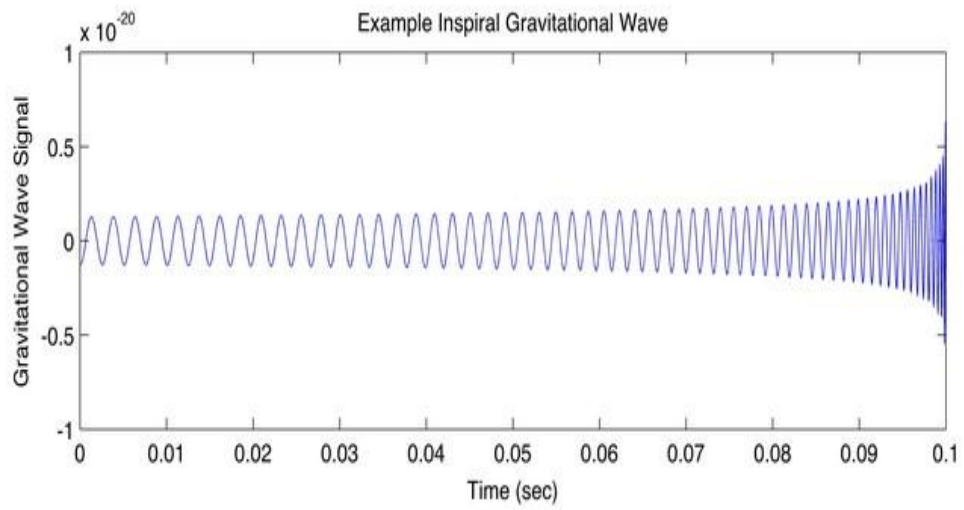


그림 4 : 나선회전 단계에서 중력파 소스의 예  
A. Stuver/LIGO

### 3. 중력파 검출

중력파 검출의 역사는 1960년부터 미국의 실험 물리학자인 조셉웨버(Joseph Weber)에 의해 독보적으로 시작되었다. 중력파의 사중편광이란 성질을 이용하여 막대 검출기(Bar Detector)를 고안하였다. 시공간이 사중편광으로 진동하면 실제 중력파에 의한 거대한 막대검출기의 원의 길이가 미세하게 변화될 것이다. 그때의 길이 변화를 측정하고자 했다. 그러나 막대 검출기의 민감도는  $10^{-16} \sim 10^{-17}$  정도로 최소 측정 가능한 민감도인  $10^{-21}$ 에 크게 미치지 못하였다. 그 후 1990대 캘리포니아 공대에 킵 쏜(Kip Thorne)교수가 마이켈슨 간섭계라고 부르는 형태의 레이저 간섭형 검출기를 제안했다(그림 5).

레이저 간섭형 검출기는 레이저를 빛 분배기로 보내면 두 갈래로 50%씩 갈라진 빛이 양쪽 팔로 갔다가 끝에 있는 거울에 반사되어 다시 돌아온다. 그때 만나는 점에서 간섭을 하게 된다. 중력파가 지나가지 않을 경우 양쪽의 팔 길이가 같기 때문에 돌아온 빛은 위상차 없이 진폭이 두 배가 되는 일정한 패턴을 가진 보강간섭을 일으키게 된다. 그러나 중력파가 지나가게 되면 때문에 중력파가 진행하는 방향에 따라 한 쪽 팔이 길어지고, 다른 한쪽이 짧아 질수 있는데, 만나는 점에서 빛은 다른 위상을 가지고 간섭무늬의 패턴을 형성하게 된다. 그러면 이 무늬를 이용하여 중력파가 지나갔는지 확인 할 수 있다. 이 레이저 간섭계를 이용하여 미국에서 LIGO(Laser Interferometer Gravitational-wave Observatory, 라이고)라 불리는 중력파 검출기가 건설되기 시작하였다. 미국의 루이지애나 주의 리빙스턴(Livingston)과 워싱턴주의 핸포드(Hanford) 두 곳에 각각 양쪽 팔 길이가 4km에 달하는 레이저 간섭계를 건설하였다.



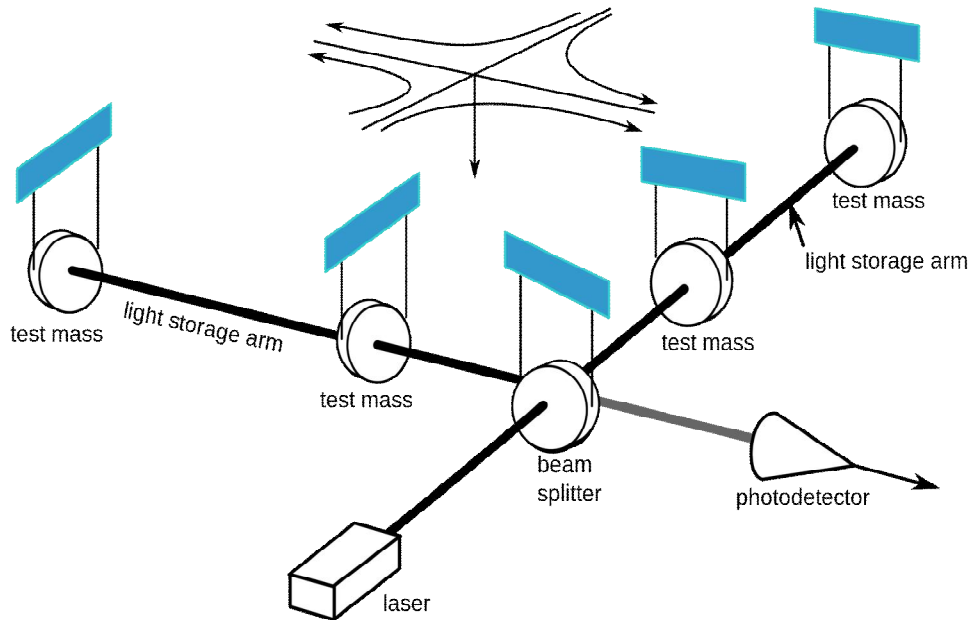


그림 5 : 레이저 간섭계의 개략도

[http://en.wikipedia.org/wiki/Gravitational-wave\\_observatory](http://en.wikipedia.org/wiki/Gravitational-wave_observatory)

라이고 검출기에는 현대에 개발된 많은 최신 기술들이 집약적으로 사용된다. 중력과 검출에는 고출력 레이저를 필요로 하지만, 레이저 출력을 높이면 거울의 온도가 올라가는 한계가 있다. 이를 해결하기 위하여 파브리-페로 공동(Fabry-Perot Cavity)라는 것을 이용한다. 이 파브리-페로 기술은 양쪽 팔에 설치하게 되는데 양팔 내부 내리저의 강도를 높이는 반사경을 하나 더 두어 반사된 빛이 계속 반사하게 만들어 양팔 내부 레이저의 강도를 높이는 기술이다. 반복횟수가 많을수록 검출기의 유효길이는 더 길어지게 만들 수 있다. 이 기술을 사용하는 이유는 검출기의 팔의 유효 길이가 길면 길수록 중력파의 신호를 검출하는데 유리하기 때문이다.

하지만 라이고 검출기는 지구상에서 가장 민감한 진동검출기로 물리적 길이가 길어지면 해당 환경진동에 노출이 심하기 때문에 물리적인 길이를

늘이는 것은 기술적인 한계가 있다. 실제 중력파를 검출하기 위해서는 이 진동 잡음이 원래 신호에 비해 덜 민감하도록 기기를 고안해야 하는데, 이를 위해 진동을 상쇄하는 장치를 별도로 달고 있다. 진자를 이용하면 높은 진동수에 진동에는 취약하지만 낮은 진동수에 덜 민감한 검출기를 만들 수 있다.

검출기에 의해 측정된 중력파의 변형(gravitational strain)은 다음과 같이 주어진다.

$$h_{strain}(f) = F_+ h_+(t) + F_\times h_\times(t). \quad (31)$$

여기에서  $F_+, F_\times$  는 검출기의 민감도에 두 개의 다른 편광에 의해 나타나는 반응함수(response function)이다.

라이고(LIGO)와 버고(Virgo)와 같이 팔을 가지는 레이저 간섭계의 오른쪽 각도, 구좌표계의 하늘의 위치( $\bar{\theta}, \bar{\phi}$ ), 편광각( $\bar{\psi}$ )에서 오는 중력파의 안테나 반응함수는 다음과 같다.

$$\begin{aligned} F_+ &= \frac{1}{2}(1 + \cos^2 \bar{\theta}) \cos 2\bar{\phi} \cos 2\bar{\psi} - \cos \bar{\theta} \sin 2\bar{\phi} \sin 2\bar{\psi}, \\ F_\times &= \frac{1}{2}(1 + \cos^2 \bar{\theta}) \cos 2\bar{\phi} \sin 2\bar{\psi} - \cos \bar{\theta} \sin 2\bar{\phi} \cos 2\bar{\psi}. \end{aligned} \quad (32)$$

## IV. 밀집쌍성계의 중력과 모델

### 1. post-Newtonian 방법론

포스트-뉴토니안(post-Newtonian) 방법론은 느린-운동( $\frac{v}{c} \ll 1$ )과 약한-장( $\frac{GM}{rc^2} \ll 1$ ) 시스템에서 방출되는 중력파형을 계산하는 체계를 제공한다. 이러한 양 중 하나 또는 둘 모두는 작은 매개변수를 처리하고, 아인슈타인의 방정식을 반복적으로 해결된다. 포스트-뉴토니안(pN) 근사는 매개변수( $\gamma \equiv \frac{GM}{ac^2}$ )와 뉴턴의 중력이론에 대한 수정의 전개를 포함한다. 이 방법론에서, 뉴턴 방정식은 포스트-뉴토니안(pN) 전개식의 첫 번째 항에 대응한다. 도식적 운동 방정식은 다음과 같이 주어진다.

$$\frac{d^2 x^i}{dt^2} = -\frac{GMx^i}{a^3} \left[ 1 + O\left(\gamma^{\frac{1}{2}}\right) + O(\gamma^1) + O\left(\gamma^{\frac{3}{2}}\right) + \dots \right]. \quad (34)$$

여기서,  $x^i$ 는 두 물체와  $a = |x|$ 사이의 거리 벡터이다.

케플러의 제3법칙의 일반화된 버전은 (35)에서 3PN까지 유도되었다.

$$\omega^2 = \frac{M}{a^3} \left\{ 1 + (-3 + \nu)\gamma + \left( 6 + \frac{41}{4}\nu + \nu^2 \right) \gamma^2 + \left( -10 + \left[ -\frac{75707}{840} + \frac{41}{64}\pi^2 + 22\ln\left(\frac{a}{a_0}\right) \right] \nu \right) \gamma^3 + \left( +\frac{19}{2}\nu^2 + \nu^3 \right) \gamma^4 \right\}. \quad (35)$$

여기서  $\nu$ 는 대칭 질량 비율(symmetric mass ratio)이고,  $a_0$ 은 상수이다. 원 궤도를 통해, 에너지 또한 다음과 같이 전개될 수 있다.

$$E = -\frac{\mu\gamma}{2} \left\{ 1 + \left( -\frac{7}{4} + \frac{1}{4}\nu \right) \gamma + \left( -\frac{7}{8} + \frac{49}{8}\nu + \frac{1}{8}\nu^2 \right) \gamma^2 \right. \\ \left. + \left( -\frac{235}{64} + \left[ -\frac{46031}{2240} - \frac{123}{64}\pi^2 \right] \nu + \frac{22}{3} \ln\left(\frac{a'}{a^0}\right) \right) \gamma^3 \right. \\ \left. + \frac{27}{32}\nu^2 + \frac{5}{64}\nu^3 \right\} \quad (36)$$

pN파라미터  $\gamma$ 는 분리점  $a$ 에 의해 정의된 바와 같이 이들 식은 좌표계의 선택에 의존한다. 프레임-불변 표현을 발견하기 위해, 우리는

$x \equiv \left( \frac{GM\omega}{c^3} \right)^{\frac{2}{3}}$ 로 정의된 주파수 관련 파라미터  $x$ 를 사용할 수 있으며, 우

리는 다음의 3PN차수에서  $x$ 항에서의  $\gamma$ 표현을 가진다.

$$\gamma = x \left\{ 1 + \left( 1 - \frac{1}{3}\nu \right) x + \left( 1 - \frac{65}{12}\nu \right) x^2 \right. \\ \left. + \left( 1 - \left[ -\frac{2203}{2520} - \frac{41}{192}\pi^2 + \frac{22}{3} \ln\left(\frac{a'}{a^0}\right) \right] \nu \right) x^3 \right. \\ \left. + \left( -\frac{229}{36}\nu^2 + \frac{1}{81}\nu^3 \right) \right\} \quad (37)$$

식 (37)를  $\gamma$ 관련 PN방정식에 대입하면 우리는 독립적인 좌표의 PN방정식을 가진다. 관련된 게이지(gauge)상수  $a'^0$ 와 로그가 함께 상쇄될 수 있다. 에너지는 다음과 같다.

$$E = -\frac{\mu x}{2} \left\{ 1 + \left( -\frac{3}{4} - \frac{1}{12}\nu \right) x + \right. \\ \left( -\frac{27}{8} + \frac{19}{8}\nu - \frac{1}{24}\nu^2 \right) x^2 \\ \left. + \left[ -\frac{675}{64} + \left( \frac{34445}{576} - \frac{205}{96}\pi^2 \right) \nu \right] x^3 \right. \\ \left. + \left[ -\frac{155}{96}\nu^2 - \frac{35}{5184}\nu^3 \right] \right\} \quad (38)$$

원형궤도의  $x^{\frac{7}{2}}$  차수의 항이다. 그래서 이 결과는 3.5PN차수까지 유효하다.  
중력다발은 (39)에 의해 3.5PN로 유도될 수 있다.

$$\mathcal{L} = -\frac{32\nu^2 x^5}{5} \left\{ 1 + \left( -\frac{1247}{336} - \frac{35}{12}\nu \right) x + 4\pi x^{\frac{3}{2}} \right. \\ \left( -\frac{44711}{9072} + \frac{9271}{504}\nu + \frac{65}{18}\nu^2 \right) x^2 \\ + \left( -\frac{8191}{672} + \frac{583}{24}\nu \right) \pi x^{\frac{5}{2}} \\ + \left[ \frac{6643739519}{69854400} + \frac{16}{3}\pi^2 \right. \\ \left. - \frac{1712}{105}\gamma - \frac{856}{105}\ln(16x) \right. \\ \left. + \left( -\frac{134543}{7776} + \frac{41}{48}\pi^2 \right) \nu \right. \\ \left. - \frac{94403}{3024}\nu^2 - \frac{775}{324}\nu^3 \right] x^3 \\ \left. \left( -\frac{16285}{504} + \frac{214745}{1728}\nu + \frac{193385}{3024}\nu^2 \right) \pi x^{\frac{7}{2}} \right\} \quad (39)$$

포스트-뉴토니안(pN) 근사에서, 중력파 변형 TT는 TT를 가지는 사중극자  
식을 넘어 전개항으로 다음과 같이 표현될 수 있다.

$$h^{TT}_{ij} = \frac{2}{r} \left\{ \ddot{I}_{ij} \left[ 1 + O(x^{\frac{1}{2}}) + O(x) + O(x^{\frac{3}{2}}) + \dots \right] \right\}_{TT}. \quad (40)$$

## 2. TaylorF2 모델

근사의 가장 공통적으로 사용되는 형태는 SPA(Stationary phase approximation)사용하여 계산된 푸리에 표현이다. 주파수의 함수인 SPA(Stationary phase approximation)파형은 다음과 같이 쓰인다.

$$\tilde{h}^{spa}(f) = \frac{a(t_f)}{\sqrt{\dot{F}(t_f)}} e^{i[\psi_f(t_f) - \frac{\pi}{4}]}, \quad \psi_f(t) \equiv 2\pi ft - 2\phi(t). \quad (41)$$

여기서  $t_f$ 는  $t$ 의 해결에 의해 정의된 안장점(saddle point)이다,  $\frac{d\psi_f(t)}{dt} = 0$ . 즉, 중력파 주파수  $F(t)$ 의 푸리에 변수  $f$ 가 같아 질 때 시간을  $t_f$ 라 한다.

단열근사에서  $t_f$ 와  $\psi_f(t_f)$ 의 값을 ( $v_f \equiv (\pi M f)^{\frac{1}{3}}$ 를 나타내는 것은) 다음의 적분 값에 의해 주어진다.

$$t_f = t_{ref} + M \int_{v_f}^{v_{ref}} \frac{E'(v)}{F(v)} dv, \quad (42)$$

$$\psi_f(t_f) = 2\pi f t_{ref} - \phi_{ref} + 2 \int_{v_f}^{v_{ref}} (v^3 - v^3) \frac{E'(v)}{F(v)} dv.$$

시간 영역의 경우와 마찬가지로 그 등가 차동 형태인 (43)을 사용하는 것이 더 효율적이며, 이는 TaylorF2 근사의 특징이다.

$$\frac{d\psi}{df} - 2\pi t = 0, \quad \frac{dt}{df} + \frac{\pi M^2}{3v^2} \frac{E'(f)}{F(f)} = 0. \quad (43)$$

주파수 도메인에서 TaylorF2의 아날로그는 명확하게 줄이는 에너지와 다발의 함수를 이루고 있는 포스트-뉴토니안(pN) 차수와 위에서  $v$ 적분을 설명을 따른다.

푸리에 도메인파형인 TayloF2, 이것은 포스트-뉴토니안(pN)근사를 따라

다음과 같이 주어진다.

$$\tilde{h}(f) = Af^{-\frac{7}{6}}e^{i\psi(f)} = \frac{M^{\frac{5}{6}}Q}{D}f^{-\frac{7}{6}}e^{i\psi(f)}. \quad (44)$$

여기에서  $A \propto \frac{M^{\frac{5}{6}}Q(\angle)}{D}$ , 그리고  $D$ 는 연성사이의 거리이다.

SPA(Stationary phase approximation)는 주파수영역에서 나선회전 단계의 파형을 나타내는데 사용할 수 있다. TaylorF2는 TaylorT2와 유사하지만, SPA(Stationary phase approximation)이다.

TaylorF2의 파형은 다음과 같다.

$$\begin{aligned} \tilde{h}_+(f) &\approx -\frac{1+\cos^2\iota}{2}\left(\frac{5\pi}{24}\right)^{\frac{1}{2}}\eta^{\frac{1}{2}}\frac{G^2M^2}{c^5r}x^{-\frac{7}{4}}e^{-2\pi ift_c}e^{2\pi i\psi_c}e^{-i\Psi(f)}, \\ \tilde{h}_\times(f) &\approx i\cos\iota\left(\frac{5\pi}{24}\right)^{\frac{1}{2}}\eta^{\frac{1}{2}}\frac{G^2M^2}{c^5r}x^{-\frac{7}{4}}e^{-2\pi ift_c}e^{2\pi i\psi_c}e^{-i\Psi(f)}. \end{aligned} \quad (45)$$

여기에서  $\Psi(f)$ 는 다음과 같이 주어진다.

$$\Psi = -\frac{\pi}{4} + \frac{3}{128}\frac{1}{\eta}x^{-\frac{5}{2}}\left\{ \begin{aligned} &1 + \left(\frac{3715}{756} + \frac{55}{9}\eta\right)x - 16\pi x^{\frac{3}{2}} \\ &+ \left(\frac{15293365}{508032} + \frac{27145}{504}\eta + \frac{3085}{72}\eta^2\right)x^2 \\ &+ \left(\frac{38645}{756} - \frac{65}{9}\eta\right)\left[1 + \frac{3}{2}\ln\left(\frac{x}{x_0}\right)\right]\pi x^{\frac{5}{2}} \\ &\left[ \frac{11583231236531}{4694215680} - \frac{640}{3}\pi^2 \right. \\ &\quad \left. - \frac{6848}{21}\gamma_E - \frac{3424}{21}\ln(16x) \right. \\ &\quad \left. + \left(-\frac{15737765635}{3048192} + \frac{2255}{12}\pi^2\right)\eta \right. \\ &\quad \left. + \frac{76055}{1728}\eta^2 - \frac{127825}{1296}\eta^3 \right] x^3 \\ &+ \left(\frac{77096675}{254016} + \frac{378515}{1512}\eta - \frac{74045}{756}\eta^2\right)\pi x^{\frac{7}{2}} \end{aligned} \right\}. \quad (46)$$

그리고  $x = \left( \pi GM \frac{f}{c^3} \right)^{\frac{2}{3}}$  이다.



### 3. 진폭항이 추가된 TaylorF2 모델

포스트-뉴토니안(pN)전개 매개 변수는  $V = (2\pi MF)^{\frac{1}{3}}, F = \frac{\omega_{orb}}{2\pi}$ 로 정의된다. 여기서 중력파의 주파수는  $f$ 로 표시할 것이다.  $k^{th}$ 조화(harmonic)을 위해 우리는  $f = f_k = kF$ 의 관계를 가지고 있다. 즉 다음과 같이 나타낼 수 있다.

$$V_k = \left(2\pi M \frac{f_k}{k}\right)^{\frac{1}{3}}. \quad (47)$$

주어진 함수  $h(t) = A(t)\cos\phi(t)$  인데,  $\phi(t)$ 는  $\frac{d\ln A(t)}{dt} \ll \frac{d\phi(t)}{dt}$ 를 만족하는 감소하지 않고 증가하는 함수이다. 이것을 SPA(Stationary phase approximation)를 적용하여 푸리에 변환을 하면 다음과 같다.

$$\tilde{h}(f) \simeq \frac{1}{2}A(t(f))\sqrt{\frac{2\pi}{\ddot{\phi}(t(f))}}e^{i\left(2\pi ft(f) - \phi(t(f)) - \frac{\pi}{4}\right)}. \quad (48)$$

$t_f$ 가 있는  $\frac{d\phi}{dt}t = 2\pi f$ 에 대한  $t$ 의 값으로 각주파수  $f$ 에 대해 정의된다.

$$\Psi_{SPA}(f) = 2\pi ft(f) - \Psi(f). \quad (49)$$

(49)로서 주파수 별 SPA(Stationary phase approximation)단계 정의, 2.5차 까지의 포스트-뉴토니언(pN) 진폭항 보정과 주파수 도메인 파형은 다음과 같다.

$$\begin{aligned}
\tilde{h}(f) &= \frac{M\nu}{D_L} \sum_{n=0}^5 \sum_{k=1}^7 V_k^{2+n} \left( k \frac{dF}{dt} \right)^{-\frac{1}{2}} \\
&\quad \times \left( \alpha_k^{(n)} e^{i \left( 2\pi f t(F) - k\Psi(F) - \frac{\pi}{4} \right)} + \beta_k^{(n)} e^{i \left( 2\pi f t(F) - \left( k\Psi(F) - \frac{\pi}{2} \right) - \frac{\pi}{4} \right)} \right), \\
&= \frac{M\nu}{D_L} \sum_{n=0}^5 \sum_{k=1}^7 V_k^{n-\frac{7}{2}} \sqrt{\frac{5\pi}{\kappa 48\nu}} M(1 + S_2 V_k^2 + S_3 V_k^3 + S_4 V_k^4 + S_5 V_k^5) \\
&\quad \times \left( \alpha_k^{(n)} + e^{i\frac{\pi}{2}} \beta_k^{(n)} \right) e^{i \left( k\Psi_{SPA}\left(\frac{f}{k}\right) - \frac{\pi}{4} \right)}. \tag{50}
\end{aligned}$$

$$\tilde{h}(f) = \frac{M^2}{D_L} \sqrt{\frac{5\pi\nu}{48}} \sum_{n=0}^5 \sum_{k=1}^7 V_k^{n-\frac{7}{2}} C_k^{(n)} e^{i \left( k\Psi_{SPA}\left(\frac{f}{k}\right) - \frac{\pi}{4} \right)}. \tag{51}$$

여기서 인덱스  $n$ 은 PN차수 및 인덱스  $k$ 의 조화를 나타낸다.

SPA위상  $\Psi_{SPA}(F)$ 의 PN전개는 에너지 균형 방정식 (52)를 통해 연성 무게 중심 에너지  $E$ 와 중력파 다발  $F$ 의 PN 확장으로부터 얻을 수 있다.

$$-\frac{dE}{dt} = F. \tag{52}$$

$\frac{d\Psi}{dt} = 2\pi F = \frac{v^3}{M}$ 을 사용하면, 우리는 다음의 미분방정식을 에너지 균형 방정식으로 다시 작성할 수 있다.

$$dt = -\frac{dE}{dv} \frac{1}{F} dv, \quad d\Psi = -\frac{dE}{dv} \frac{1}{F} \frac{v^3}{M} dv. \tag{53}$$

$E, F$ 의 양은  $v = (2\pi M F)^{\frac{1}{3}}$  역급수로 알려져 있다.

무게중심에너지와 다발은 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
E(v) &= E_{newt} v^2 \left( 1 + \sum_{i=2}^6 E_i v^i \right), \\
F(v) &= F_{newt} v^{10} \left( 1 + \sum_{i=2}^7 F_i v^i \right). \tag{54}
\end{aligned}$$

우리는 2.5pN차수를 통래 SPA단계의 다음 식을 얻을 수 있다.

$$\Psi_{SPA}(F) = 2\pi Ft_c - \Psi_c + \frac{3}{256}(2\pi MF)^{-\frac{5}{3}} \times \left\{ 1 + \left( \frac{3715}{756} + \frac{55}{9}\nu \right) (2\pi MF)^{\frac{2}{3}} + (4\beta - 16\pi)(2\pi MF)^{\frac{2}{3}} \right. \\ \left. + \left( \frac{15293354}{508032} + \frac{27145}{504}\nu + \frac{3085}{72}\nu^2 - 10\sigma \right) (2\pi MF)^{\frac{4}{3}} \right. \\ \left. + \left( \frac{38645}{756}\pi - \frac{65}{9}\pi\nu - \gamma \right) (1 + 3\log(v))(2\pi MF)^{\frac{5}{3}} \right\}. \quad (55)$$

본 논문에서 식(51),(55)를 사용하여 기존의 LSC에서 사용하는 라이브러리인 LAL에서 LALSimInspiralTaylorF2.c 파일을 수정하여 진폭항이 2.5PN 차수까지 추가된 TaylorF2모델을 새로운 소스파일인 LALSimInspiralTaylorF2Amp.c 파일을 만들어 TaylorF2Amp라고 부르는 모델을 만들었다. 이 코드를 사용하여 본 연구에서 사용하는 계산을 수행하였다.

## V. 마르코프 체인 몬테카를로(Markov Chain Monte Carlo) 모수추정

### 1. 베이지언(Bayesian) 추론

베이즈 정리는 어떤 사건의 관측 전 원인에 대한 가능성과 관측 후 원인에 대한 가능성 사이의 관계를 설명하는 이론으로, 베이지언 추론은 확률이 빈도나 비례와 같은 것이 아닌 믿음의 정도로서 해석되는 통계적 추론이다. A가 발생할 확률을  $P(A)$ , B가 발생할 확률을  $P(B)$ , A와 B가 동시에 발생할 확률  $P(A, B)$ . 조건부 확률인  $P(A|B)$ 는 B가 발생할 때 A가 발생할 확률은 다음과 같이 정의된다.

$$P(A|B) = \frac{P(A, B)}{P(B)}, P(B|A) = \frac{P(A, B)}{P(A)}. \quad (56)$$

두 식으로부터 베이즈 정리(Bayes's Theorem)를 나타낼 수 있다.

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)}. \quad (57)$$

베이즈 정리는 사후확률(posterior probability), 사전확률(prior probability)로 나타낼 수 있다.  $P(B|A)$ 는 A가 발생할 때 B에 대한 조건부 확률이다. 즉 A 조건 하에 B의 사후 확률 이고,  $P(B)$ 는 B의 사후확률이다. 베이즈 정리는 다음과 같이 편리한 형태로 표현 할 수 있다.

$$P(A) = P(A|B)P(B) + P(A|\neg B)P(\neg B). \quad (58)$$

여기서  $P(A|\neg B)$ 는 A가 주어질 때 B가 발생되지 않을 확률이다.

즉,  $P(\neg B) = 1 - P(B)$ 는 B가 발생되지 않을 확률이다.

이로부터 (57)은 다음과 같이 쓸 수 있다.

$$\begin{aligned}
 P(B|A) &= \frac{P(B)P(A|B)}{P(A|B)P(B) + P(A|\neg B)P(\neg B)}, \\
 &= \frac{\Lambda(B|A)}{\Lambda(B|A) + \frac{P(\neg B)}{P(B)}}.
 \end{aligned}
 \tag{ 59 }$$

으로 나타낼 수 있다. 여기서  $\Lambda(B|A) \equiv \frac{P(A|B)}{P(A|\neg B)}$ 는 라이클리후드 비  
 (likelihood ratio)이다.

## 2. MCMC 모수추정

중력파원의 물리량을 찾는 방법에는 모수추정을 사용한다.

현재까지 검출된 중력파신호가 없기 때문에, 중력파 커뮤니티에서는 중력파형 모델을 사용하여 만들어낸 인공 중력파 신호를 가지고 모수추정 파이프라인을 테스트하고 있다. 중력파 후보들을 겹쳐 가면서 N차원 파라미터 공간에서 중력파 후보와 가장 잘 일치하는 해당 파라미터를 구하게 된다. 이때 사용하는 통계이론은 앞에서 보인 베이지 이론이다.

모수추정방법에 마르코프 체인 몬테카를로(MCMC) 샘플링 알고리즘을 사용한다.

마르코프 체인 몬테카를로(MCMC)방법은 변하지 않는 확률분포를 갖는 임의의 상태공간에서 무작위행보를 통하여 얻어진 마르코프 체인을 사용하여 주어진 매개변수의 결정과 오차를 예측할 수 있다. 이 논문에 사용한 데이터는 SNR(signal to noise ratio)이 20 이내의 값들을 사용하는데, 이때 SNR계산은 신호 템플릿과 데이터를 이용하여 다음 식을 통하여 계산한다.

$$\rho = 4R \left( \int_0^{+\infty} \frac{\tilde{h}(f) \hat{h}^*(f)}{s_n(f)} df \right). \quad (60)$$

수치적인 추론은 종종 MCMC라 불리는 기술을 사용한다. 대부분의 복잡한 시뮬레이션에서 베이지 정리 이벤트를 사용하여 사후분포를 수치적으로 사상(mapping)하는데 사용한다. 여기서 라이클리후드는 오직 수치적인 시뮬레이션에 의해 포함되고, 파라미터 공간은 수 십 개의 차원을 가지고, 포스테리어는 여러 개의 봉우리(peak)와 복잡한 구조를 가지고 있다.

이 라이클리후드 함수는 다음과 같이 계산된다.

$$\mathcal{L}(s|\lambda) \propto e^{\left[ -2 \int_0^{\infty} \frac{|\tilde{s}(f) - \tilde{h}(\lambda, f)|^2}{s_n(f)} df \right]}. \quad (61)$$

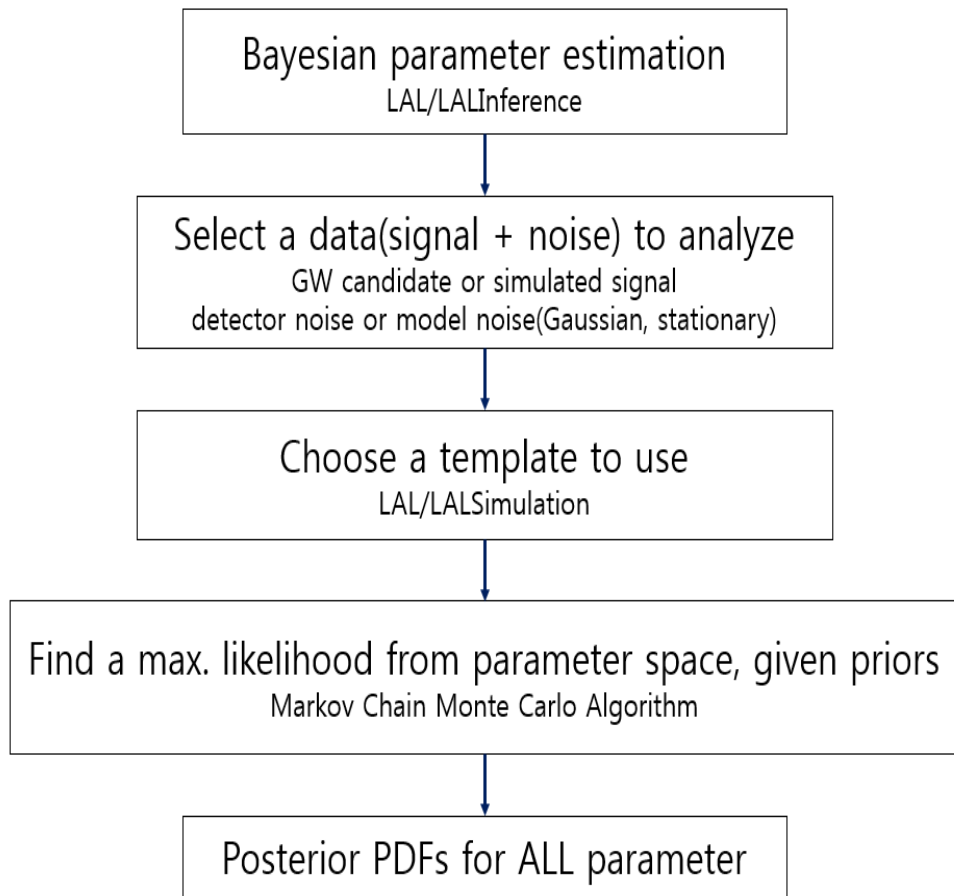


그림 6 : LAL에서 모수추정이 사용되는 과정을 도식화해 보여준다.

MCMC 알고리즘의 목적은 파라미터공간의 밀도에 비례하는 포스테리어의 포인트 시퀀스를 생성하는 것이다.

데이터로부터 통계학적으로 만족할 만한 결과를 구하기 위해, MCMC의 메트로폴리스-해스팅(Metropolis-hasting) 알고리즘을 사용한다.

LSC에서 사용하는 라이브러리인 LAL에서 모수추정방법을 적용하는 것을 보여주고 있다(그림 6).

## VI. 연구결과

현재까지는 직접적으로 관측된 데이터가 없기 때문에 시뮬레이션 하여 얻은 중력과 신호에 검출기의 노이즈를 넣어 만든 소프트웨어 인젝션을 사용하여 모수추정을 수행하였다. 이때 검출기의 노이즈는 Initial LIGO-Virgo 네트워크를 사용하여 시작 주파수는 30Hz로 고정되어있다. 쌍성계의 질량에 따른 진폭항의 변화를 확인하기 위해 질량비가 다른 블랙홀-중성자별 쌍성계와, 블랙홀-블랙홀 쌍성계를 이용하여 결과를 도출하였다. 본 논문에서는 스핀을 고려하지 않은 인젝션을 사용하였기에 9개의 파라미터를 사용하였고, 질량과 거리를 조절하여 SNR(signal to noise ratio)를 20에 고정하기 위해 나머지 파라미터들은 고정하였다(표 1).

binary	chirp mass ( $M_{sun}$ )	symmetric mass ratio	distance ( $M_{pc}$ )	inclination (rad)	polarization (rad)	orbital phase (rad)	coalescence time (s)	right ascension (rad)	declination (rad)
BH-NS	2.9943	0.1077	23.1	0.785	2.606	3.31	894383679	0.645	0.575
BH-BH	8.705	0.245	45.0						

표 1 : 주입한 물리량



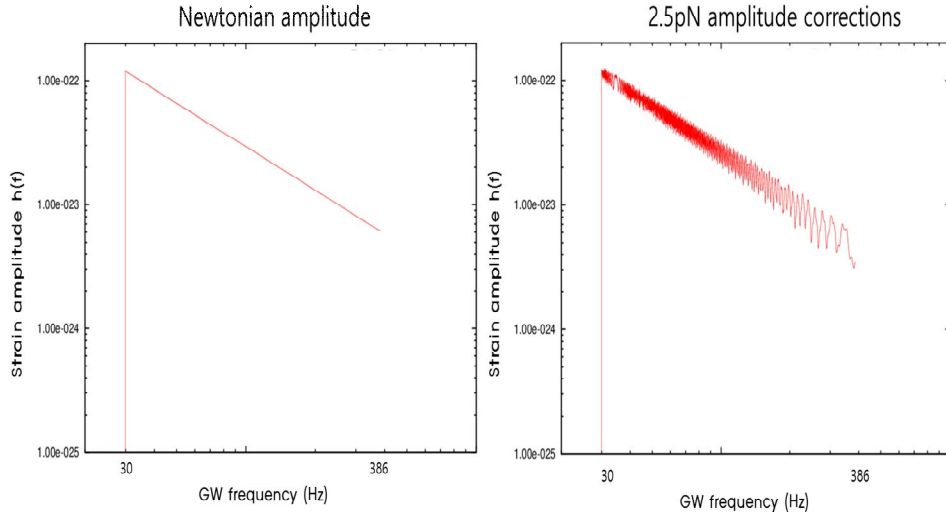


그림 7 : 10Msun - 1.4Msun 쌍성계의 진폭항에 따른 TaylorF2 파형

주입한 물리량을 사용하여 중력파형의 모델인 TaylorF2와 TaylorF2에 진폭항을 추가한 TaylorF2Amp의 파형을 비교하였다.

블랙홀-중성자별의 쌍성계를 고려한 경우의 중력파형(그림 7).

이때 사용한 별들의 질량은 10 Msun-1.4Msun으로 질량비는 0.14이다.

각 그래프의 x축은 중력파의 주파수이고 y축은 strain amplitude  $h(f)$ 를 나타 낸 것이다. 왼쪽 그림은 위상항이 3.5pN, 진폭항이 뉴토니안인 TaylorF2모델을 이용하여 그린 파형이고, 오른쪽그림은 위상항이 3.5pN, 진폭항이 2.5pN인 TaylorF2Amp를 이용하여 그린 파형이다.

시작 주파수는 30Hz부터이고 끝 주파수는 386Hz이다.

진폭항이 뉴토니안인 파형과 진폭항에 2.5pN까지 추가한 파형을 비교하여 보았을 때 진폭항이 추가된 파형(오른쪽)에서는 진폭항의 효과로 전체적인 주파수 대역에서 y축의 변화가 진동하는 듯한 효과를 나타나고 있는 것을

확인하였다. 또한  $h(f)$ 의 기울기는  $\tilde{h}(f) = Af^{-\frac{7}{6}}e^{i\psi(f)}$  식의  $f^{-\frac{7}{6}}$  와 같은 것을 확인하였다.

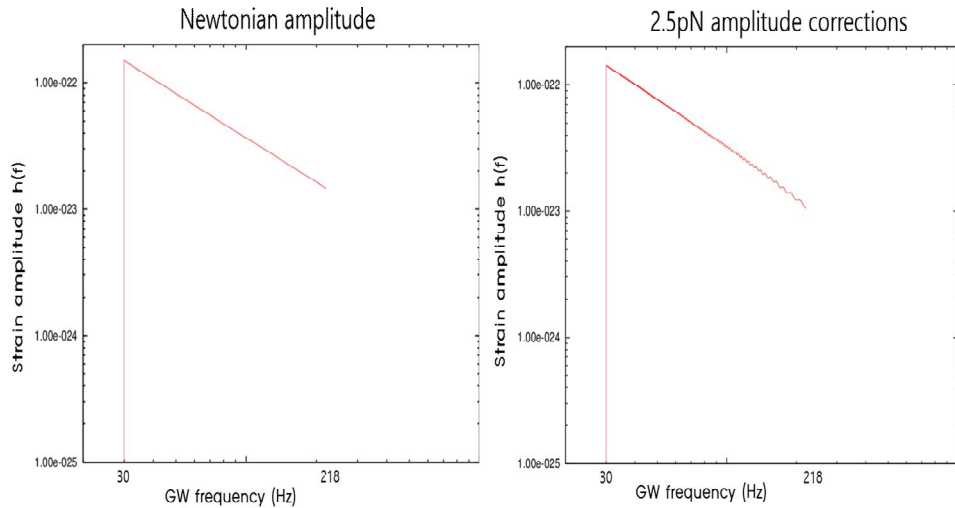


그림 8 : 10.1Msun - 9.9Msun 쌍성계의 진폭항에 따른 TaylorF2 파형

블랙홀-블랙홀 쌍성계의 파형을 비교하였다(그림 8).

(그림 8)에서 사용한 별들의 질량은 10.1 Msun-9.9Msun으로 질량비는 0.98이다. 각 그래프의 x축은 중력파의 주파수이고 y축은 strain amplitude  $h(f)$ 를 나타 낸 것이다. 왼쪽 그림은 위상항이 3.5pN, 진폭항이 뉴토니안인 TaylorF2모델을 이용하여 그린 파형이고, 오른쪽그림은 위상항이 3.5pN, 진폭항이 2.5pN인 TaylorF2Amp를 이용하여 그린 파형이다. 블랙홀-블랙홀의 경우 질량이 크기 때문에 상대적으로 더 빨리 돌고, 빨리 병합과정으로 도달 하게 되어, 끝 주파수는 218Hz로 블랙홀 중성자별 보다 짧다. 두 파형을 비교하였을 때 블랙홀-중성자별의 파형과는 달리 진폭항의 효과가 끝에서 미미하게 나타내고 있다.

이것은 10.1 Msun-9.9Msun을 가지는 블랙홀-블랙홀일 경우 질량비는 0.98로 거의 1에 가깝기 때문이다. 각각의 질량비가 다른 이벤트에 따른 파형을 비교하였을 때, 질량비가 1에 가까울 경우에는 진폭항의 효과가 제대로 나타나지 않는 것을 확인하였다.

모수추정한 결과를 이용하여 각 파라미터들의 사후확률(posterior probability)을 구할 수 있다. 각 그래프에서 점선은 소프트웨어주입한 파형의 물리량이다. 확률밀도함수의 분포가 정확한 가우시안은 아니지만, 인젝션 값이 확률밀도함수의 가장 높은 값에 일치하는 것을 확인할 수 있다. 밀도가  $10M_{\text{sun}}-1.4M_{\text{sun}}$ 의 질량을 가지는 블랙홀-중성자별의 파라미터들에 대한 확률밀도함수이다. 이때 3.5pN 위상향과, 뉴토니안 진폭향을 포함하고 있는 TaylorF2모델의 사용하였다(그림 9).

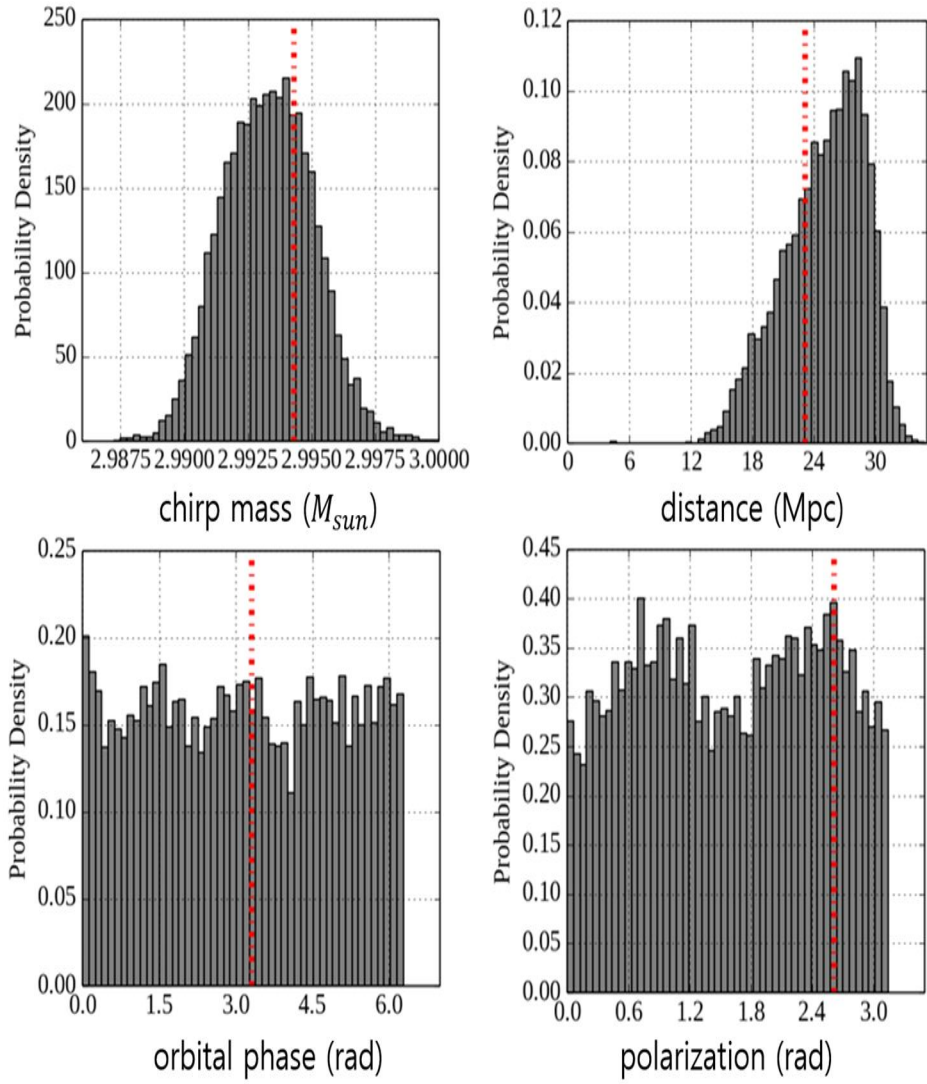


그림 9 : 10Msun-1.4Msun 질량을 가지는 블랙홀 중성자의 확률밀도분포 함수. 위상항이 3.5pN을 포함하고 있고, 진폭항이 0pN인 TaylorF2함수.

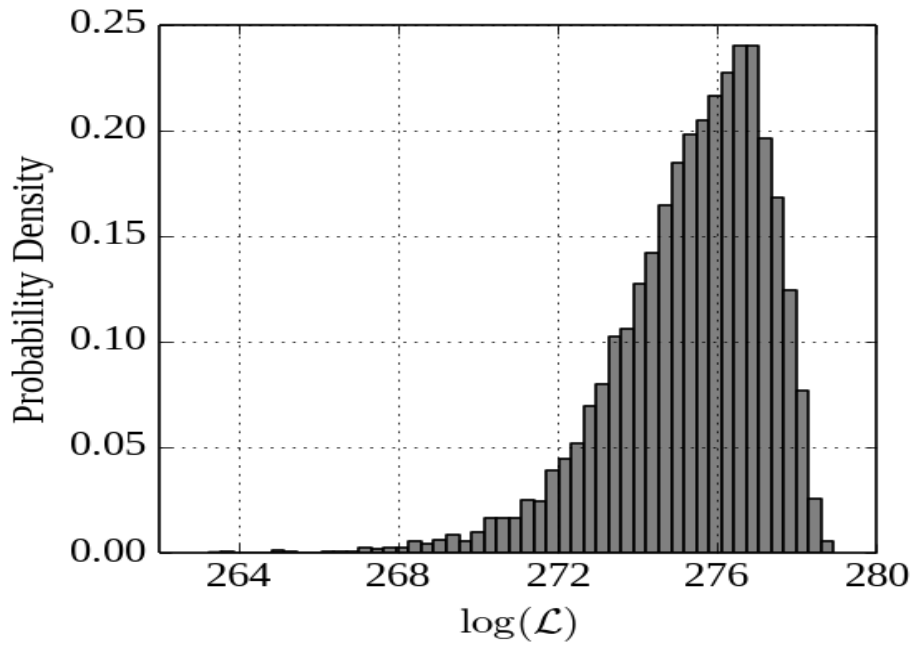


그림 10 : 10Msun-1.4Msun 질량을 가지는 블랙홀 중성자의  
로그라이클리후드

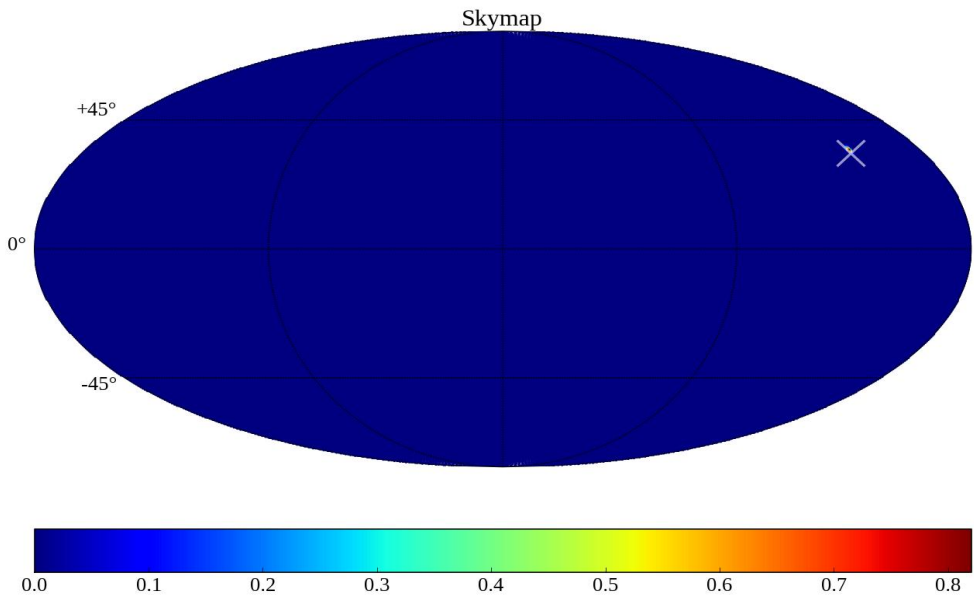


그림 11 : 10Msun-1.4Msun 질량을 가지는 블랙홀 중성자의 위치를  
나타내는 하늘지도

밀도가  $10.0M_{\text{sun}}-1.4M_{\text{sun}}$ 의 질량을 가지는 블랙홀-중성자별의 파라미터들에 대한 확률밀도함수이다(그림 12). 이때 3.5pN 위상항과, 2.5pN 진폭항을 포함하고 있는 TaylorF2Amp모델의 사용하였다.

(그림 9)와 비교해보면 진폭항이 뉴토니안일때는 위상각과 편광각의 결정이 못하고 있는 것을 확인하였다.

하지만 모든 조건은 동일하게 두고 진폭항에 2.5pN을 반영한 경우에는 확연하게 파라미터 결정을 하는 것을 확인할 수 있다.

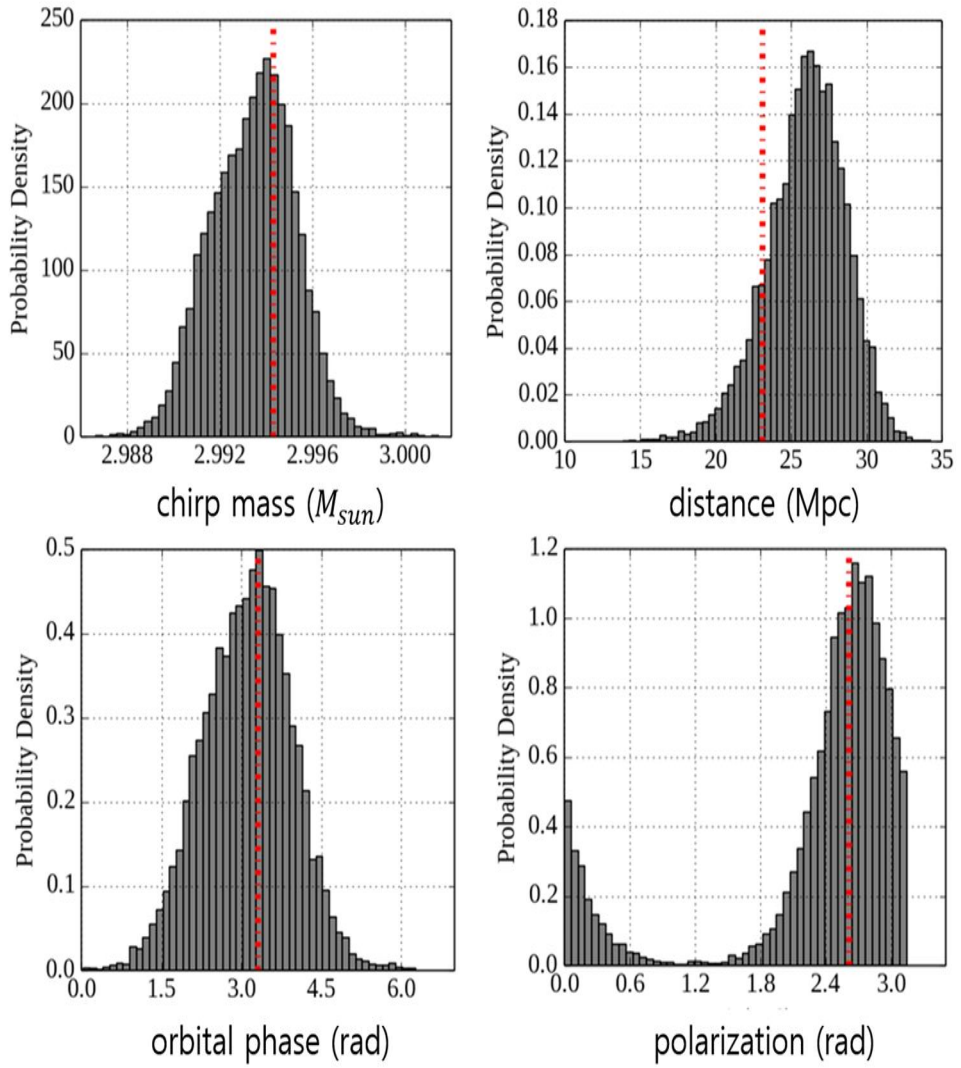


그림 12 : 10 $M_{sun}$ -1.4 $M_{sun}$  질량을 가지는 블랙홀 중성자의  
확률밀도분포함수. 위상항이 3.5pN을 포함하고 있고,  
진폭항이 2.5pN인 TaylorF2함수.

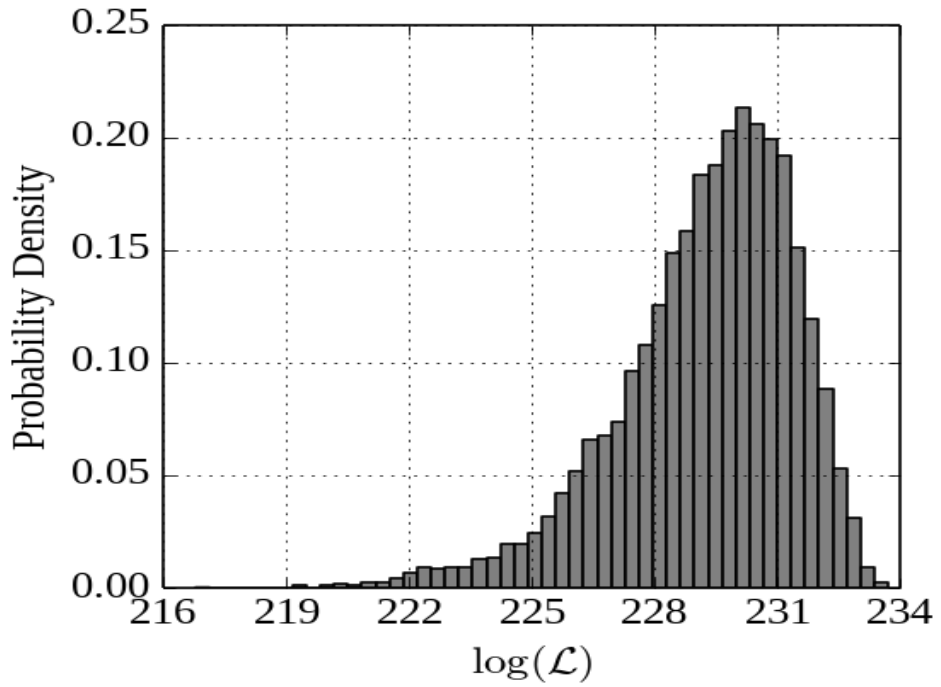


그림 13 : 10Msun-1.4Msun 질량을 가지는 블랙홀 중성자의 로그라이클리후드

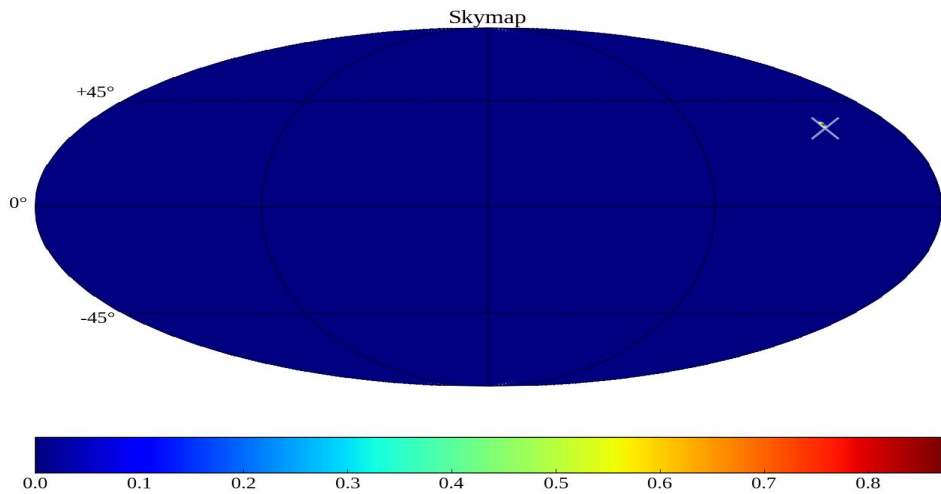


그림 14 : 10Msun-1.4Msun 질량을 가지는 블랙홀 중성자의 위치를 나타내는 하늘지도



질량이  $10.1M_{\text{sun}}-9.9M_{\text{sun}}$ 인 블랙홀-블랙홀 쌍성계의 파라미터들을 확률분포함수로 나타낸 것이다(그림 15). 블랙홀-중성자별의 관찰과 모든 조건을 같게 하고 모수추정을 수행한 결과이다. 전체적으로 블랙홀 중성자별과 비슷한 양상을 나타내고 있다.

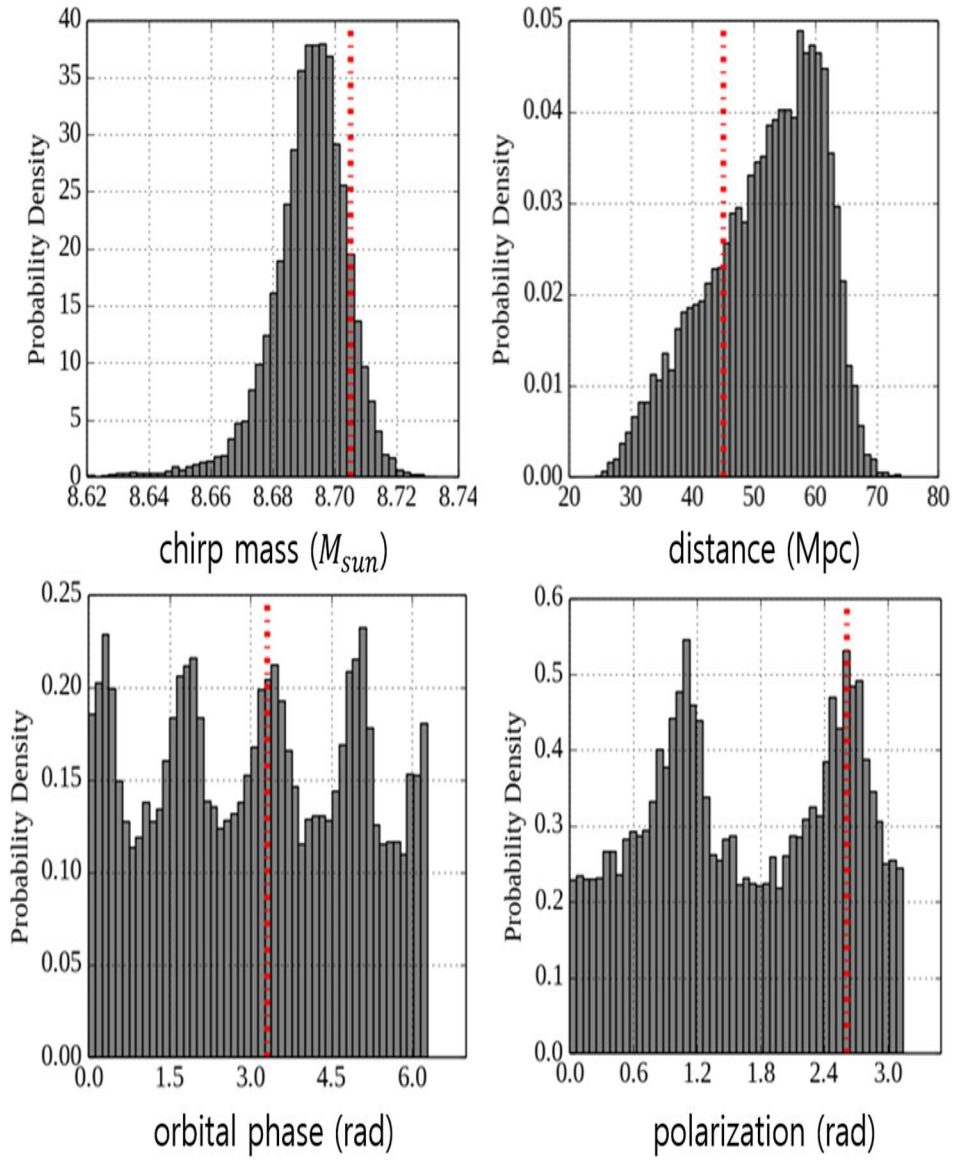


그림 15 : 10.0Msun-9.9Msun 질량을 가지는 블랙홀 블랙홀.  
진폭항이 3.5pN을 포함하고, 진폭항이 0pN인 TaylorF2모델.

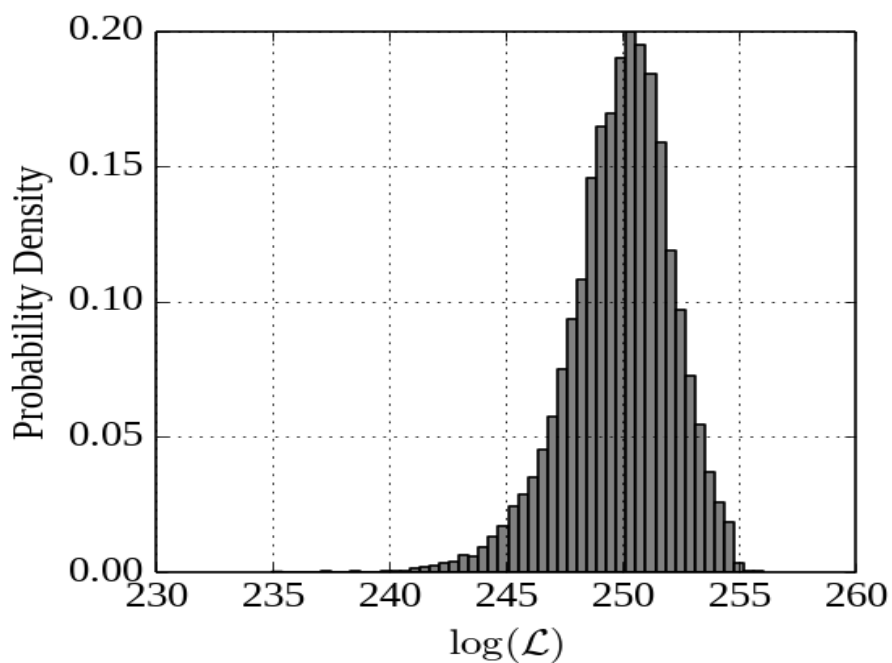


그림 16 : 10.1Msun-9.9Msun 질량을 가지는 블랙홀 블랙홀의 로그라이클리후드

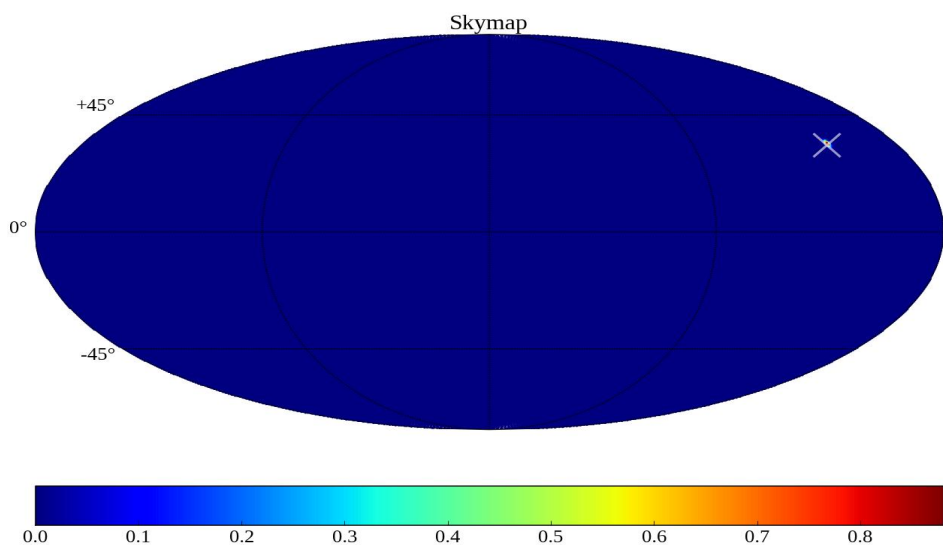


그림 17 : 10.0Msun-9.9Msun 질량을 가지는 블랙홀 블랙홀의 위치를 나타내는 하늘지도

질량이  $10.1M_{\text{sun}}-9.9M_{\text{sun}}$ 인 블랙홀-블랙홀 쌍성계의 모수추정한 결과를 확률분포함수로 나타낸 것이다(그림 18).

블랙홀-중성자별과 전체적인 결과는 비슷하지만 블랙홀-중성자별에서 진폭항이 포함되어있는 경우에 잘 결정하던 위상각과 편광각이 블랙홀-블랙홀에서는 제대로 결정하지 못하는 것을 확인하였다.

이것은 파형을 통해서도 확인한 결과로 질량비가 1에 가까울 경우, 진폭항의 효과가 파라미터결정에 큰 영향을 미치지 못하는 것으로 확인하였다. 각 중성자별의 위치는 천문학적으로 만족할만한 결과는 아니지만, 대략적인 위치를 나타내는 것을 여러 쌍성계의 스카이맵을 통하여 확인하였다.

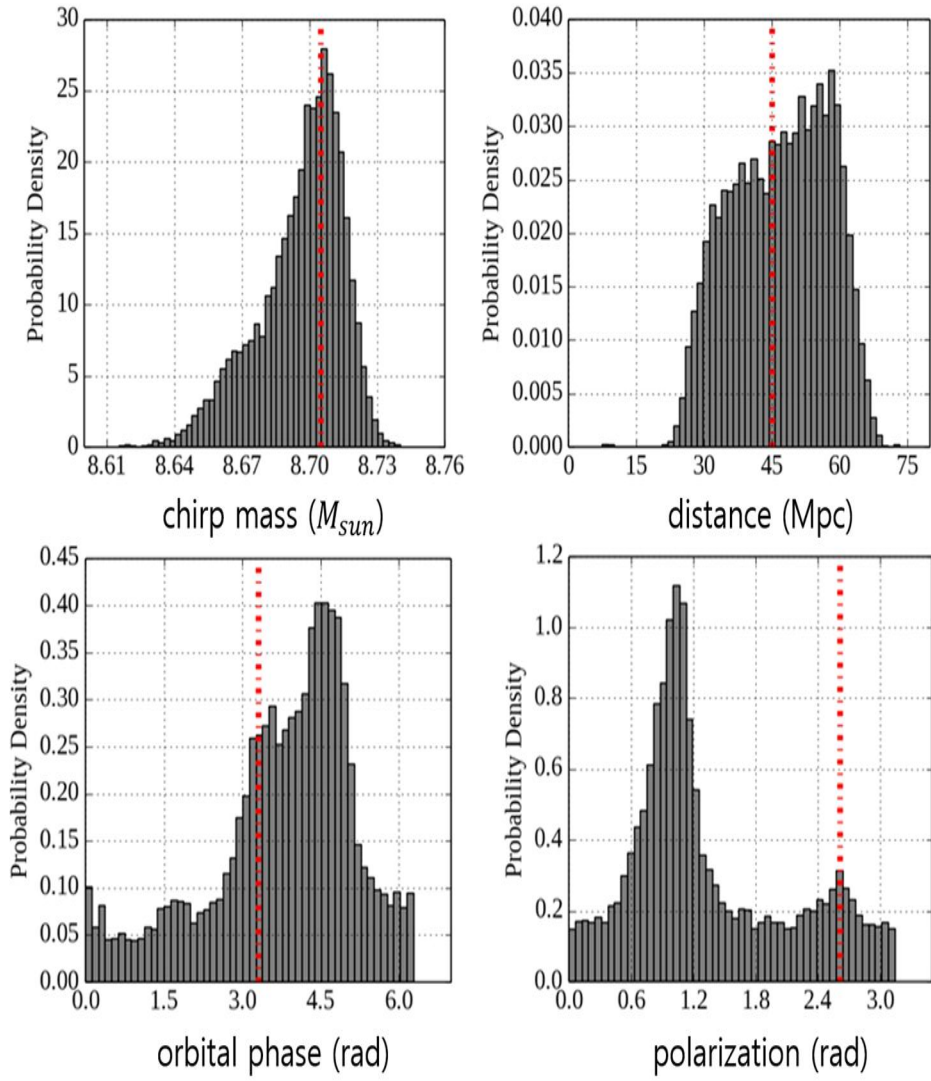


그림 18 : 10.1Msun-9.9Msun 질량을 가지는 블랙홀 블랙홀의 확률밀도함수.

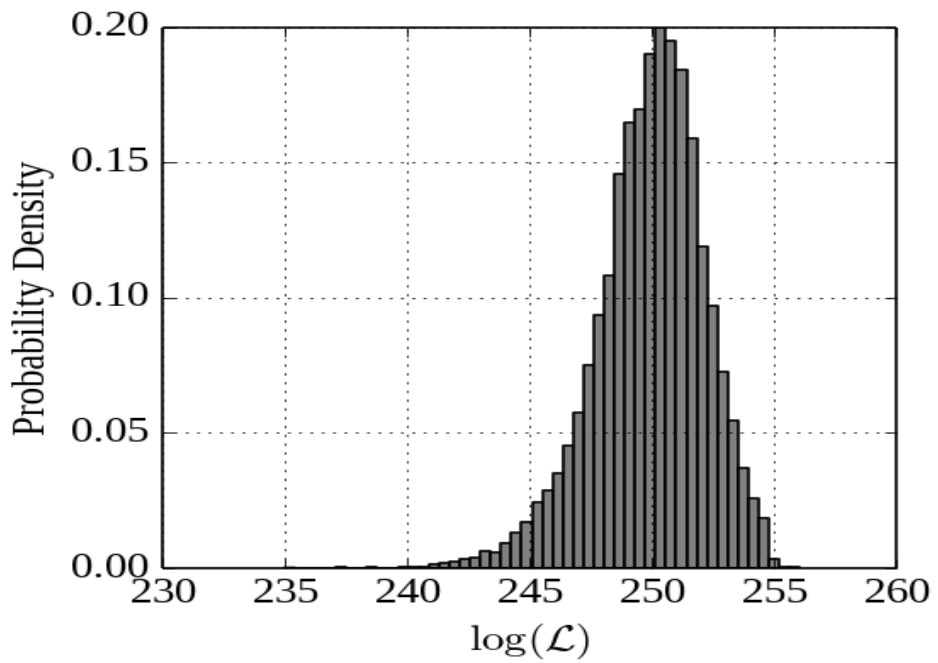


그림 19 : 10.1Msun-9.9Msun 질량을 가지는 블랙홀 블랙홀의 로그라이클리후드(loglikelihood)

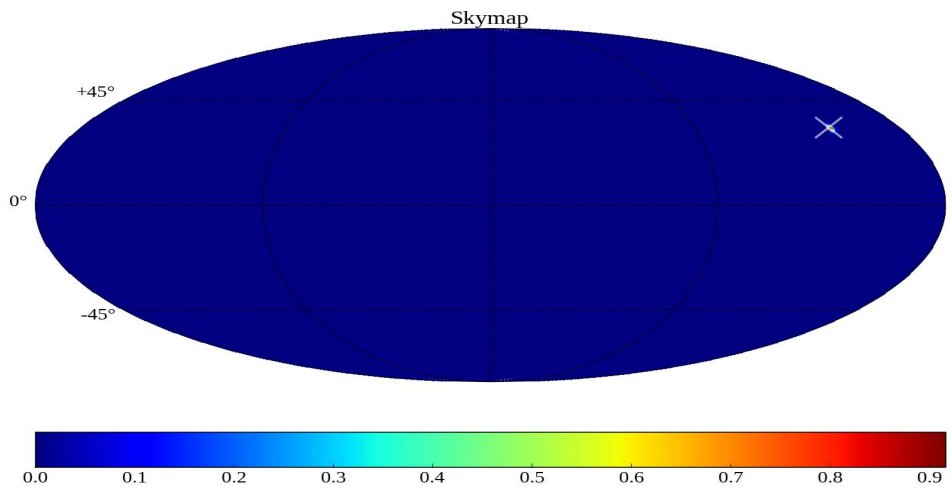


그림 20 : 10.1Msun-9.9Msun 질량을 가지는 블랙홀 블랙홀의 위치를 나타내는 하늘지도

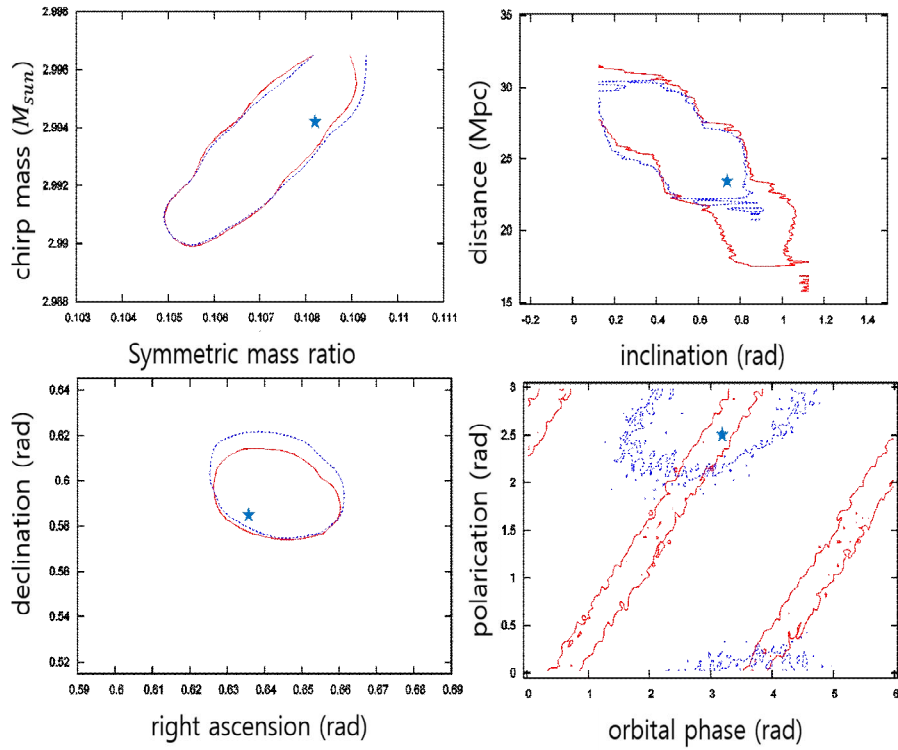


그림 21 :  $10M_{\text{sun}} - 1.4M_{\text{sun}}$  질량을 가지는 블랙홀-중성자별의 물리량의 상관관계 그래프.

각각의 이벤트에 따른 파라미터들의 상관관계를 나타내는 그래프이다. 그래프를 통하여 파라미터들 사이의 관계를 확인하였다(그림 21).

블랙홀 중성자별에서 TaylorF2와 TaylorF2Amp 모델을 이용하여 모수추정을 수행한 결과이다.

실선이 나타내것이 진폭향이 뉴토니안인 경우이고, 점선이 나타내는 것이 진폭향이  $2.5\text{pN}$ 까지 포함하고 있는 것이다. 질량이 각각  $10M_{\text{sun}} - 1.4M_{\text{sun}}$  인 이벤트에 대한 상관관계 그래프의 컨투어 그래프를 통하여 볼 때, 진폭향의 변화가 진폭향이  $2.5\text{pN}$ 일 경우 위상각과 편광각에 대해 확연히 결정함을 확인할 수 있다.

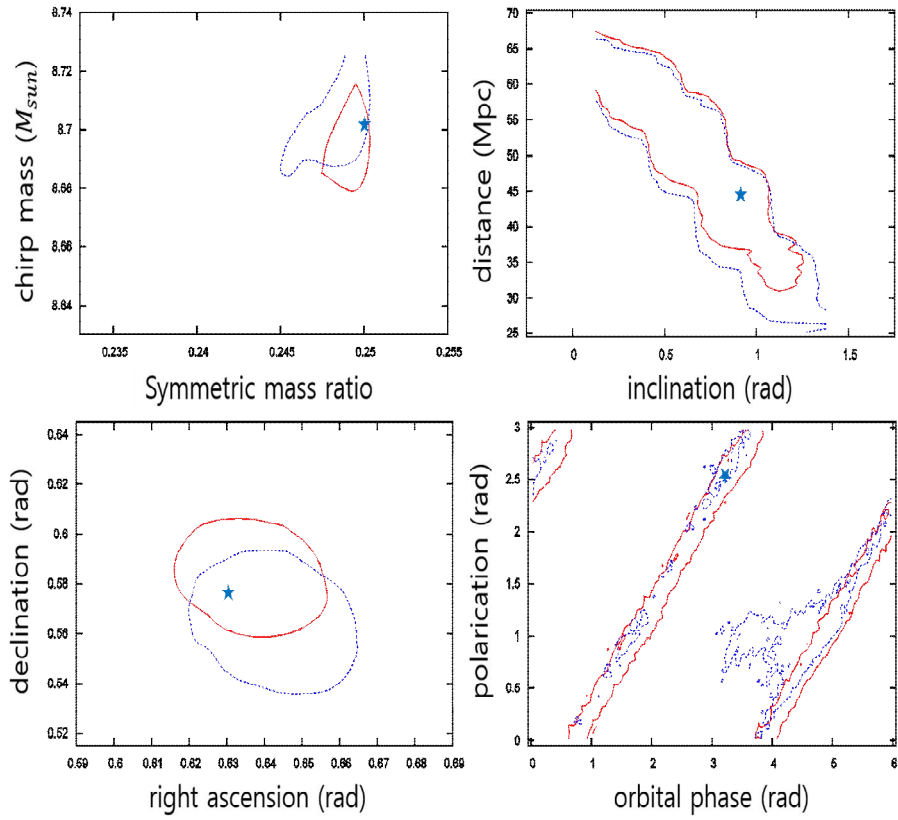


그림 22 : 10.1 $M_{\text{sun}}$  - 9.9 $M_{\text{sun}}$  질량을 가지는 블랙홀-블랙홀의 물리량의 상관관계 그래프.

질량비가 10.1 $M_{\text{sun}}$  9.9 $M_{\text{sun}}$ 인 쌍성계의 경우 파라미터들과의 상관관계 그래프를 확인할 수 있다 (그림 22).

질량비가 거의 1에 가까운 10.1 $M_{\text{sun}}$ -9.9 $M_{\text{sun}}$ 의 쌍성계에서는 진폭향의 효과가 크게 나타나지 않음을 확인하였다.



다음은 지금 까지 모수추정을 수행한 결과로 얻은 파라미터결정의 정확도를 계산한 것이다. 전체적인 정확도를 확인하였을 때 큰 차이는 보이지 않지만 위상각에서는 진폭항을 추가하였을 때 정확도가 좋아진 것을 확인할 수 있다.

		SNR	Mchirp	eta	RA	dec	cos(i)	orbital phase	polariz ation
BH NS	Newto nian	22.01	0.0018	0.000 9	0.086 4	0.034 7	0.140 5	1.835	0.8918
	2.5pN	19.72	0.0018	0.001 1	0.007 9	0.012 3	0.084 3	0.8393	0.8374
BH BH	Newto nian	25.93	0.0123	0.0023	0.0101	0.0125	0.1663	1.814	0.8887
	2.5pN	22.46	0.0188	0.0042	0.1119	0.0439	0.1956	1.388	0.7648

[ 표 2 ] 정확도 계산값

## VII. 결론

본 논문에서는 아이슈타인의 상대성이론의 마지막 검증인 중력파에 대한 기본적인 이론과 중력파를 방출하는 중력파원에 대하여 조사하였다. 이를 바탕으로 하여 중력파 검출을 위한 데이터 분석연구를 수행하였다.

직접 관측된 중력파가 있을 경우, 중력파를 내는 파원의 모든 물리량을 데이터로부터 밝혀내는 모수추정을 수행하여 어느 위치, 어떤 현상으로부터 중력파가 오는지, 그 천체의 물리량들을 통계학적으로 구한다.

현재까지는 직접 관측된 중력파가 없었기 때문에, 중력파모델에 따른 시뮬레이션된 파형을 이용하여 모수추정을 수행하여 왔다.

이 논문에서는 위상항이 3.5pN, 진폭항이 뉴토니안인 TaylorF2모델을 사용하여 모수추정을 수행하고, 새롭게 TaylorF2모델을 위상항이 3.5pN, 진폭항이 2.5pN으로 수정하여 진폭항에 pN효과를 고려하여 모수추정을 하는 연구를 수행하여 결과를 비교하였다.

밀집쌍성계의 질량비가 1과 같지 않은 경우에는 주파수 대역 전체에서 진폭항의 pN 효과를 나타냈다. 하지만 질량비가 거의 1일 경우에는  $f_{isco}$  (innermost stable circular orbit) 대역에서만 미세하게 pN효과를 확인할 수 있었다. 이를 통해 밀집쌍성계의 질량비에 따라 중력파모델을 다르게 하여, 모수추정을 수행해야 한다는 것을 확인하였다.

진폭항의 pN효과는 파형뿐만 아니라 파라미터 결정에도 많은 영향을 미치는 것을 확인하였다. 밀집쌍성계의 질량비가 1과 같지 않을 경우에 위상각과 편광각의 물리량을 명확하게 결정할 수 있으며, 중력파원의 천구상의 위치와 거리의 결정에는 큰 영향을 미치지 않음을 알 수 있었다. 하지만 밀집쌍성계의 질량비가 1에 가까울 경우에는 물리량결정 또한 큰 효과를 나타내지 않았다.

본 논문에서는 중력파모델인 TaylorF2에 고차원인 진폭항까지 추가 하여 파형과, 모수추정에서의 진폭항의 효과를 비교하였다.

따라서 모수추정 방법으로 직접적인 중력파의 검출이 있을 경우, 중력파원의 물리적 매개변수를 결정할 때 목표로 하는 물리적 매개변수에 따라 올바른 중력파형 모델을 선택하는 것이 효과적이라는 것을 확인 하였다.

## VIII. 인사말

우선 학부전공과는 다른 길에 들어선 저를 이끌어 주신 이형원 교수님. 부족한 저에게 항상 가르침을 전해주고자 하셨던 교수님께 머리 숙여 깊이 감사드립니다. 항상 격려의 말과 많은 가르침을 주신 명연수 교수님 정말 감사드립니다. 타 지역에서 논문지도를 위해 김해까지 방문하여 많은 도움을 주신 이창환 교수님께도 감사드립니다.

그리고 항상 격려의 말씀을 해주시는 컴퓨터시뮬레이션학과 모든 교수님 들께 깊은 감사의 말씀을 드립니다.

한국중력파연구원단(KGWG)의 일원으로서 항상 연구에 대한 조언을 해주시는 이형목 교수님과 그 외 한국중력파연구원단 분들에게도 감사드립니다. 처음 시작한 일이라 기본적으로 알지 못하는 부분을 하나하나 알려주신 김정리박사님께 감사드리고, 하나하나 세세하게 부족한 부분을 미리 알려주시고 챙겨주신 김경이 박사님께 감사합니다.

마지막으로 지금까지 공부하는 저를 언제나 지원해주시는 사랑하는 우리 가족인 부모님과 언니, 형부 그리고 내 동생 감사합니다.

공부에 쌓인 나의 스트레스를 항상 받아주는 내친구 닭고기, 뽕어, 김민녕에게 고마움을 전하며, 항상 나를 응원해 주는 박돼지님에게 감사함을 전합니다. 마지막으로 공부에 지친 저를 항상 다시 일어날 수 있게 해주는 박지희언니와 강동혜언니에게 감사의 말씀을 전합니다.

모든 고마운 분들에게 정말 감사합니다.

## IX.참고문헌

- [1] 김경민. 짧은 감마선 폭발과 관련된 중력파 검출연구. 한양대학교. 2015. 1:152
- [2] 이형목. 중력파. 한국물리학회지 물리학과 첨단기술. 2011. 1:4 DOI: 10.3938/PhiT.20.012
- [3] 이형목, 이창환, 강궁원 등. 천문한논총. 2011. 26:71~87 DOI : 10.5303/PKAS.2011.26.2.071
- [4] 조희석. Gravitational Waves from Compact Binary Coalescence and Parameter Measurement Error. 부산대학교. 2013. 1:126
- [5] Andrew Gelman, John B. Carlin, Hal S. Stern et al. Bayesian Data Analysis. CHAPMAN & HALL/CRC. 200.
- [6] Arun, K. G., Buonanno, Alessandra et al. Higher-order spin effects in the amplitude and phase of gravitational waveforms emitted by inspiraling compact binaries: Ready-to-use gravitational waveforms. 2009. Physical Review D, vol. 79, Issue 10, id. 104023
- [7] Christian Rover. Bayesian inference on astrophysical binary inspirals based on gravitational-wave measurements. The University of Auckland. 2007. 1:1
- [8] Cutler, Curt; Thorne, Kip S. An Overview of Gravitational-Wave Sources. 2002. eprint arXiv:gr-qc/0204090
- [9] Droz, Serge, Knapp, Daniel J., Poisson, Eric. Gravitational waves from inspiraling compact binaries: Validity of the stationary-phase approximation to the Fourier transform. 1999. Physical Review D (Particles, Fields, Gravitation, and Cosmology), Volume 59, Issue 12, 15 June 1999, id. 124016
- [10] Evan Ochsner. Improving Analytical Templates and Searching for Gravitational Waves From Coalescing Black Hole Binaries. University of Maryland. 2010. 1:408
- [11] J. D. E. Creighton and W. G. Anderson, Gravitational-Wave Physics and Astronomy (Wiley-VCH, Weinheim, Germany, 2011).
- [12] Owen, Benjamin J.; Sathyaprakash, B. S. Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement. 1999. Physical Review D (Particles, Fields, Gravitation, and Cosmology), Volume 60, Issue 2, 15 July 1999, id. 022002
- [13] Poisson and Will. Gravitational waves from inspiraling

compact binaries: Parameter estimation using second-post-Newtonian waveforms. 1995. Physical Review D (Particles, Fields, Gravitation, and Cosmology), Volume 52, Issue 2, 15 July 1995, pp.848-855

[14] Rodriguez, Carl L.; Farr, Benjamin; Farr, Will M. Inadequacies of the Fisher Information Matrix in gravitational-wave parameter estimation. 2013. Physical Review D, vol. 88, Issue 8, id. 084013

[15] Vivien Raymond. Parameter Estimation Using Markov Chain Monte Carlo Methods for Gravitational Waves from Spinning Inspirals of Compact Objects. NORTHWESTERN UNIVERSITY. 2012. 1:127

## 부록 A. LIGO Alogorithm Library Build

### - LAL Download

```
$ git config --global user.name "NAME"
```

```
$ git config --global user.email LIGO MAIL
```

```
$ git clone LIGO PATH :/usr/local/git/lalsuite.git
```

```
password : *****
```

- lalsuite 폴더에 buildicc.sh, setup-lscsoft.sh 복사

- lalsuite와 같은 위치에 master 폴더 만들기

- master 하위에 - etc - lscsoft ( etc 폴더 생성후 lscsoft 파일 복사 )

- opt - lscsoft ( 두개의 폴더 생성 )

```
$ touch ~/master/opt/lscsoft/lscsoft-user-env.sh (빈파일 생성)
```

```
$ nohup ./buildicc.sh &
```

```
$ tail -f nohup.out
```

### - LAL Update

```
$ git pull
```

```
$ nohup ./buildicc.sh &
```

- Directory Compile (해당하는 폴더에서) buildicc.sh file

```
$ export CC=icc
```

```
$ module load openmpi-intel-x86_64
```

```
$ source /opt/intel/bin/iccvars.sh intel64
```

```
$ export PYTHONPATH=/usr/lib64/python.6/site-packages/
```

```
$ source working directory path/etc/lscsoft
```

```
$ ./00boot
```

```
$ ./configure --prefix=/working directory path --disable-debug  
--disable-gcc-flags --enable-mpi
```

```
$ make
```

```
$ make install
```

## 부록 B. job submit script

- submit file

```
universe    = vanilla
getenv      = true
executable  = /usr/lib64/openmpi_intel/bin/mpirun
arguments   =          -np          11          working
directory/lalsuite/lalapps/src/inspiral/posterior/mpi/lalinference_m
cmc --ifo H1 --ifo L1 --ifo V1 --H1-cache LALLIGO --L1-cache
LALLIGO --V1-cache LALVirgo --H1-flow 30 --L1-flow 30
--V1-flow 30 --trigtime 984383679 --psdstart 894383379.0
--psdlength 512.0 --srate 4096 --seglen 16 --inj
/data/ligo/scratch/mcmc/injections/taylorF2Ampinjections.xml
--event 0 --inj-fref 100 --inj-spinOrder 0 --inj-tidalOrder 0
--approx TaylorF2AmpthreePointFivePN --fref 100 --spinOrder 0
--tidalOrder 0 --amporder 0 --Niter 10000000 --noSpin
--dataseed 2001 --radiation-frame #--tempVerbose --propTrack

output      = $(Cluster)-$(Process).modified.event0.seglen16.out
error       = $(Cluster)-$(Process).modified.event0.seglen16.err
log         = $(Cluster)-$(Process).modified.event0.seglen16.log
request_cpus = 11
request_memory = 11*2048

queue 1
```



## - MCMC option

---

### General Algorithm Parameters

---

(--Niter N)            Number of iterations (2\*10<sup>7</sup>).  
(--Neff N)            Number of effective samples.  
                      (ends if chain surpasses Niter)  
(--Nskip N)           Number of iterations between disk save (100)  
(--trigSNR SNR)        Network SNR from trigger,  
                          used to calculate tempMax (injection SNR).  
(--randomseed seed)    Random seed of sampling distribution (random).  
(--adaptTau)           Adaptation decay power, results in adapt length of  
10<sup>tau</sup> (5).  
(--noAdapt)            Do not adapt run.

---

### Likelihood Functions

---

(--zeroLogLike)            Use flat, null likelihood.  
(--studentTLikelihood)    Use the Student-T Likelihood  
                          that marginalizes over noise.  
(--correlatedGaussianLikelihood) Use analytic,  
                          correlated Gaussian for Likelihood.  
(--bimodalGaussianLikelihood) Use analytic,  
                          bimodal correlated Gaussian for Likelihood.  
(--rosenbrockLikelihood)    Use analytic,  
                          Rosenbrock banana for Likelihood.  
(--analyticnullprior)      Use analytic null prior.  
(--nullprior)            Use null prior in the sampled parameters.  
(--noiseonly)            Use signal-free log likelihood (noise model only).

---

### Noise Model

---

(--psdFit)                Run with PSD fitting  
(--psdNblock)            Number of noise parameters per IFO channel  
(--psdFlatPrior)        Use flat prior on psd parameters (Gaussian)

(--glitchFit)	Run with glitch fitting
(--glitchNmax)	Maximum number of glitch basis functions per IFO (20)

---

#### Proposals

---

(--rapidSkyLoc)	Use rapid sky localization jump proposals.
(--covarianceMatrix file)	Find the Cholesky decomposition of the covariance matrix for jumps in file.
(--noProposalSkyRing)	Disable the proposal that rotates sky position around vector connecting any two IFOs in network.
(--noProposalCorrPsiPhi)	Disable the proposal that jumps along psi-phi correlation
(--noDifferentialEvolution)	Disable the differential-evolution proposal
(--differential-buffer-limit)	Limit the number of stored differential-evolution points

---

#### Parallel Tempering Algorithm Parameters

---

(--inverseLadder)	Space temperature uniform in $1/T$ , rather than geometric.
(--tempLadderBottomUp)	Construct the a geometric temperature ladder with $\text{tempDelta}=1+\sqrt{2/n\text{Par}}$ .
(--tempSkip N)	Number of iterations between temperature swap proposals (100).
(--tempKill N)	Iteration number to stop temperature swapping (Niter).
(--tempMin T)	Lowest temperature for parallel tempering (1.0).
(--tempMax T)	Highest temperature for parallel tempering (50.0).
(--anneal)	Anneal hot temperature linearly to $T=1.0$ .
(--annealStart N)	Iteration number to start annealing ( $5 \cdot 10^5$ ).
(--annealLength N)	Number of iterations to anneal all chains to $T=1.0$ ( $1 \cdot 10^5$ ).

-----  
Output  
-----

(--data-dump)                      Output waveforms to file.  
(--adaptVerbose)                  Output parameter jump sizes and acceptance rate  
stats to file.  
(--tempVerbose)                  Output temperature swapping stats to file.  
(--propVerbose)                  Output proposal stats to file.  
(--propTrack)                    Output useful info for track proposal behavior.  
(--outfile file)                Write output files <file>.  
                                 <chain\_number> (PTMCMC.output.<random\_seed>.<chain\_number>).

– condor submit

#!/bin/sh

source /opt/intel/bin/iccvars.sh intel64  
module load openmpi-intel-x86\_64  
export masterdir=/working directory/master  
source \$masterdir/etc/lscsoftrc

export PYTHON=/usr/lib64/python2.6/site-packages

date

cd home/directory

- post processing file

\*post processing 할 Job의 PTMCMC.output.\*.00파일을 가지고 실행.

```
#!/bin/bash
```

```
export masterdir=directory path
```

```
${masterdir}/bin/test.py -d $dd/PTMCMC.output.*.00 -d  
$dd/PTMCMC.output.*.00 -d $dd/PTMCMC.output.*.00  
--lalinfmtcmc --deltaLogL 4.5 --downsample=10000 --skyres=0.5  
--dievidence --outpath=path -i xmlFilepath -eventnum 3
```

```
exit
```

- post processing option

```
·parser.add_option("-i","--inj",dest="injfile",help="SimInsipral injection  
file",metavar="INJ.XML",default=None)  
·parser.add_option("-t","--trig",dest="trigfile",help="Coinc XML  
file",metavar="COINC.XML",default=None)  
·parser.add_option("--skyres",dest="skyres",help="Sky resolution to use to  
calculate sky box size",default=None)  
·parser.add_option("--xflag",action="store_true",default=False,help="(inspne  
st) Convert x to iota.")  
·parser.add_option("--lalinfmtcmc",action="store_true",default=False,help="(LALInferenceMCMC) Parse input from LALInferenceMCMC.")  
·parser.add_option("--oldMassConvention",action="store_true",default=False,  
help="(LALInferenceMCMC) if activated, m2 >m1; otherwise m1 >m2 in  
PTMCMC.output.*.00")
```

```

#FM
·parser.add_option("--fm",action="store_true",default=False,help="(followup
MCMC) Parse input as if it was output from followupMCMC.")
# ACF plots off?
·parser.add_option("--no-acf",      action="store_true",      default=False,
dest="noacf")
# Turn on 2D kdes
·parser.add_option("--twodkdeplots",  action="store_true",  default=False,
dest="twodkdeplots")
# Turn on R convergence tests
·parser.add_option("--RconvergenceTests",          action="store_true",
default=False, dest="RconvergenceTests")
·parser.add_option("--nopdfs",action="store_false",default=True,dest="nopdf
s")
·parser.add_option("-c","--covarianceMatrix",dest="covarianceMatrices",acti
on="append",default=None,help="CSV file containing covariance (must
give accompanying mean vector CSV. Can add more than one matrix.")
·parser.add_option("-m","--meanVectors",dest="meanVectors",action="appe
nd",default=None,help="Comma separated list of locations of the
multivariate gaussian described by the correlation matrix. First line
must be list of params in the order used for the covariance matrix.
Provide one list per covariance matrix.")
·parser.add_option("--email",action="store",default=None,type="string",meta
var="user@ligo.org",help="Send an e-mail to the given address with a
link to the finished page.")
·parser.add_option("--archive",action="store",default=None,type="string",me
tavar="results.tar.gz",help="Create the given tarball with all results")
·parser.add_option("--psdfiles",action="store",default=None,type="string",me
tavar="H1,L1,V1",help="comma separater list of ASCII files with PSDs,
one per IFO")
·parser.add_option("--kdecredibleregions",action="store_true",default=False
,help="If given, will use 2-step KDE trees to estimate 1-d credible
regions [default false: use greedy binning]")
(opts,args)=parser.parse_args()

```

## 부록 C. TaylorF2Amp code

```
/*
 * Copyright (C) 2014 Jeongcho Kim, Hyung Won Lee, Chunglee Kim
 * All equations and Appendix D cited here are in Evan Ochsner's thesis.
 * Adapting features in:
 *   - LALSimInspiral.c
 *   - LALSimInspiral.h
 *   - LALSimInspiralTaylorF2.c
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation: either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with with program; see the file COPYING. If not, write to the
 * Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,
 * MA 02111-1307 USA
 */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <lal/XLALError.h>
#include <lal/Date.h>
#include <lal/FrequencySeries.h>
#include <lal/LALConstants.h>
#include <lal/LALDatatypes.h>
#include "LALSimInspiral.h"
#include <lal/Units.h>

#define MAX_HARMONICS 7 // maximum harmonics to integrate
#define MAX_AMP_ORDER 5 // maximum PN order

/**
 * Find the least nonnegative integer power of 2 that is
 * greater than or equal to n. Inspired by similar routine in gstlal.
 */
```

```

static size_t CeilPow2(double n) {
    double signif;
    int exponent;
    signif = frexp(n, &exponent);
    if (signif < 0)
        return 1;
    if (signif == 0.5)
        exponent -= 1;
    return ((size_t) 1) << exponent;
}

/* Calculate the spin corrections for TaylorF2
    reference -> <http://arxiv.org/pdf/0810.5336v3.pdf>
*/

typedef struct tag_sf2_spin_corr_amp {
    REAL8 beta;
    REAL8 sigma;
    REAL8 gamma;
}sf2_spin_corr_amp;
typedef struct tag_sf2_amp_corr_param {
    REAL8 costh;      // cos(theta), cosine of inclination of the source
    REAL8 sinth;      // sin(theta), sine of inclination of the source
    REAL8 f_cross;    // F_cross
    REAL8 f_plus;     // F_plus
    REAL8 m;          // m1+m1 total mass in solar mass
    REAL8 eta;        // eta = m1*m1/m^2
    REAL8 delta;      // delta = (m1-m1)/m
    REAL8 s1x;        // x component of spin1
    REAL8 s1y;        // y component of spin1
    REAL8 s1z;        // z component of spin1
    REAL8 s2x;        // x component of spin2
    REAL8 s2y;        // y component of spin2
    REAL8 s2z;        // z component of spin2
    REAL8 LNhatx;     // x component of LNhat
    REAL8 LNhaty;     // y component of LNhat
    REAL8 LNhatz;     // z component of LNhat
    REAL8 Xs_x;
    REAL8 Xs_y;
    REAL8 Xs_z;
    REAL8 Xa_x;
    REAL8 Xa_y;
    REAL8 Xa_z;
    REAL8 Xs2;        // Xs*Xs (=chi_s dot chi_s)

```

```

REAL8 Xa2;          // Xa*Xa (=chi_a dot chi_a)
REAL8 XsXa;         // Xs dot Xa (=xchi_s dot chi_a)
REAL8 XsLn;         // chi_s dot Ln_hat
REAL8 XaLn;         // chi_a dot Ln_hat
REAL8 cosh2;
REAL8 cosh3;
REAL8 cosh4;
REAL8 cosh5;
REAL8 cosh6;
REAL8 sinh2;
REAL8 sinh3;
REAL8 sinh4;
REAL8 sinh5;
} sf2_amp_corr_param;

// add new parameters for spin of the companion 2 to calculate
// Xs(=chi_s) and Xa(=chi_a), also it need to calculate XsLn(=chi_s dot Ln_hat)
// XaLn(=chi_a dot Ln_hat)
sf2_spin_corr_amp sf2_spin_corrections_amp(
    const REAL8 m1,          /** mass of companion 1 (solar masses) */
    const REAL8 m2,          /** mass of companion 2 (solar masses) */
    const REAL8 S1x,         /** x component of the spin of companion 1 */
    const REAL8 S1y,         /** y component of the spin of companion 1 */
    const REAL8 S1z,         /** z component of the spin of companion 1 */
    const REAL8 S2x,         /** x component of the spin of companion 2 */
    const REAL8 S2y,         /** y component of the spin of companion 2 */
    const REAL8 S2z,         /** z component of the spin of companion 2 */
    const REAL8 lnhatx,      /** x component of the vector Ln hat */
    const REAL8 lnhaty,      /** y component of the vector Ln hat */
    const REAL8 lnhatz      ) ; /** z component of the vector Ln hat */

sf2_spin_corr_amp sf2_spin_corrections_amp(
    const REAL8 m1,          /** mass of companion 1 (solar masses) */
    const REAL8 m2,          /** mass of companion 2 (solar masses) */
    const REAL8 S1x,         /** x component of the spin of companion 1 */
    const REAL8 S1y,         /** y component of the spin of companion 1 */
    const REAL8 S1z,         /** z component of the spin of companion 1 */
    const REAL8 S2x,         /** x component of the spin of companion 2 */
    const REAL8 S2y,         /** y component of the spin of companion 2 */
    const REAL8 S2z,         /** z component of the spin of companion 2 */
    const REAL8 lnhatx,      /** x component of the vector Ln hat */
    const REAL8 lnhaty,      /** y component of the vector Ln hat */
    const REAL8 lnhatz      /** z component of the vector Ln hat */
) ; /** z component of the spin of companion 2 */

```



```

{
    sf2_spin_corr_amp spin_corrections:
    REAL8 M = m1 + m2;
    REAL8 eta = m1 * m2 / (M * M);
    REAL8 delta = (m1 - m2) / (m1 + m2);
    REAL8 Xs_x = 0.5*(S1x+S2x);
    REAL8 Xs_y = 0.5*(S1y+S2y);
    REAL8 Xs_z = 0.5*(S1z+S2z);
    REAL8 Xa_x = 0.5*(S1x-S2x);
    REAL8 Xa_y = 0.5*(S1y-S2y);
    REAL8 Xa_z = 0.5*(S1z-S2z);
    REAL8 Xs2 = Xs_x*Xs_x+Xs_y*Xs_y+Xs_z*Xs_z; // Xs*Xs (=chi_s dot chi_s)
    REAL8 Xa2 = Xa_x*Xa_x+Xa_y*Xa_y+Xa_z*Xa_z; // Xa*Xa (=chi_a dot chi_a)
    REAL8 XsXa = Xs_x*Xa_x + Xs_y*Xa_y + Xs_z*Xa_z; // Xs dot Xa (=xchi_s dot
chi_a)
    REAL8 XsLn = (Xs_x*lnhatx)+(Xs_y*lnhaty)+(Xs_z*lnhatz); // chi_s dot Ln_hat
    REAL8 XaLn = (Xa_x*lnhatx)+(Xa_y*lnhaty)+(Xa_z*lnhatz); // chi_a dot Ln_hat

    // correcte beta, gamma, sigma based on the Evans thesis
    REAL8 sf2_beta = (113.L/12.L- 19.L/3.L * eta) * (XsLn) + 113.L/12.L * delta * (XaLn);

    REAL8 sf2_sigma = eta * (721.L/48.L * ((XsLn)*(XsLn) -
(XaLn)*(XaLn))-247.L/48.L*(Xs2 - Xa2));
    sf2_sigma += (1-2*eta)* (719/96.0 * ((XsLn)*(XsLn) + (XaLn)*(XaLn)) - 233.L/96.L *
(Xs2 + Xa2));
    sf2_sigma += delta * (719/48.0 *(XsLn)*(XaLn) - 233.L/48.L * XsXa);

    // should be check the sign of gamma with respect to the Evans thesis Eq. (4.84)
    REAL8 sf2_gamma = (732985.L/2268.L - 24260.L/81.L * eta - 340.L/9.L * (eta*eta) ) *
(XsLn);
    sf2_gamma += (732985.L/2268.L +140.L/9.0L * eta) * delta * XaLn;

    spin_corrections.beta = sf2_beta;
    spin_corrections.sigma = sf2_sigma;
    spin_corrections.gamma = -sf2_gamma;
    return spin_corrections;
}

/**
calculate coefficients for each SPA phase PN order
assume the maximum twice PN order to be 9, array size is 10
*/

```

```

int sf2_psi_SPA_coeffs_PN_order(
    REAL8 *PN_coeffs, /** coeffs for each PN order*/
    const int phaseO, /**< twice PN phase order */
    const sf2_spin_corr_amp *spin_corrections, /** spin correction coeffs. Eqs.
(4.82,83,84)*/
    const REAL8 eta /** Evans Eq. (4.12) */
);
/**
calculate coefficients for each SPA phase PN order
assume the maximum twice PN order to be 9
*/
int sf2_psi_SPA_coeffs_PN_order(
    REAL8 *PN_coeffs, /** coeffs for each PN order*/
    const int phaseO, /**< twice PN phase order */
    const sf2_spin_corr_amp *spin_corrections, /** spin correction coeffs. Eqs.
(4.82,83,84)*/
    const REAL8 eta /** Evans Eq (4.12) */
)
{
    REAL8 euler_number = 0.5772156649015328606065120900824024310421;
    PN_coeffs[0] = 1.0; /** 0th oprder newtonian */
    PN_coeffs[1] = 0.0; /** 0.5 PN order v^1*/
    PN_coeffs[2] = 3715.0/756.0 + 55.0*eta/9.0; /** 1.0 PN order v^2*/
    PN_coeffs[3] = 4.0*spin_corrections->beta - 16.0*LAL_PI ; /** 1.5 PN order v^3*/
    PN_coeffs[4] = 15293365.0/508032.0 + 27145.0*eta/504.0 + 3085.0*eta*eta/72.0
        - 10.0*spin_corrections->sigma; /** 2.0 PN order v^4*/
    PN_coeffs[5] = 38645.0*LAL_PI/756.0 - 65.0*LAL_PI*eta/9.0 -
spin_corrections->gamma; /** 2.5 PN order v^5*/
    PN_coeffs[6] = 11583231236531.0/4694215680.0 - 6848.0*euler_number/21.0 -
640.0*LAL_PI*LAL_PI/3.0
        + (2255.0*LAL_PI*LAL_PI/12.0 - 15737765635.0/3048192.0)*eta
        + 76055.0*eta*eta/1728.0 - 127825.0*eta*eta*eta/1296.0; /** 3.0 PN
order v^6 from Evans nb file*/
    PN_coeffs[7] = LAL_PI*(77096675.0/254016.0 + 378515.0*eta/1512.0
        - 74045.0*eta*eta/756.0); /** 3.5 PN order v^7 from Evans nb file*/
    PN_coeffs[8] = 0.0; /** 4.0 PN order v^8*/
    PN_coeffs[9] = 0.0; /** 4.5 PN order v^9*/
    return 0;
}

/** prototype for new function for psi_SPA, SPA phase calculation in Eq. (4.81)
REAL8 sf2_psi_SPA(
    const REAL8 f, /** frequency of GW */
    const int k, /** harmonics number */

```

```

const REAL8 shft,      /** shift factor  $2\pi t_c$  */
const REAL8 phic,      /** orbital coalescence phase (rad) */
const int phaseOrder,  /** twice PN phase order */
const REAL8 *PN_coeffs, /** coeffs for each PN order */
const REAL8 m,         /** total mass */
const REAL8 eta        /** symmetric mass ratio Eq (4.12) */
);
// definition of psi SPA calculation in Eq. (4.81)
REAL8 sf2_psi_SPA(
    const REAL8 f,      /** frequency of GW */
    const int k,        /** harmonics number */
    const REAL8 shft,    /** shift factor  $2\pi t_c$  */
    const REAL8 phic,    /** orbital coalescence phase (rad) */
    const int phaseOrder, /** twice PN phase order */
    const REAL8 *PN_coeffs, /** coeffs for each PN order this should be calculated
before calling this function */
    const REAL8 m,      /** total mass */
    const REAL8 eta      /** symmetric mass ratio Eq (4.12) */
)
{
    REAL8 psi;
    REAL8 f_k = f/k;
    const REAL8 m_sec = m * LAL_MTSUN_SI; /* total mass in seconds */
    const REAL8 two_piM = LAL_TWOPi * m_sec;
    const REAL8 v = cbrt(two_piM*f_k);
    REAL8 vs[9]={0};
    REAL8 pre_factor;
    int i;
    // set values for power of v
    vs[0] = 1.0;
    vs[5] = v*v*v*v*v*v;
    for(i=1; i<=phaseOrder; i++)
    {
        vs[i] = vs[i-1]*v;
    }
    pre_factor = 0.0117187500000/vs[5]/eta;    //3.0/256.0/vs[5]/eta;
    // calculate phase value upto the given order
    psi = 0;
    for(i=0; i<=phaseOrder; i++)
    {
        if(i==5)
        {
            psi += PN_coeffs[i]*(1.0+3.0*log(vs[1]))*vs[i];
        }
    }
}

```

```

        else if(i==6)
        {
            psi += (PN_coeffs[i] - 6848.0/21.0*log(4.0*vs[1]))*vs[i];
        }
        else
        {
            psi += PN_coeffs[i]*vs[i];
        }
    }
    return shift*f_k - phic + pre_factor*psi;
}

// prototype for the function calculating amplitude corrections for
// given frequency, PN order, and harmonics.
// returns amplitude value of COMPLEX16 data
COMPLEX16 sf2_amp_SPA_plus(
    REAL8 f,          /** given frequency */
    INT4 n,           /** twice order of PN for amplitude correction*/
    INT4 k,           /** harmonic number */
    REAL8 fStart,     /** initial frequency */
    REAL8 fISCO,      /** the final cut frequency corresponding to f_LSO*/
    sf2_amp_corr_param *amp_corr_param /** various parameters*/
);

// prototype for the function calculating amplitude corrections for
// given frequency, PN order, and harmonics
// returns amplitude value of COMPLEX16 data
COMPLEX16 sf2_amp_SPA_plus(
    REAL8 f,          /** given frequency */
    INT4 n,           /** twice order of PN for amplitude correction*/
    INT4 k,           /** harmonic number */
    REAL8 fStart,     /** initial frequency */
    REAL8 fISCO,      /** the final cut frequency corresponding to f_LSO*/
    sf2_amp_corr_param *param /** various parameters*/
)
{
    COMPLEX16 amp;
    REAL8 f_k = f/k;
    const REAL8 m_sec = param->m * LAL_MTSUN_SI; /* total mass in seconds */
    const REAL8 two_piM = LAL_TWOPi * m_sec;
    const REAL8 v = cbrt(two_piM*f_k);
    REAL8 pre_factor;
    amp = 0.0 + 0.0j;
    if(f < fStart || f > k*fISCO) return amp; // return 0 if out of range
    //if(f < k*Fin || f > k*fISCO) return amp; // return 0 if out of range, Fin is the

```

orbital frequency at start of observation Eq. (4.87)

```
pre_factor = pow(v, n-3.5);
switch(n)
{
    case 0: // Newtonian case
        if(k==2)
        {
            amp = -(1+param->costh2);
            amp = amp/sqrt(2.0);
        }
        else
        {
            amp = 0.0 + 0.0j;
        }
        break;
    case 1: // 0.5 PN order
        if(k==1)
        {
            amp = -(5.0/8.0+param->costh2/8.0);
            amp = param->sinh*param->delta*amp;
        }
        else if(k==3)
        {
            amp = (1+param->costh2);
            amp = 9.0/8.0*param->sinh*param->delta*amp/sqrt(3.0);
        }
        else
        {
            amp = 0.0 + 0.0j;
        }
        break;
    case 2: // 1.0 PN order
        if(k==1)
        {
            amp = 1;
            amp = param->sinh*(param->delta*param->XsLn + param->XaLn)*amp;
        }
        else if(k==2)
        {
            amp = (1385.0/672.0 - 109.0*param->eta/24.0 +
                    (265.0/672.0+11.0*param->eta/24.0)*param->costh2 +
                    (-1.0/3.0 + param->eta)*param->costh4);
            amp = amp/sqrt(2.0);
        }
}
```

```

else if(k==4)
{
    amp = (1.0+param->costh2);
    amp = param->sinh2*amp*2/3.0*(3.0*param->eta-1.0);
}
else
{
    amp = 0.0 + 0.0j;
}
break;
case 3: // 1.5 PN order
if(k==1)
{
    amp = (-2119.0/5376.0 -263.0*param->eta/192.0 +
        (937.0/5376.0 -3.0*param->eta/64.0)*param->costh2 +
        (-1.0/192.0 + param->eta/96.0)*param->costh4);
    amp = param->sinh*param->delta*amp;
}
else if(k==2)
{
    amp = (-27.0/8.0*(1.0+param->costh2)*param->delta*param->XaLn +
        (-27.0/8.0*(1.0+param->costh2)+
        0.5*(9.0-7.0*param->costh2)*param->eta)*param->XsLn);
    amp = amp/sqrt(2.0);
}
else if(k==3)
{
    amp = (-6969.0/1792.0 + 81.0*param->eta/16.0 +
        (-2811.0/1792.0+27.0*param->eta/64.0)*param->costh2 +
        81.0/128.0 * (1.0-2.0*param->eta)*param->costh4);
    amp = param->sinh*param->delta*amp/sqrt(3.0);
}
else if(k==5)
{
    amp = (1.0+param->costh2);
    amp
    param->sinh3*param->delta*amp*625.0/384.0/sqrt(5.0)*(1.0-2.0*param->eta);
}
else
{
    amp = 0.0 + 0.0j;
}
break;
case 4: // 2.0 PN order

```

```

        if(k==1)
        {
            amp = (5.0*LAL_PI/8.0 + LAL_PI/8.0*param->costh2)
                  + (((11.0/40.0+5.0*log(2.0)/4.0) + (7.0/40.0 +
log(2.0)/4.0)*param->costh2))*1.0j;
            amp = param->sinh*param->delta*amp;
        }
        else if(k==2)
        {
            amp = (113419241.0/40642560.0 + 152987.0*param->eta/16128.0 -
11099.0*param->eta*param->eta/1152.0 + (165194153.0/40642560.0 -
149.0*param->eta/1792.0
+ 6709.0*param->eta*param->eta/1152.0)*param->costh2 +
(1693.0/2016.0 -5723.0*param->eta/2016.0
+13.0*param->eta*param->eta/12.0)*param->costh4 +
(-1.0/24.0 +5.0*param->eta/24.0
5.0*param->eta*param->eta/24.0)*param->costh6 +
param->eta*(721.0/96.0*(param->XsLn*param->XsLn
param->XaLn*param->XaLn) -
439.0/96.0*(param->Xs2 - param->Xa2))*(1.0 + param->costh2));
            amp = amp/sqrt(2.0);
        }
        else if(k==3)
        {
            amp = (9.0*LAL_PI/8.0*(1.0+param->costh*param->costh))
                  + ((-189.0/40.0
27.0*log(1.5)/4.0)*(1.0+param->costh*param->costh))*1.0j;
            amp = param->sinh*param->delta*amp/sqrt(3.0);
        }
        else if(k==4)
        {
            amp = (16109.0/2520.0 - 13367.0*param->eta/504.0 +
39.0*param->eta*param->eta*0.5 +
(16.0/15.0 - 16.0*param->eta/3.0
16.0*param->eta*param->eta/3.0)*param->costh2*(param->costh4 - 3.0) +
(-10733.0/2520.0 + 7991.0*param->eta/504.0 -
53.0*param->eta*param->eta/6.0)*param->costh4);
            amp = 0.5*amp;
        }
        else if(k==6)
        {
            amp = (1.0 + param->costh2);
            amp = (-81.0/40.0 + 81.0*param->eta/8.0
81.0*param->eta*param->eta/8.0)*param->sinh4*amp/sqrt(6.0);

```

```

    }
    else
    {
        amp = 0.0 + 0.0j;
    }
    break;
case 5: // 2.5 PN order
    if(k==1)
    {
        amp = (37533839.0/325140480.0 + 76171.0*param->eta/43008.0 -
            8407.0*param->eta*param->eta/4608.0 +
            (-29850823.0/325140480.0 + 56543.0*param->eta/129024.0 +
            139.0*param->eta*param->eta/576.0)*param->costh2 +
            (255.0/14336.0 - 2659.0*param->eta/64512.0 +
            127.0*param->eta*param->eta/9216.0)*param->costh4 +
            (-1.0/9216.0 + param->eta/2304.0 -
            param->eta*param->eta/3072.0)*param->costh6);
        amp = param->sinh*param->delta*amp;
    }
    else if(k==2)
    {
        amp = (85.0*LAL_PI/64.0*(1.0-4.0*param->eta)*(1.0+param->costh2))+
            ((-9.0/5.0 + 32.0*param->eta +
            (14.0/5.0 *(1.0+4.0*param->eta))*param->costh2 +
            (7.0/5.0*(1.0-4.0*param->eta))*param->costh4))*1.0j;
        amp = amp/sqrt(2.0);
    }
    else if(k==3)
    {
        amp = (-8781361.0/7225344.0 -366781.0*param->eta/17920.0 +
            15193.0*param->eta*param->eta/1280.0 +
            (-238136057.0/36126720.0 + 37829.0*param->eta/71680.0 -
            7073.0*param->eta*param->eta/1280.0)*param->costh2 +
            (-328347.0/143360.0 + 396009.0*param->eta/71680.0 -
            10557.0*param->eta*param->eta/5120.0)*param->costh4 +
            (729.0/5120.0 - 729.0*param->eta/1280.0 +
            2187.0*param->eta*param->eta/5120.0)*param->costh6);
        amp = param->sinh*param->delta*amp/sqrt(3.0);
    }
    else if(k==4)
    {
        amp = (8.0*LAL_PI/3.0*(3.0*param->eta - 1.0)*(1.0+param->costh2))+
            ((56.0/5.0 - 1193.0*param->eta/30.0 +
            32.0*log(2.0)/3.0*(3.0*param->eta - 1.0))*(1.0+param->costh2))*1.0j;
    }

```



```

        amp = 0.5*param->sinh2*amp;
    }
    else if(k==5)
    {
        amp = (-854375.0/86016.0 + 3919375.0*param->eta/129024.0 -
            160625.0*param->eta*param->eta/9216.0 +
            (40625.0/9216.0 - 40625.0*param->eta/2304.0 +
            40625.0*param->eta*param->eta/3072.0)*param->cosh2 +
            (1863125.0/258048.0 - 2519375.0*param->eta/129024.0 +
            85625.0*param->eta*param->eta/9216.0)*param->cosh4 +
            (-15625.0/9216.0 +15625.0*param->eta/2304.0 -
            15625.0*param->eta*param->eta/3072.0)*param->cosh6);
        amp = param->sinh*param->delta*amp/sqrt(5.0);
    }
    else if(k==7)
    {
        amp =(1.0+param->cosh2);
        amp = param->sinh5*param->delta*amp/sqrt(7.0)*
            (117649.0/46080.0 - 117649.0*param->eta/11520.0 +
            117649.0*param->eta*param->eta/15360.0);
    }
    else
    {
        amp = 0.0 + 0.0j;
    }
    break;
case 6: // 3.0 PN order
    amp = 0.0 + 0.0j;
    break;
case 7: // 3.5 PN order
    amp = 0.0 + 0.0j;
    break;
case 8: // 4.0 PN order
    amp = 0.0 + 0.0j;
    break;
case 9: // 4.5 PN order
    amp = 0.0 + 0.0j;
    break;
default:
    amp = 0.0 + 0.0j;
    break;
}
return pre_factor*amp;
}

```

```

COMPLEX16 sf2_amp_SPA_cross(
    REAL8 f,      /** given frequency */
    INT4 n,      /** twice order of PN for amplitude correction*/
    INT4 k,      /** harmonic number */
    REAL8 fStart, /** initial frequency */
    REAL8 fISCO,  /** the final cut frequency corresponding to f_LSO*/
    sf2_amp_corr_param *param /** various parameters*/
);

COMPLEX16 sf2_amp_SPA_cross(
    REAL8 f,      /** given frequency */
    INT4 n,      /** twice order of PN for amplitude correction*/
    INT4 k,      /** harmonic number */
    REAL8 fStart, /** initial frequency */
    REAL8 fISCO,  /** the final cut frequency corresponding to f_LSO*/
    sf2_amp_corr_param *param /** various parameters*/
)
{
    COMPLEX16 amp;
    REAL8 f_k = f/k;
    const REAL8 m_sec = param->m * LAL_MTSUN_SI; /* total mass in seconds */
    const REAL8 two_piM = LAL_TWOPI * m_sec;
    const REAL8 v = cbrt(two_piM*f_k);
    REAL8 pre_factor;
    amp = 0.0 + 0.0j;
    if(f < fStart || f > k*fISCO) return amp; // return 0 if out of range
    //if(f < k*Fin || f > k*fISCO) return amp; // return 0 if out of range, Fin is the
    orbital frequency at the start of observation Eq. (4.87)
    pre_factor = pow(v, n-3.5);
    switch(n)
    {
        case 0: // Newtonian case
            if(k==2)
            {
                amp = - 2.0*param->costh*1.0j;
                amp = amp/sqrt(2.0);
            }
            else
            {
                amp = 0.0 + 0.0j;
            }
            break;
        case 1: // 0.5 PN order
            if(k==1)

```

```

    {
        amp = - 0.75*param->costh*1.0j;
        amp = param->sinh*param->delta*amp;
    }
    else if(k==3)
    {
        amp = 2.0*param->costh*1.0j;
        amp = 9.0/8.0*param->sinh*param->delta*amp/sqrt(3.0);
    }
    else
    {
        amp = 0.0 + 0.0j;
    }
    break;
case 2: // 1.0 PN order
    if(k==1)
    {
        amp = param->costh*1.0j;
        amp = param->sinh*(param->delta*param->XsLn + param->XaLn)*amp;
    }
    else if(k==2)
    {
        amp = ((387.0/112.0-85.0*param->eta/12.0)*param->costh +
            (-4.0/3.0 + 4.0*param->eta)*param->costh3)*1.0j;
        amp = amp/sqrt(2.0);
    }
    else if(k==4)
    {
        amp = (2.0*param->costh)*1.0j;
        amp = param->sinh2*amp*2/3.0*(3.0*param->eta-1.0);
    }
    else
    {
        amp = 0.0 + 0.0j;
    }
    break;
case 3: // 1.5 PN order
    if(k==1)
    {
        amp = -(155.0/896.0 + 145.0*param->eta/96.0)*param->costh +
            5.0/96.0*(2.0*param->eta-1.0)*param->costh3)*1.0j;
        amp = param->sinh*param->delta*amp;
    }
    else if(k==2)

```

```

{
    amp = (-27.0/4.0*param->delta*param->XaLn +
        ( - 2 7 . 0 / 4 . 0
+ (5.0-4.0*param->costh2)*param->eta)*param->XsLn)*param->costh*1.0j;
    amp = amp/sqrt(2.0);
}
else if(k==3)
{
    amp = ((-6213.0/896.0 + 135.0*param->eta/16.0)*param->costh +
        135.0/64.0*(1.0-2.0*param->eta)*param->costh3)*1.0j;
    amp = param->sinh*param->delta*amp/sqrt(3.0);
}
else if(k==5)
{
    amp = (2.0*param->costh)*1.0j;
    amp
param->sinh3*param->delta*amp*625.0/384.0/sqrt(5.0)*(1.0-2.0*param->eta);
}
else
{
    amp = 0.0 + 0.0j;
}
break;
case 4: // 2.0 PN order
if(k==1)
{
    amp = -(9.0/20.0 + 3.0*log(2.0)/2.0)*param->costh
        + ((0.75*LAL_PI)*param->costh)*1.0j;
    amp = param->sinh*param->delta*amp;
}
else if(k==2)
{
    amp = ((114020009.0/20321280.0 + 133411*param->eta/8064.0 -
        7499.0*param->eta*param->eta/576.0)*param->costh +
        (5777.0/2520.0 - 5555.0*param->eta/504.0
+ 34.0*param->eta*param->eta/3.0)*param->costh3 +
        (-0.25 + 5.0*param->eta/4.0
- 5.0*param->eta*param->eta/4.0)*param->costh5 +
        param->eta*(721.0/96.0*(param->XsLn*param->XsLn
- param->XaLn*param->XaLn) -
        439.0/96.0*(param->Xs2 - param->Xa2))*2.0*param->costh)*1.0j;
    amp = amp/sqrt(2.0);
}
else if(k==3)

```

```

{
    amp = (-2.0*(-189.0/40.0 + 27.0*log(1.5)/4.0)*param->costh)
        + (9.0*LAL_PI/4.0*param->costh)*1.0j;
    amp = param->sinh*param->delta*amp/sqrt(3.0);
}
else if(k==4)
{
    amp = ((2953.0/252.0 - 12023.0*param->eta/252.0 +
        101.0*param->eta*param->eta/3.0)*param->costh +
        (-18797.0/1260.0 + 16055.0*param->eta/252.0 -
        149.0*param->eta*param->eta/3.0)*param->costh3 +
        (16.0/5.0 - 16.0*param->eta
        +
        16.0*param->eta*param->eta)*param->costh5)*1.0j;
    amp = 0.5*amp;
}
else if(k==6)
{
    amp = (2.0*param->costh)*1.0j;
    amp = (-81.0/40.0 + 81.0*param->eta/8.0
    -
    81.0*param->eta*param->eta/8.0)*param->sinh4*amp/sqrt(6.0);
}
else
{
    amp = 0.0 + 0.0j;
}
break;
case 5: // 2.5 PN order
if(k==1)
{
    amp = ((-3453823.0/54190080.0 +163015.0*param->eta/64512.0 -
        4237.0*param->eta*param->eta/2304.0)*param->costh +
        (34373.0/322560.0 - 11755.0*param->eta/32256.0 +
        631.0*param->eta*param->eta/2304.0)*param->costh3 +
        (-7.0/4608.0 + 7.0*param->eta/1152.0
        -
        7.0*param->eta*param->eta/1536.0)*param->costh5)*1.0j;
    amp = param->sinh*param->delta*amp;
}
else if(k==2)
{
    amp = ((2.0 - 282.0*param->eta/5.0)*param->costh + (-22.0/5.0 +
    94.0*param->eta/5.0)*param->costh3)+
        ((85.0*LAL_PI/32.0*(1.0 - 4.0*param->eta))*param->costh)*1.0j;
    amp = amp/sqrt(2.0);
}
}

```

```

else if(k==3)
{
    amp = ((-63633869.0/18063360.0 - 89609.0*param->eta/2560.0 +
        697.0*param->eta*param->eta/40.0)*param->costh +
        (-508689.0/71680.0 + 812727.0*param->eta/35840.0 -
        4707.0*param->eta*param->eta/320.0)*param->costh3 +
        (1701.0/2560.0 - 1701.0*param->eta/640.0 +
        5103.0*param->eta*param->eta/2560.0)*param->costh5)*1.0j;
    amp = param->sinh*param->delta*amp/sqrt(3.0);
}
else if(k==4)
{
    amp = - (56.0/5.0 - 1193.0*param->eta/30.0 +
        32.0*log(2.0)/3.0*(3.0*param->eta - 1.0))*2.0*param->costh +
        ((8.0*LAL_PI/3.0*(3.0*param->eta - 1.0))*2.0*param->costh)*1.0j;
    amp = 0.5*param->sinh2*amp;
}
else if(k==5)
{
    amp = ((-2388125.0/129024.0 + 3569375.0*param->eta/64512.0 -
        141875.0*param->eta*param->eta/4608.0)*param->costh +
        (3000625.0/129024.0 - 1598125.0*param->eta/21504.0 +
        51875.0*param->eta*param->eta/1152.0)*param->costh3 +
        (-21875.0/4608.0 + 21875.0*param->eta/1152.0 -
        21875.0*param->eta*param->eta/1536.0)*param->costh5)*1.0j;
    amp = param->sinh*param->delta*amp/sqrt(5.0);
}
else if(k==7)
{
    amp =(2.0*param->costh)*1.0j;
    amp = param->sinh5*param->delta*amp/sqrt(7.0)*
        (117649.0/46080.0 - 117649.0*param->eta/11520.0 +
        117649.0*param->eta*param->eta/15360.0);
}
else
{
    amp = 0.0 + 0.0j;
}
break:
case 6: // 3.0 PN order
    amp = 0.0 + 0.0j;
    break:
case 7: // 3.5 PN order
    amp = 0.0 + 0.0j;

```

```

        break;
    case 8: // 4.0 PN order
        amp = 0.0 + 0.0j;
        break;
    case 9: // 4.5 PN order
        amp = 0.0 + 0.0j;
        break;
    default:
        amp = 0.0 + 0.0j;
        break;
}
return pre_factor*amp;
}
/**
 * Computing the stationary phase approximation to the Fourier transform of
 * a chirp waveform with phase given by Eq.(4.82)
 * and amplitude given by Eq. (4.72) with  $C_k^{(n)}$  given in Appendix D
 * PN order is the highest order being used to calculate Eq. (4.72)
 */
int XLALSimInspiralTaylorF2AmpPlus(
    COMPLEX16FrequencySeries **h_tilde_out, /* frequency-domain waveform */
    REAL8 phic, /* orbital coalescence phase (rad) */
    REAL8 deltaF, /* sampling frequency (Hz) */
    REAL8 inclination, /* inclination of source (rad), corresponds to theta of
N_hat */
    REAL8 m1_SI, /* mass1 (kg) */
    REAL8 m2_SI, /* mass2 (kg) */
    REAL8 S1x, /* initial value of S1x */
    REAL8 S1y, /* initial value of S1y */
    REAL8 S1z, /* initial value of S1z */
    REAL8 S2x, /* initial value of S2x */
    REAL8 S2y, /* initial value of S2y */
    REAL8 S2z, /* initial value of S2z */
    REAL8 lnhatx, /* initial value of LNhatx */
    REAL8 lnhaty, /* initial value of LNhaty */
    REAL8 lnhatz, /* initial value of LNhatz */
    REAL8 fStart, /* start GW frequency (Hz) */
    REAL8 f_max0, /* ending GW frequency (Hz) */
    REAL8 r, /* distance of source (m) */
    int phaseO, /* twice PN phase order */
    int amplitudeO /* twice PN amplitude order */
)
{
    const REAL8 lambda = -1987./3080.;

```

```

const REAL8 theta = -11831./9240.;

/* external: SI; internal: solar masses */
const REAL8 m1 = m1_SI / LAL_MSUN_SI;
const REAL8 m2 = m2_SI / LAL_MSUN_SI;
const REAL8 m = m1 + m2;
const REAL8 m_sec = m * LAL_MTSUN_SI; /* total mass in seconds */
const REAL8 eta = m1 * m2 / (m * m);
const REAL8 delta = (m1 - m2)/ m;
const REAL8 piM = LAL_PI * m_sec;
const REAL8 vISCO = 1. / sqrt(6.);
const REAL8 fISCO = vISCO * vISCO * vISCO / piM;
const REAL8 v0 = cbrt(piM * fStart);
REAL8 shft, amp0, f_max, f;
size_t i, n, k, iStart, iEnd;
int mm;
COMPLEX16 *data = NULL; /** actual storage for amplitude data pointer */
COMPLEX16FrequencySeries *htilde; /** waveform data */
LIGOTimeGPS tC = {0, 0};
REAL8 overall_factor;
REAL8 PN_coeffs_SPA[10]; /** PN order coefficients for SPA phase calculation*/
sf2_spin_corr_amp spin_corrections_SPA; /** spin correction coeffs beta, sigma,
gamma Eqs. (4.82,83,84)*/
REAL8 f_plus, f_cross, costh, sinth;
COMPLEX16 amp;
sf2_amp_corr_param amp_corr_param;

REAL8 alpha, alpha_ref, zeta, zeta_ref, beta, phasing;
COMPLEX16 prec_fac;

/** calculate frequency independent coefficients */
overall_factor = m*m/r*LAL_MTSUN_SI*LAL_MRSUN_SI*sqrt(5.0*LAL_PI*eta/48.0); /**
overall factor of Eq. (4.72) */
// calculate spin correction coeffs
spin_corrections_SPA =
    sf2_spin_corrections_amp(m1, m2, S1x, S1y, S1z, S2x, S2y, S2z, lnhatx, lnhaty,
lnhatz);
// calculate PN_coeffs_SPA
sf2_psi_SPA_coeffs_PN_order(PN_coeffs_SPA, phase0, &spin_corrections_SPA, eta);
//costh = cos(dec);
//sinth = sin(dec);
f_cross = 1.0; //0.5*(1+costh*costh)*cos(2*ra)*sin(2*psi) + costh*sin(2*ra)*cos(2*psi);
f_plus = 1.0; //0.5*(1+costh*costh)*cos(2*ra)*cos(2*psi) - costh*sin(2*ra)*sin(2*psi);
printf("====DEBUG f_plus : %e, f_cross : %e, sum : %e====", f_cross, f_plus,

```



```

f_cross+f_plus); // f_cross + f_plus must be one **//
amp_corr_param.cosh = cos(inclination); // cosine of inclination of the source
amp_corr_param.sinh = sin(inclination); // sine of inclination of the source
amp_corr_param.cosh2 = amp_corr_param.cosh*amp_corr_param.cosh;
amp_corr_param.cosh3 = amp_corr_param.cosh2*amp_corr_param.cosh;
amp_corr_param.cosh4 = amp_corr_param.cosh3*amp_corr_param.cosh;
amp_corr_param.cosh5 = amp_corr_param.cosh4*amp_corr_param.cosh;
amp_corr_param.cosh6 = amp_corr_param.cosh5*amp_corr_param.cosh;
amp_corr_param.sinh2 = amp_corr_param.sinh*amp_corr_param.sinh;
amp_corr_param.sinh3 = amp_corr_param.sinh2*amp_corr_param.sinh;
amp_corr_param.sinh4 = amp_corr_param.sinh3*amp_corr_param.sinh;
amp_corr_param.sinh5 = amp_corr_param.sinh4*amp_corr_param.sinh;
amp_corr_param.f_cross = f_cross;
amp_corr_param.f_plus = f_plus;
amp_corr_param.m = m;
amp_corr_param.eta = eta;
amp_corr_param.delta = delta;
amp_corr_param.s1x = S1x;
amp_corr_param.s1y = S1y;
amp_corr_param.s1z = S1z;
amp_corr_param.s2x = S2x;
amp_corr_param.s2y = S2y;
amp_corr_param.s2z = S2z;
amp_corr_param.LNhatx = LNhatx;
amp_corr_param.LNhaty = LNhaty;
amp_corr_param.LNhatz = LNhatz;
amp_corr_param.Xs_x = 0.5*(amp_corr_param.s1x + amp_corr_param.s2x);
amp_corr_param.Xs_y = 0.5*(amp_corr_param.s1y + amp_corr_param.s2y);
amp_corr_param.Xs_z = 0.5*(amp_corr_param.s1z + amp_corr_param.s2z);
amp_corr_param.Xa_x = 0.5*(amp_corr_param.s1x - amp_corr_param.s2x);
amp_corr_param.Xa_y = 0.5*(amp_corr_param.s1y - amp_corr_param.s2y);
amp_corr_param.Xa_z = 0.5*(amp_corr_param.s1z - amp_corr_param.s2z);
amp_corr_param.Xs2
=
amp_corr_param.Xs_x*amp_corr_param.Xs_x+amp_corr_param.Xs_y*amp_corr_param.Xs_y
+amp_corr_param.Xs_z*amp_corr_param.Xs_z; // Xs*Xs (=chi_s dot chi_s)
amp_corr_param.Xa2
=
amp_corr_param.Xa_x*amp_corr_param.Xa_x+amp_corr_param.Xa_y*amp_corr_param.Xa_y
+amp_corr_param.Xa_z*amp_corr_param.Xa_z; // Xa*Xa (=chi_a dot chi_a)
amp_corr_param.XsXa = amp_corr_param.Xs_x*amp_corr_param.Xa_x +
amp_corr_param.Xs_y*amp_corr_param.Xa_y +
amp_corr_param.Xs_z*amp_corr_param.Xa_z; // Xs dot Xa (=xchi_s dot chi_a)
amp_corr_param.XsLn
=
(amp_corr_param.Xs_x*amp_corr_param.LNhatx)+(amp_corr_param.Xs_y*amp_corr_param.
LNhaty)+(amp_corr_param.Xs_z*amp_corr_param.LNhatz); // chi_s dot Ln_hat

```

```

    amp_corr_param.XaLn
    (amp_corr_param.Xa_x*amp_corr_param.LNhatx)+(amp_corr_param.Xa_y*amp_corr_param.
    LNhaty)+(amp_corr_param.Xa_z*amp_corr_param.LNhatz); // chi_a dot Ln_hat

    /* initial checks */
    if (!htilde_out) {
        printf("htilde_out is NULL.\n");
        XLAL_ERROR(XLAL_EFAULT);
    }
    if (*htilde_out) {
        printf("*htilde_out is NOT NULL.\n");
        XLAL_ERROR(XLAL_EFAULT);
    }
    if (m1_SI <= 0) XLAL_ERROR(XLAL_EDOM);
    if (m2_SI <= 0) XLAL_ERROR(XLAL_EDOM);
    if (fStart <= 0) XLAL_ERROR(XLAL_EDOM);
    if (r <= 0) XLAL_ERROR(XLAL_EDOM);

    /* allocate htilde */
    if ( f_max0 == 0. ) // End at ISCO
        //f_max = MAX_HARMONICS*0.5*fISCO; // experiment for upper limit
        f_max = fISCO;
    else // End at user-specified freq.
        f_max = f_max0;
    n = (size_t) (f_max / deltaF + 1);

    if (amplitudeO < 0)
        amplitudeO = MAX_AMP_ORDER;

    XLALGPSAdd(&tC, -1 / deltaF); /* coalesce at t=0 */
    htilde = XLALCreateCOMPLEX16FrequencySeries("hplustilde: FD waveform", &tC, 0.0,
    deltaF, &lalStrainUnit, n);
    if (!htilde) {
        XLAL_ERROR(XLAL_EFUNC);
    }
    else {
    }
    memset(htilde->data->data, 0, n * sizeof(COMPLEX16));
    XLALUnitDivide(&htilde->sampleUnits, &htilde->sampleUnits, &lalSecondUnit);

    /* extrinsic parameters */
    shft = LAL_TWOPI * (tC.gpsSeconds + 1e-9 * tC.gpsNanoSeconds); // 2*Pi*t_c

```

```

iStart = (size_t) ceil(fStart / deltaF);
iEnd = n;
//iEnd = (size_t) (f_max / deltaF);
//iEnd = (iEnd < n) ? iEnd : n; /* overflow protection: should we warn? */

data = htilde->data->data;
for (i = iStart; i < iEnd; i++) {
    f = i * deltaF;
    data[i] = 0.0 + 0.0j;
    for (k = 1; k <= MAX_HARMONICS; k++) // up to 7th harmonics
    {
        phasing = sf2_psi_SPA(f, k, shft, phic, phaseO, PN_coeffs_SPA, m, eta);
        for (n = 0; n <= amplitudeO; n++)
        {
            amp = sf2_amp_SPA_plus(f, n, k, fStart, fISCO, &amp_corr_param);
            data[i] += amp*(cos(k*phasing - LAL_PI_4) - sin(k*phasing - LAL_PI_4)*1.0j);
        }
    }
    data[i] = overall_factor*data[i];
}

*htilde_out = htilde;
return XLAL_SUCCESS;
}
/**
 * Computes the stationary phase approximation to the Fourier transform of
 * a chirp waveform with phase given by Eq.(4.82)
 * and amplitude given by Eq. (4.72) with  $C_k(n)$  given in Appendix D.
 * PN order is the highest order being used by Eq (4.72)
 */
int XLALSimInspiralTaylorF2AmpCross(
    COMPLEX16FrequencySeries **htilde_out, /** frequency-domain waveform */
    REAL8 phic, /** orbital coalescence phase (rad) */
    REAL8 deltaF, /** sampling frequency (Hz) */
    REAL8 inclination, /** inclination of source (rad), corresponds to
theta of  $N_{\hat{a}}$  */
    REAL8 m1_SI, /** mass 1 (kg) */
    REAL8 m2_SI, /** mass 2 (kg) */
    REAL8 S1x, /** initial value of S1x */
    REAL8 S1y, /** initial value of S1y */
    REAL8 S1z, /** initial value of S1z */
    REAL8 S2x, /** initial value of S2x */
    REAL8 S2y, /** initial value of S2y */
    REAL8 S2z, /** initial value of S2z */

```

```

REAL8 lnhatx,          /** initial value of LNhatx */
REAL8 lnhaty,          /** initial value of LNhaty */
REAL8 lnhatz,          /** initial value of LNhatz */
REAL8 fStart,          /** start GW frequency (Hz) */
REAL8 f_max0,          /** ending GW frequency (Hz) */
REAL8 r,               /** distance of source (m) */
int phaseO,            /** twice PN phase order */
int amplitudeO         /** twice PN amplitude order */
)
{
    const REAL8 lambda = -1987./3080.;
    const REAL8 theta = -11831./9240.;

    /* external: SI; internal: solar masses */
    const REAL8 m1 = m1_SI / LAL_MSUN_SI;
    const REAL8 m2 = m2_SI / LAL_MSUN_SI;
    const REAL8 m = m1 + m2;
    const REAL8 m_sec = m * LAL_MTSUN_SI; /* total mass in seconds */
    const REAL8 eta = m1 * m2 / (m * m);
    const REAL8 delta = (m1 - m2)/ m;
    const REAL8 piM = LAL_PI * m_sec;
    const REAL8 vISCO = 1. / sqrt(6.);
    const REAL8 fISCO = vISCO * vISCO * vISCO / piM;
    const REAL8 v0 = cbrt(piM * fStart);
    REAL8 shft, amp0, f_max, f;
    size_t i, n, k, iStart, iEnd;
    int mm;
    COMPLEX16 *data = NULL; /** actual storage for amplitude data pointer */
    COMPLEX16FrequencySeries *htilde; /** waveform data */
    LIGOTimeGPS tC = {0, 0};
    REAL8 overall_factor;
    REAL8 PN_coeffs_SPA[10]; /** PN order coefficients for SPA phase calculation*/
    sf2_spin_corr_amp spin_corrections_SPA: /** spin correction coeffs beta, sigma,
gamma Eqs. (4.82,83,84)*/
    REAL8 f_plus, f_cross, cosh, sinh;
    COMPLEX16 amp;
    sf2_amp_corr_param amp_corr_param;

    REAL8 alpha, alpha_ref, zeta, zeta_ref, beta, phasing;
    COMPLEX16 prec_fac;

    /** calculate frequency independent coefficients */
    overall_factor = m*m/r*LAL_MTSUN_SI*LAL_MRSUN_SI*sqrt(5.0*LAL_PI*eta/48.0); /**
overall factor of Eq. (4.72) */

```

```

// calculate spin correction coeffs
spin_corrections_SPA =
    sf2_spin_corrections_amp(m1, m2, S1x, S1y, S1z, S2x, S2y, S2z, lnhatx, lnhaty,
lnhatz);
// calculate PN_coeffs_SPA
sf2_psi_SPA_coeffs_PN_order(PN_coeffs_SPA, phaseO, &spin_corrections_SPA, eta);
//costh = cos(dec);
//sinth = sin(dec);
f_cross = 1.0; //0.5*(1+costh*costh)*cos(2*ra)*sin(2*psi) + costh*sin(2*ra)*cos(2*psi);
f_plus = 1.0; //0.5*(1+costh*costh)*cos(2*ra)*cos(2*psi) - costh*sin(2*ra)*sin(2*psi);
amp_corr_param.costh = cos(inclination); // cosine of inclination of the source
amp_corr_param.sinth = sin(inclination); // sine of inclination of the source
amp_corr_param.costh2 = amp_corr_param.costh*amp_corr_param.costh;
amp_corr_param.costh3 = amp_corr_param.costh2*amp_corr_param.costh;
amp_corr_param.costh4 = amp_corr_param.costh3*amp_corr_param.costh;
amp_corr_param.costh5 = amp_corr_param.costh4*amp_corr_param.costh;
amp_corr_param.costh6 = amp_corr_param.costh5*amp_corr_param.costh;
amp_corr_param.sinth2 = amp_corr_param.sinth*amp_corr_param.sinth;
amp_corr_param.sinth3 = amp_corr_param.sinth2*amp_corr_param.sinth;
amp_corr_param.sinth4 = amp_corr_param.sinth3*amp_corr_param.sinth;
amp_corr_param.sinth5 = amp_corr_param.sinth4*amp_corr_param.sinth;
amp_corr_param.f_cross = f_cross;
amp_corr_param.f_plus = f_plus;
amp_corr_param.m = m;
amp_corr_param.eta = eta;
amp_corr_param.delta = delta;
amp_corr_param.s1x = S1x;
amp_corr_param.s1y = S1y;
amp_corr_param.s1z = S1z;
amp_corr_param.s2x = S2x;
amp_corr_param.s2y = S2y;
amp_corr_param.s2z = S2z;
amp_corr_param.LNhatx = lnhatx;
amp_corr_param.LNhaty = lnhaty;
amp_corr_param.LNhatz = lnhatz;
amp_corr_param.Xs_x = 0.5*(amp_corr_param.s1x + amp_corr_param.s2x);
amp_corr_param.Xs_y = 0.5*(amp_corr_param.s1y + amp_corr_param.s2y);
amp_corr_param.Xs_z = 0.5*(amp_corr_param.s1z + amp_corr_param.s2z);
amp_corr_param.Xa_x = 0.5*(amp_corr_param.s1x - amp_corr_param.s2x);
amp_corr_param.Xa_y = 0.5*(amp_corr_param.s1y - amp_corr_param.s2y);
amp_corr_param.Xa_z = 0.5*(amp_corr_param.s1z - amp_corr_param.s2z);
amp_corr_param.Xs2
=
amp_corr_param.Xs_x*amp_corr_param.Xs_x+amp_corr_param.Xs_y*amp_corr_param.Xs_y
+amp_corr_param.Xs_z*amp_corr_param.Xs_z; // Xs*Xs (=chi_s dot chi_s)

```

```

    amp_corr_param.Xa2
amp_corr_param.Xa_x*amp_corr_param.Xa_x+amp_corr_param.Xa_y*amp_corr_param.Xa_y
+amp_corr_param.Xa_z*amp_corr_param.Xa_z; // Xa*Xa (=chi_a dot chi_a)
    amp_corr_param.XsXa      =      amp_corr_param.Xs_x*amp_corr_param.Xa_x      +
amp_corr_param.Xs_y*amp_corr_param.Xa_y
amp_corr_param.Xs_z*amp_corr_param.Xa_z; // Xs dot Xa (=xchi_s dot chi_a)
    amp_corr_param.XsLn
(amp_corr_param.Xs_x*amp_corr_param.LNhatx)+(amp_corr_param.Xs_y*amp_corr_param.
LNhaty)+(amp_corr_param.Xs_z*amp_corr_param.LNhatz); // chi_s dot Ln_hat
    amp_corr_param.XaLn
(amp_corr_param.Xa_x*amp_corr_param.LNhatx)+(amp_corr_param.Xa_y*amp_corr_param.
LNhaty)+(amp_corr_param.Xa_z*amp_corr_param.LNhatz); // chi_a dot Ln_hat

/* Perform some initial checks */
if (!htilde_out) {
    XLAL_ERROR(XLAL_EFAULT);
}
if (*htilde_out) {
    XLAL_ERROR(XLAL_EFAULT);
}
if (m1_SI <= 0) XLAL_ERROR(XLAL_EDOM);
if (m2_SI <= 0) XLAL_ERROR(XLAL_EDOM);
if (fStart <= 0) XLAL_ERROR(XLAL_EDOM);
if (r <= 0) XLAL_ERROR(XLAL_EDOM);

/* allocate htilde */
if ( f_max0 == 0. ) // End at ISCO
    //f_max = MAX_HARMONICS*0.5*fISCO; // experment for upper limit for
integration
    f_max = fISCO;
else // End at user-specified freq.
    f_max = f_max0;
n = (size_t)(f_max / deltaF + 1);

if (amplitudeO < 0)
    amplitudeO = MAX_AMP_ORDER;

XLALGPSAdd(&tC, -1 / deltaF); /* coalesce at t=0 */
htilde = XLALCreateCOMPLEX16FrequencySeries("hcsrosstilde: FD waveform", &tC, 0.0,
deltaF, &lalStrainUnit, n);
if (!htilde) {
    XLAL_ERROR(XLAL_EFUNC);
}
else {

```

```

}
memset(htilde->data->data, 0, n * sizeof(COMPLEX16));
XLALUnitDivide(&htilde->sampleUnits, &htilde->sampleUnits, &lalSecondUnit);

/* extrinsic parameters */
shft = LAL_TWOPI * (tC.gpsSeconds + 1e-9 * tC.gpsNanoSeconds); // 2*Pi*t_c

iStart = (size_t) ceil(fStart / deltaF);
iEnd = n;

data = htilde->data->data;
for (i = iStart; i < iEnd; i++) {
    f = i * deltaF;
    data[i] = 0.0 + 0.0j;
    for (k = 1; k <= MAX_HARMONICS; k++) // up to 7th harmonics
    {
        phasing = sf2_psi_SPA(f, k, shft, phic, phaseO, PN_coeffs_SPA, m, eta);
        for (n = 0; n <= amplitudeO; n++)
        {
            amp = sf2_amp_SPA_cross(f, n, k, fStart, fISCO, &amp_corr_param);
            data[i] += amp*(cos(k*phasing - LAL_PI_4) - sin(k*phasing - LAL_PI_4)*1.0j);
// changed sign for time reversal
        }
    }
    data[i] = overall_factor*data[i];
}

*htilde_out = htilde;
return XLAL_SUCCESS;
}

```