

This is the accepted manuscript made available via CHORUS. The article has been published as:

# Towards low-latency real-time detection of gravitational waves from compact binary coalescences in the era of advanced detectors

Jing Luan, Shaun Hooper, Linqing Wen, and Yanbei Chen

Phys. Rev. D **85**, 102002 — Published 14 May 2012

DOI: [10.1103/PhysRevD.85.102002](https://doi.org/10.1103/PhysRevD.85.102002)

# Towards low-latency real-time detection of gravitational waves from compact binary coalescences in the era of advanced detectors

Jing Luan,<sup>1</sup> Shaun Hooper,<sup>2,3</sup> Linqing Wen,<sup>2,3,\*</sup> and Yanbei Chen<sup>1,†</sup>

<sup>1</sup> *Division of Physics, Mathematics, and Astronomy, Caltech, Pasadena, CA 91125, USA*

<sup>2</sup> *Australian International Gravitational Research Centre, School of Physics,  
University of Western Australia, 35 Stirling Hwy, Crawley, WA 6009, Australia*

<sup>3</sup> *International Centre for Radio Astronomy Research, School of Physics,  
University of Western Australia, 35 Stirling Hwy, Crawley, WA 6009, Australia*

(Dated: April 10, 2012)

Electromagnetic (EM) follow-up observations of gravitational wave (GW) events will help shed light on the nature of the sources, and more can be learned if the EM follow-ups can start as soon as the GW event becomes observable. In this paper, we propose a computationally efficient time-domain algorithm capable of detecting inspiral gravitational waves (GWs) from coalescing binaries of compact objects with nearly no further delay in addition to the time required to condition the data into a time series of calibrated gravitational-wave strain. Our algorithm, if can be expanded to include sky localization, will serve as the first step towards triggering EM observation before the merger. The key to the efficiency of our algorithm arises from the use of chains of so-called Infinite Impulse Response (IIR) filters, which filter time-series data recursively. Computational cost is further reduced by a template interpolation technique that requires filtering only done for a “coarse bank”, much sparser than the “fine bank” normally required to sufficiently recover the optimal signal-to-noise ratio: the filter chain of each coarse-bank template is divided into several sections, filtering output from these sections are combined appropriately to reconstruct the output of each of the nearby fine-bank templates. The filter construction and interpolation techniques are illustrated in this paper using Newtonian-chirp waveforms, although these will be generalizable to more accurate post-Newtonian waveforms. Towards future detectors with sensitivity extending to lower frequencies, our algorithm’s computational cost is shown to increase rather insignificantly compared to the conventional time-domain correlation method using Finite Impulse Response (FIR) filters.

PACS numbers: 04.80.Nn, 95.75.-z, 97.80.-d, 97.60.Gb

## I. INTRODUCTION

Coalescences of neutron-star (NS) binaries are primary sources for ground-based gravitational-wave detectors. It has been estimated that Advanced LIGO may be able to detect 10 to 100 such events per year [1, 2]. The mergers of neutron star binaries are also possible progenitors of short hard  $\gamma$ -ray bursts. Although these bursts are believed to be mostly beamed away from us, the prompt emission and afterglow they induce in X-ray, optical, infrared and radio frequency bands may well be less beamed, and therefore be visible to us [3, 4]. If a statistically significant gravitational-wave *trigger* can be obtained before or right after such a coalescence, electromagnetic (especially optical) observatories can then be alerted to search for possible prompt and afterglow emissions — such follow-up observations are likely able to resolve whether these mergers are indeed the progenitors of short hard  $\gamma$ -ray bursts, and provide further knowledge about the nature of these events.

Currently, neutron star - neutron star coalescence signals are being searched for in gravitational-wave data

using the matched filtering technique [5, 6], which calculates the correlation of *data* with *theoretical templates* weighted by noise. In order to reduce the computational cost, current search pipelines use a frequency-domain method, which gathers a long stretch of time-series data containing  $N$  points (the duration of which should be longer than the longest possible signal), then uses a Fast-Fourier-Transform (FFT) algorithm to search for all possible signals that end within this stretch of data, with a cost of  $O(N \log N)$ , as opposed to the  $O(N^2)$  required by a one-by-one search over merger time. Such a trick, although efficient, implies that we cannot start analyzing the data until the collection finishes.

Unless significant changes from current frequency-domain analysis method are made, the latency caused by data collection will compromise our ability to obtain a trigger with the shortest possible delay after the merger, and will totally prevent us from obtaining the trigger before the merger. At least two efforts are underway to suppress latencies for coalescence signals, the Multi-Band Template Analysis (MBTA) [7] and the Low-Latency Online Inspiral Detector (LLOID) [8]. MBTA is a two-band frequency-domain search method while LLOID provides an infrastructure that accommodates either time or frequency domain searches. The time-domain aspect of the LLOID pipeline based on Finite-Impulse-Response (FIR) filters [9] is described in [8]. Note for a different search

---

\*Electronic address: [linqing.wen@uwa.edu.au](mailto:linqing.wen@uwa.edu.au)

†Electronic address: [yanbei@tapir.caltech.edu](mailto:yanbei@tapir.caltech.edu)

Detector	$f_{\min}$ (Hz)	duration (s)	$N_{\text{cyc}}$	$\mathcal{N}_{0.98}$
iLIGO	40	25	$1.6 \times 10^3$	$1.7 \times 10^3$
aLIGO	10	$1.0 \times 10^3$	$1.6 \times 10^4$	$6.6 \times 10^3$
ET	3	$2.5 \times 10^4$	$1.2 \times 10^5$	$2.9 \times 10^4$

TABLE I: Basic information for the detection of Newtonian chirp signals by initial, Advanced LIGO and Einstein Telescope. The columns, from left to right list the names for present and future detectors, the minimum frequency of the detector, signal duration and number of wave cycles for a  $(1.4+1.4)M_{\odot}$  NS-NS binary [see Sec. III A], as well as the number of templates required in order to achieve a match of 0.98 for binaries with individual mass of  $1 - 3 M_{\odot}$  [computed from the metric Eq. (60)].

of short gravitational waves of unknown waveforms, a program has been set up to analyze available detector data in near real-time and seek for optical counterpart of candidate events [10].

In this paper, we propose a straightforward and efficient time-domain search algorithm, which allows zero and even negative latency (i.e., obtaining trigger before the merger if the signal-to-noise ratio (SNR) condition and other consistency conditions are met) in the most natural way. Admittedly, without the savings made available by FFT, the computational cost of a straightforward implementation using FIR filters can be formidable. In the correlation calculation, each template contains a large number of wave cycles, and there exists a large number of templates — and both these numbers increase dramatically with the lowering of the minimum frequency cutoff  $f_{\min}$  (Table I). This poses serious computational challenge for detecting GWs from compact object coalescence for future GW detectors.

We propose two techniques that can dramatically increase the computational efficiency for time-domain searches of the inspiral part of GWs from coalescing binaries of compact objects in real-time, and make it feasible for future detectors with frequency cut-offs at as low as  $f_{\min} = 3$  Hz. The first technique uses the well-known Infinite Impulse Response, or IIR filters [9], which can be computed with much higher efficiency than FIR filters. We propose to filter the data using a bank of IIR filters, the sum of which approximates each individual binary coalescence waveform template. The second technique reduces the number of templates by an interpolation technique that applies to the proposed IIR filter method. In this approach, we first divide the bank of IIR filters associated with each template into sub-groups, and then reconstruct the filter outputs of a fine template bank by recombining the filter outputs from each of these sub-groups with appropriate complex coefficients and time delays. This is similar to the generic multi-band interpolation scheme used in MBTA and LLOID [7, 11, 12].

Several conventions are used in this paper. The term *latency* refers generally to the delay from the time when a signal arrives at the detector to the time the data contain-

ing the signal actually starts to be analyzed. We specifically focus on the delay starting from the time when the data are ready to be analyzed. One example of the latency is the delay due to data accumulation before a Fast Fourier Transformation (FFT) can be performed. The term *real time* processing means that data points or data segments are processed (with outputs generated) at a rate that is equal to their input rate. Floating Point Operation is abbreviated as FLOP (plural FLOPs). FLOPs and *flops* are used interchangeably to stand for Floating Point Operations per Second. Throughout this paper, we follow the convention of counting each real addition and real multiplication equally as one FLOP.

This paper is structured as follows. In Section II, we briefly review the basics of matched filtering technique and introduce time-domain IIR filters. In Section III, we use Newtonian-order templates as an example to construct IIR filters, characterize the error involved and calculate the computational cost for each individual template. In Section IV, we present an interpolation technique that allows us to use a significantly decreased number of templates for which filter chains must be implemented. In Section V, we make a simple comparison between the computational cost of IIR filtering and the straightforward frequency-domain algorithm. In Section VI, we summarize our main conclusions.

## II. MATCHED FILTERING TECHNIQUE

The optimal technique to extract a signal from noisy data when we have reliable theoretical predictions for the signal *waveform* is to use *matched filtering* [5, 6]. The output of the matched filtering technique is basically the correlation of data with expected waveforms weighted by noise. This can be realized in the frequency or time domain. We will give a brief overview of the matched filtering technique, and introduce its frequency-domain implementation and its time-domain approach using the FIR and IIR filters.

### A. Frequency-domain implementation

#### 1. Single Template

Suppose the output of the interferometer  $h$  is a sum of noise  $n$  and, if exists, a signal  $s$ :

$$h = n + s \quad (1)$$

For the moment, let us assume that  $s$  is a single known *waveform*. In Eq. (1), we have intentionally left out the arguments of the functions  $h$ ,  $n$ , and  $s$ , which reflects the point of view that each of them can be equivalently *represented* both in the time and frequency domain. More specifically, we use the following convention for Fourier transform, which relates  $h(t)$  and  $\tilde{h}(f)$  (we shall use tilde

to emphasize a frequency-domain representation):

$$\tilde{h}(f) \equiv \int_{-\infty}^{\infty} dt e^{i2\pi ft} h(t). \quad (2)$$

The power spectral density of  $n(t)$  is denoted by  $S_h(f)$ , which is defined by

$$E[\tilde{n}(f)\tilde{n}^*(f')] = \frac{1}{2}\delta(f-f')S_h(f). \quad (3)$$

Here we use one-sided spectral density,  $E[\cdot]$  denotes the expectation value over an ensemble of realizations of the noise and “ $*$ ” denotes complex conjugation.  $S_h(f) = S_h(|f|)$  as the noise in the time domain  $n(t)$  is real.

In order to extract  $s$  from  $h$ , we perform *filtering*, which consists of taking the *inner product* between data  $h$  and *template*  $u$ , forming a *filter output* of  $y$ :

$$y = \langle h|u \rangle = \langle s|u \rangle + \langle n|u \rangle \quad (4)$$

Here we define inner product as

$$\langle a|b \rangle \equiv 2 \int_{-\infty}^{+\infty} df \frac{\tilde{a}^*(f)\tilde{b}(f)}{S_h(f)} \quad (5)$$

If both  $a(t)$  and  $b(t)$  are real valued, this can be further simplified into

$$\langle a|b \rangle \equiv 4 \operatorname{Re} \int_0^{+\infty} df \frac{\tilde{a}^*(f)\tilde{b}(f)}{S_h(f)}, \quad a(t), b(t) \in \mathbf{R}. \quad (6)$$

In  $y$ , we have a signal component  $\langle s|u \rangle$  and a noise component  $\langle n|u \rangle$  which fluctuates around zero. If  $s$  has a substantially high amplitude and if the template  $u$  is appropriate, the signal component  $\langle s|u \rangle$  in  $y$  will raise to a high value that merely random fluctuation of  $\langle n|u \rangle$  is very unlikely to account for. As a consequence, we can impose a threshold on  $y$  — an incidence with  $y$  higher than the threshold is viewed as a detection of a signal. The detection efficiency depends on the signal-to-noise ratio (SNR) defined generally as

$$\rho = \frac{y(n=0) - E[y(s=0)]}{\sigma_{y(s=0)}}, \quad (7)$$

where  $\sigma_{y(s=0)}$  is the standard deviation of the filter output when data contain noise only. Assuming zero-mean Gaussian noise, we have for Eq. (4)

$$\rho \equiv \frac{\langle s|u \rangle}{\sqrt{E[|\langle n|u \rangle|^2]}}, \quad (8)$$

Note that the SNR does not depend on the normalization of the template  $u$ , and it is conventional to require that  $\langle u|u \rangle = 1$ . In this case, the cross-correlation of a template with pure noise  $\langle n|u \rangle$  is a random variable with zero mean and unity variance. It can be shown that  $E[\langle a|n \rangle \langle n|b \rangle] = \langle a|b \rangle$ . So we have

$$\rho = \langle s|u \rangle. \quad (9)$$

According to the Cauchy-Schwarz inequality,

$$\rho = \frac{\langle s|u \rangle}{\sqrt{\langle s|s \rangle} \sqrt{\langle u|u \rangle}} \leq \sqrt{\langle s|s \rangle}. \quad (10)$$

where equal sign takes place when  $u = \lambda s$  where  $\lambda$  is a constant, and normalization of  $u$  gives  $\lambda = 1/\sqrt{\langle s|s \rangle}$ . This means the *optimal* SNR is given by the modulus of the signal,  $\langle s|s \rangle$ , and the *reduction* of SNR due to imperfectness of template is given by the *match*, which is also equal to unity minus mismatch,  $\varepsilon$ :

$$\frac{\langle s|u \rangle}{\sqrt{\langle u|u \rangle} \sqrt{\langle s|s \rangle}} \equiv 1 - \varepsilon. \quad (11)$$

## 2. Intrinsic and Extrinsic Parameters

In reality, templates are not necessarily placed along each parameter dimension. The maximization of SNR over certain parameters can be conducted analytically and therefore no templates are needed. These parameters are called *extrinsic parameters*, while those that still have to be searched over one by one are called *intrinsic parameters*.

For example, for any template

$$\tilde{u}(f) = \tilde{u}_0(f) e^{2\pi i f t_c + i \phi_c}, \quad f > 0 \quad (12)$$

which is constructed from  $u_0(t)$  by a time shift  $t_c$  and a phase shift  $\phi_c$ , the SNR, for a target  $h$  is given by

$$\rho = 4 \operatorname{Re} \int_0^{+\infty} df \frac{\tilde{h}^*(f)\tilde{u}_0(f)}{S_h(f)} e^{i2\pi f t_c + i \phi_c}. \quad (13)$$

Optimization over  $\phi_c$  can be carried out instantly without having to search over all possible values of  $\phi_c$ , giving

$$\max_{\phi_c} \rho = 4 \left| \int_0^{+\infty} df \frac{\tilde{h}^*(f)\tilde{u}_0(f)}{S_h(f)} e^{i2\pi f t_c} \right|. \quad (14)$$

This makes  $\phi_c$  an extrinsic parameter. Further optimization over all  $t_c$  can be done by first computing all values of  $\max_{\phi_c} \rho$  using a Fast Fourier Transform, which cost  $O(N \log N)$  FLOPs in the discretized case where  $N$  is the number of data points in the time domain. This is much faster than computing the correlation for all possible ending times, one by one, which cost  $O(N^2)$  operation counts — and in this way ending time  $t_c$  is converted into an extrinsic parameters. The method of Fourier transformation will be discussed in detail in subsection V B.

## B. Time-domain Approach: FIR and IIR method

For the time-domain filtering we need to obtain a time series of SNRs as a function of presumed signal arrival

time  $t$

$$\begin{aligned}\rho(t) &= 2 \int_{-\infty}^{\infty} \frac{\tilde{h}^*(f)\tilde{u}(f)}{S_h(f)} e^{+i2\pi ft} df \\ &= \int_{-\infty}^{+\infty} w(t')u(t'-t)dt'\end{aligned}\quad (15)$$

with

$$w(t) \equiv 2 \int_{-\infty}^{\infty} df \frac{\tilde{h}(f)}{S_h(f)} e^{-i2\pi ft}, \quad (16)$$

which can be thought of as “over-whitened data”; it is a real-valued function of time. Note that in order to generate the over-whitened data, we need to convolve  $h(t)$  with the Inverse Fourier Transform of  $1/S_h(f)$ , which is a time-symmetric, oscillatory function that decays towards zero when  $t$  is much larger than the inverse of the interferometer’s bandwidth ( $\gtrsim 100$  Hz), which is about  $\lesssim 10$  ms. This means the over-whitening process has an inherent latency not much larger than 10 ms, which is negligible compared to the duration of the signal. Note that we have intentionally not used the real-valued definition (6) for inner product, because we would like to keep the possibility of using complex-valued templates.

We now discretize the filtering algorithm. The discrete form of Eq (15) becomes,

$$\rho_k = \sum_{j=-\infty}^k w_j u_{j-k} \Delta t, \quad (17)$$

Here we assume  $t_k = k\Delta t$ , and that  $u$  only have support within  $t \leq 0$ . While in principle the waveform  $u_k$  could have an infinite support in time,  $-\infty < k\Delta t < 0$ . However, the waveform  $u(t)$  is always assumed to begin only after its amplitude reaches sensitivity within the LIGO band. Hence we instead define the waveform to exist on the domain  $-N\Delta t \leq t \leq 0$ , and Eq (17) becomes,

$$\rho_k = \sum_{j=k-N}^k w_j u_{j-k} \Delta t, \quad (18)$$

This summation of the product of data and template at each step turns out to be the general form of Finite Impulse Response (FIR) filters. The term *finite* comes from the fact that the output  $\rho_k$  of the filter (its response) will become exactly zero after  $N$  time steps have passed since a single initial impulse in the data. For example, if we assume  $w_0 = 1$ ,  $w_k = 0$  for  $k \neq 0$ , then  $\rho_k$  will vanish for  $k > N$ . As seen from Eq. (18), each  $\rho_k$  costs  $N$  multiplications and  $N$  additions to calculate. This translates into a computational cost, in terms of FLOPs per unit time, of  $\sim N/\Delta t$ .

For certain types of waveforms, Infinite Impulse Response (IIR) filters can be used to dramatically reduce computational cost. The simplest IIR filter is a first-order recursive algorithm, in which the  $k$ -th output  $y_k$  is

a linear combination of the  $(k-1)$ -th output,  $y_{k-1}$  and the  $k$ -th data,  $w_k$ :

$$y_k = e^{-(\gamma-i\Omega)\Delta t} y_{k-1} + w_k \Delta t, \quad (19)$$

where  $\gamma, \Omega$  are real-valued constants with  $\gamma > 0$  to ensure stable solutions. It can be shown that, e.g., by using tools of Z-transform [13], as long as  $w_k$  does not diverge towards  $k \rightarrow -\infty$ , then even if the recursion starts at a finite time step, after an initial transient of several times  $1/\gamma$ , the output of the filter achieves a steady state of

$$y_k = \sum_{j=-\infty}^k w_j e^{(\gamma-i\Omega)(j-k)\Delta t} \Delta t. \quad (20)$$

Note this is the discretized version of the continuous integration

$$y(t) = \int_{-\infty}^{\infty} w(t') e^{(\gamma-i\Omega)(t'-t)} \Theta(t-t') dt'. \quad (21)$$

Note first that Eq. (20) indeed gives an infinite impulse response, because for a data series containing only one impulse,  $w_0 = 1$  and  $w_k = 0$  ( $k \neq 0$ ), the output of the filter, even at very late time steps, never vanishes. More over, by comparing this with Eq. (15), the IIR filter can be viewed as a template of a “damped” (looking back in time) sinusoid:

$$u(t) = e^{(\gamma+i\Omega)t} \Theta(-t), \quad (22)$$

where  $\Theta(t)$  is the Heaviside function

$$\Theta(t) = \begin{cases} 0, & t < 0; \\ 1, & t \geq 0. \end{cases} \quad (23)$$

The IIR filter described above requires only one complex multiplication and one summation per sampling time, which means the computational cost is  $\sim 1/\Delta t$ .

For a simple proof of concept on the computational efficiency of the IIR over the FIR filtering technique, we examine the case when we *do* need to filter for a damped sinusoid signal with frequency  $\Omega$  and decay rate  $\gamma$ . The data to be filtered has a duration of at least on the order of  $1/\gamma$ . The Nyquist sampling theorem limits the sampling interval to be at most  $\sim 1/\Omega$ , meaning that the FIR template would have the number of data points several times larger than  $\Omega/\gamma$ . Subsequently the computational cost of the FIR in FLOPs per unit time is larger than  $\Omega^2/\gamma$ . An IIR filter, on the other hand, only has a cost of  $\Omega$ , which means the cost of IIR filter is  $\gamma/\Omega \sim 1/Q$  times that of FIR filter, where  $Q \equiv \Omega/\gamma$  is the quality factor of the damped sinusoid. As a consequence, if we can convert our waveforms into a sum of a series of high- $Q$  damped sinusoids, IIR filters can be used over the FIR to dramatically reduce the computational cost.

### III. CONSTRUCTION OF IIR FILTERS FOR AN INDIVIDUAL INSPIRAL WAVEFORM

The simple IIR filter discussed in the previous section has the special waveform of a damped sinusoid [Cf. Eq. (22)]. In this section, we will show that a *chain* of IIR filters can be used to “piece together” the waveforms of compact binary coalescence. This is possible because these waveforms are basically sinusoids with slowly varying amplitude and frequency. For simplicity, in this paper, we will restrict ourselves to Newtonian Chirps.

#### A. The Newtonian Chirp Waveform

The Newtonian-chirp is the leading-order waveform from a compact coalescing binary. In the time domain it can be written as the real part of the complex expression (see, e.g., [6], Sec. C),

$$u(t) \propto (t_c - t)^{-1/4} e^{-i2(5M_c)^{-5/8}(t_c - t)^{5/8} + i\phi_c} \equiv \mathcal{A}(t)e^{i\Phi(t)} \quad (24)$$

where we follow the convention of the geometrized units with gravitational constant  $G = 1$  and the speed of light  $c = 1$ ,  $M_c$  is the *chirp mass* of the binary,

$$M_c = M\eta^{3/5} \quad (25)$$

which depends on the total mass of the binary  $M$  and  $\eta \equiv m_1 m_2 / M^2$ , the symmetric mass ratio. The signal finishes at the ending time  $t_c$ , and  $\phi_c$  is the constant phase at the end time. Here we have ignored time-independent factors of proportionality in the amplitude, which do not affect template construction. Note that because we are modeling the inspiral part of the wave, the ending time  $t_c$  here is intended to model the end of the inspiral stage, not the end of the entire coalescence waveform.

We have assigned real-valued functions  $\mathcal{A}(t)$  and  $\Phi(t)$  to denote the amplitude and phase of the waveform. Although the actual waveform is the real part of  $u(t)$ , we have intentionally kept its complex form, because the imaginary part of  $u(t)$  represents the waveform of a binary with a phase shift of  $\pi/2$  from the real part—therefore the real and imaginary parts together form a *basis* for the *linear space of signals of all phases*. This is a feature of all *adiabatic waveforms*, which satisfy

$$\dot{\mathcal{A}}/(\Omega\mathcal{A}) \ll 1, \quad \dot{\Omega}/\Omega^2 \ll 1. \quad (26)$$

Here dot represents time derivative, and  $\Omega \equiv \dot{\Phi}$  is the instantaneous angular frequency. In other words, the amplitude  $\mathcal{A}(t)$  and angular frequency  $\dot{\Phi}(t)$  both evolve at rates much slower than the instantaneous frequency  $\dot{\Phi}$ . This allows us to use the Stationary Phase Approximation (SPA) to compute the Fourier Transform of the waveform in Eq. (24),

$$\tilde{u}(f) \propto f^{-7/6} e^{i(Af^{-5/3} + 2\pi f t_c + \phi_c - \pi/4)}, \quad f > 0, \quad (27)$$

where

$$A = \frac{3}{4}(8\pi M_c)^{-5/3} \quad (28)$$

is the intrinsic parameter we need to search for in the case of Newtonian chirp. Note that when we Fourier-transform the complex signal of Eq. (24), there is only positive-frequency component, with  $\tilde{u}(f) = 0$  for  $f < 0$ . On the other hand, if we took the real part of the signal, we would have  $\tilde{u}(-f) = \tilde{u}^*(f)$  for  $f > 0$ .

The duration of the inspiral wave can be well approximated as a function of chirp mass  $M_c$  [See Eq. (25)] and the detector’s minimum cut-off frequency  $f_{\min}$  [33],

$$T(M_c, f_{\min}) = \frac{647013}{(f_{\min}/\text{Hz})^{8/3}(M_c/M_\odot)^{5/3}} \text{ s}. \quad (29)$$

One can see that for a fixed  $f_{\min}$ , the longest signal duration corresponds to the smallest chirp mass. The sample signal durations for the initial, advanced and future GW detectors of various  $f_{\min}$  can be found in Table I, column 3. It is shown that GWs from a canonical  $(1.4+1.4)M_\odot$  NS-NS binary system will have a duration 40 times longer for advanced detectors, and possibly 10000 times longer for the future ET detector than that of the initial detector.

#### B. An IIR filter chain

The adiabatic condition in Eq. (26) also implies that the waveform can be divided into *constant-frequency intervals*: within each interval it can be approximated as a sinusoid with constant frequency, while neighboring intervals have slightly different frequencies. This further indicates that we can attempt to write the entire waveform into the sum of a series of damped sinusoids: the frequency of each sinusoid corresponds to a constant-frequency interval, the ending time of the sinusoid corresponds to the ending time of this constant-frequency interval, while the decay time should be comparable to the length of the constant-frequency interval. The amplitude of the damped sinusoid can be set to be comparable to the amplitude of the original waveform during the corresponding constant-frequency interval.

Mathematically, our target is therefore to approximate the signal template  $u(t)$  with the sum of a chain of IIR filters [Cf. Eq. (22)], which we denote by  $U(t)$ :

$$U(t) \equiv \sum_{l=1}^M B_l e^{(\gamma_l - i\Omega_l)(t - t_l)} \Theta(t_l - t). \quad (30)$$

Here the chain consists of  $M$  filters; for filter  $l$  ( $1 \leq l \leq M$ ),  $B_l$  is the amplitude of the filter  $l$ ,  $\Omega_l$  and  $\gamma_l$  are the angular frequency and decay rate, and  $t_l$  is its ending time.

As a first step, let us determine the relevant portion of the signal that we need to approximate: this is bounded

by the low frequency cut-off  $f_{\min}$ , below which the chirp only contributes negligible signal-to-noise ratio, as well as the high frequency cut-off  $f_{\max}$ . The minimum frequency  $f_{\min}$  is normally determined by the seismic wall of the detector which is set to be 40 Hz for initial LIGO, 10 Hz for Advanced LIGO, and might extend to lower frequencies in future detectors, such as the Einstein Telescope (ET). The maximum frequency  $f_{\max}$  is either determined by the end of the Newtonian chirp or the upper end of the detection band. In this paper we set  $f_{\max} = 2000$  Hz.

Now suppose our Newtonian chirp has a particular value for the intrinsic parameter  $A$ , and  $t_c = 0$ ,  $\phi_c = 0$ . Let us define  $t_0 \equiv t_{\text{ini}}$  as the time at which the instantaneous frequency of the waveform is equal to  $f_{\min}$  (which means  $|t_0| = -t_0$  is the duration of the Newtonian chirp from  $f_{\text{ini}}$  to coalescence), and incrementally define

$$t_l = t_{l-1} + T_l, \quad \left| \frac{1}{2} \ddot{\Phi}(t_l) T_l^2 \right| = \epsilon \ll 1, \quad l = 1, 2, \dots \quad (31)$$

until we reach  $t_M$ , which corresponds to a frequency at or beyond  $f_{\max}$ . These intervals,

$$[t_0, t_1], [t_1, t_2], \dots, [t_{M-1}, t_M] \quad (32)$$

will be the constant-frequency intervals described previously. The parameter  $\epsilon$  should be substantially less than unity, so that the phase error caused by assuming a constant frequency is significantly less than one radian.

For  $t \in [t_{l-1}, t_l]$ , we expand  $\Phi(t)$  at  $t_* = t_l - \alpha T_l$  (where  $\alpha$  is an ad hoc parameter to be adjusted later)

$$\Phi(t) \simeq \Phi(t_l^*) + \dot{\Phi}(t_l^*)(t - t_l^*) + \frac{1}{2} \ddot{\Phi}(t_l^*)(t - t_l^*)^2 \quad (33)$$

such that the first term is a constant phase, the second term gives a single angular frequency of  $\dot{\Phi}(t_l^*)$ , while the third term gives the error of a single-frequency approximation, which will be small if  $\epsilon$  is small enough in Eq. (31). We will then use  $\Omega_l \equiv -\dot{\Phi}(t_l^*)$  as the oscillation frequency of the IIR filter assigned for this constant-frequency interval, and prescribe a complex amplitude of

$$B_l \equiv \mathcal{A}(t_l^*) e^{i\Phi(t_l^*) - i\Omega_l(t_l - t_l^*)}. \quad (34)$$

These will assemble into

$$\begin{aligned} B_l e^{-i\Omega_l(t-t_l)} &= \mathcal{A}(t_l^*) e^{i\Phi(t_l^*) + i\dot{\Phi}(t_l^*)(t-t_l^*)} \\ &\approx \mathcal{A}(t) e^{i\Phi(t)}, \quad t_{l-1} \leq t \leq t_l. \end{aligned} \quad (35)$$

We must still add a Heaviside function and a damping component to modify (35) into a form realizable by an IIR filter. Since the validity of (35) is between  $t_{l-1}$  and  $t_l$ , it is natural to have the Heaviside function cut off values for  $t > t_l$ , and to have the damping component have a time constant comparable to  $T_l$ , which gradually cuts off the filter at  $t \lesssim t_{l-1}$ . Prescribing

$$\gamma_l = \zeta/T_l, \quad (36)$$

with  $\zeta$  yet another ad hoc parameter, we write

$$\begin{aligned} U_l(t; A, t_c = 0, \phi_c = 0) \\ \equiv B_l e^{-i\Omega_l(t-t_l) - \gamma_l(t_l-t)} \Theta(t_l - t) \end{aligned} \quad (37)$$

which is our IIR filter for interval  $l$ , for chirps with parameters  $A$ ,  $t_c = 0$ ,  $\phi_c = 0$ . Summing over all  $U_l$ , we obtain an IIR chain that approximates the entire *complex* chirp signal:

$$U(t; A, t_c = 0, \phi_c = 0) = \sum_{l=1}^M U_l(t; A, t_c = 0, \phi_c = 0). \quad (38)$$

If the sum of the complex filter chain  $U(t)$  indeed approximates the complex chirp signal  $u(t)$  [Cf. Eq. (24)], then the real and imaginary parts of the output from the filter chain will be good approximations for filtering chirps with  $\phi_c = 0$  and  $\pi/2$ , respectively.

For non-zero  $t_c$ , we will have to apply

$$U(t; A, t_c, \phi_c = 0) \equiv U(t - t_c; A, t_c = 0, \phi_c = 0) \quad (39)$$

Note that having Heaviside Function  $\Theta(t - t_c - t_l)$  within  $U_l$  means we have to collect the IIR filter result of filter  $l$  at  $t_l + t_c$ . The fact that all  $t_l$  are negative means all results are obtained *before* the coalescence (which happens at  $t_c$ ) and hence IIR filtering itself causes no latency — except for the small latency due to over-whitening, as stated previously (sec. II B)

### C. Filtering for general signal phases and goodness of match

Since the construction of the IIR filter chain is of an ad hoc nature, we must test how well the resulting IIR filter chain  $U$  can approximate the original signal  $u$ . A natural candidate would be imposing that the match between the signal  $u$  and the template  $U$

$$\rho_{\text{cplx}} = \frac{|\langle u|U \rangle|}{\sqrt{\langle u|u \rangle \langle U|U \rangle}} \quad (40)$$

must be close to unity.

However, this needs to be connected to the signal-to-noise ratio achievable by IIR filtering. For doing so, we must first elaborate how to use the output of the complex IIR filtering to recover signals with arbitrary phases. If we write

$$u \equiv u_r + iu_i \quad (41)$$

with  $u_{r,i}$  represent the real and imaginary parts of  $u$  in the time domain, and similarly,

$$U \equiv U_r + iU_i, \quad (42)$$

then the true signal of arbitrary phase is a linear combination of  $u_r$  and  $u_i$  written as  $A_1 u_r + A_2 u_i$ , and we should

use a linear combination of the real and imaginary parts of  $U$ , namely  $B_1 U_r + B_2 U_i$  as the search template. For any particular coefficients  $A_{1,2}$ , the optimal overlap is given by

$$\rho_{\text{IIR}}(A_1, A_2) = \max_{B_{1,2}} \frac{\langle A_1 u_r + A_2 u_i | B_1 U_r + B_2 U_i \rangle}{\sqrt{\langle B_1 U_r + B_2 U_i | B_1 U_r + B_2 U_i \rangle}} \quad (43)$$

The worst-case scenario is given by a minimization over  $(A_1, A_2)$ :

$$\rho_{\text{IIR}}^{\text{worst}} = \min_{A_1, A_2} \frac{\rho_{\text{IIR}}(A_1, A_2)}{\sqrt{\langle A_1 u_r + A_2 u_i | A_1 u_r + A_2 u_i \rangle}}. \quad (44)$$

In fact, when the signal and the template are both highly adiabatic, it can be shown that  $\rho_{\text{IIR}}(A_1, A_2)$  is approximately independent of  $A_{1,2}$ , and that to a very good accuracy:

$$\rho_{\text{cplx}} \approx \rho_{\text{IIR}}^{\text{worst}}. \quad (45)$$

Eq. (44) is therefore used to calculate the goodness of the match of the IIR filter chain.

#### D. Implementation for $(1.4 + 1.4)M_\odot$ binaries and initial LIGO

We first apply the prescription described in Sec. III B to construct an IIR filter chain for  $(1.4 + 1.4)M_\odot$  binaries for initial LIGO and use Eq. (44) to test their overlap with the true signals. We choose (by hand)  $\alpha = 2.3$ ,  $\epsilon = 0.269$  and  $\zeta = 4$ , an overlap of 0.99 is achieved with  $N_{\text{IIR}} = 200$  IIR filters.

We next estimate the computational cost required by such IIR filtering. We focus on the floating point operation count per unit time required to generate complex outputs from the sum of individual IIR filter outputs of Eq. (19). Here we assume the maximum sample rate for compact-binary coalescence data analysis is 8192 Hz, with  $2\times$  down-sampling applied successively to provide channels with sample rates of 4096 Hz, 2048 Hz, ..., 256 Hz. The IIR filter bank is divided into 6 groups, each corresponding to a frequency band of  $2^{k+5} - 2^{k+6}$  Hz, for  $k = 0, 1, \dots, 5$ . For filters in group  $k$ , we assume they are applied to the channel with sample rate of

$$S_k = 2^{k+8} \text{ Hz}. \quad (46)$$

In Table II, we list the actual number of IIR filters required to achieve a minimum overlap of 0.99 at different frequency band with downsampling technique. For comparison, we list the corresponding numbers for the FIR method also applied with downsampling technique.

At each time step, each IIR filter needs to perform a total of 12 real-number multiplications and additions namely: 4 real-number multiplications plus 2 real-number additions for multiplying the current output by the complex recursive coefficient, 2 real-number multiplications for multiplying data (second term in Eq. (19))

with a complex normalization coefficient to yield proper SNR output, 2 real-number additions for combining the previous two products, while finally 2 real-number additions for adding the result of this filter into the total output.

If we ignore costs for down- and up-sampling, which are performed relatively rarely, the total computational cost for initial-LIGO filters in Table II is

$$C_{\text{IIR}} = \sum_{k=0}^5 12 S_k N_{\text{IIR},k} \simeq 2.4 \times 10^6 \text{ flops}. \quad (47)$$

On the other hand, if we carry out the same down sampling scheme for FIR filtering, the number of points in group 0 will be

$$N_{\text{FIR},0} = S_0 \cdot [t(64 \text{ Hz}) - t_{\text{ini}}] \quad (48)$$

where  $t(64 \text{ Hz})$  is the time at which the instantaneous frequency is 64 Hz. For  $k = 1, 2, 3, \dots, 5$ , we have

$$N_{\text{FIR},k} = S_k \cdot [t(2^{k+6} \text{ Hz}) - t(2^{k+5} \text{ Hz})] \quad (49)$$

At sample rate  $S_k$ , for each time step, we have to perform two real-valued correlations with array length  $N_{\text{FIR},k}$ , which cost  $4N_{\text{FIR},k}$  floating point operations. The total computational cost of FIR filtering is therefore

$$C_{\text{FIR}} = \sum_{k=0}^5 4 S_k N_{\text{FIR},k} \simeq 2.0 \times 10^7 \text{ flops}. \quad (50)$$

This is nearly 8 times the cost of the IIR filter method assuming downsampling technique applied to both filtering methods. The result of above cost estimation for the IIR and FIR filtering are also listed in Table II. We will show in the next subsections that the improvement is much more significant for advanced detectors as they venture into lower frequencies.

#### E. Dependence on initial frequency and future detectors

As initial frequency  $f_{\text{min}}$  is lowered in future gravitational-wave detectors, we anticipate much longer signals (see Table I), and therefore a possibly dramatic increase of computational cost. In this subsection, we will first obtain analytical scalings in IIR and FIR computational costs, assuming an idealized down-sampling scheme. We will then provide more realistic estimates of cost by constructing actual IIR filters and adopting the same successive  $2\times$  down-sampling strategy.

##### 1. Analytical Estimates

Ideally, the minimum sample rate is twice the instantaneous frequency of the signal, or  $S = 2f$ . For FIR filters,



	Type	Rate	$S_k$ ( $s^{-1}$ )	16	32	64	128	256	512	1024	2048	4096	8192	$N_{\text{tot}}$	Total Cost
			$f/\text{Hz}$	2 – 4	4 – 8	8 – 16	16 – 32	32 – 64	64 – 128	128 – 256	256 – 512	512 – 1024	>1024		
iLIGO	FIR	$N_{\text{FIR},k}$						4547	3062	965	304	96	30	9004	20
		$\mathcal{C}_{\text{FIR},k}$						4.7	6.3	4.0	2.5	1.6	1.0		
	IIR	$N_{\text{IIR},k}$						71	62	34	19	10	4	200	2.4
		$\mathcal{C}_{\text{IIR}}$						0.22	0.38	0.42	0.47	0.49	0.39		
aLIGO	FIR	$N_{\text{FIR},k}$				45835	30868	9723	3062	965	304	96	30	90883	53
		$\mathcal{C}_{\text{FIR},k}$				11.7	15.8	10.0	6.3	4.0	2.5	1.6	1.0		
	IIR	$N_{\text{IIR},k}$				220	198	111	62	34	19	10	4	658	3.0
		$\mathcal{C}_{\text{IIR}}$				0.17	0.30	0.34	0.38	0.42	0.47	0.49	0.39		
$\text{ET}_B$	FIR	$N_{\text{FIR},k}$	213010	311130	98000	30868	9723	3062	965	304	96	40		667198	120
		$\mathcal{C}_{\text{FIR},k}$	13.6	39.8	25.1	15.8	10.0	6.3	4.0	2.5	1.6	1.0			
	IIR	$N_{\text{IIR},k}$	392	631	353	198	111	62	34	19	11	3		1814	3.3
		$\mathcal{C}_{\text{IIR}}$	0.08	0.24	0.27	0.30	0.34	0.38	0.42	0.47	0.54	0.29			

TABLE II: Break-down of number of filters and computational cost (over successive two-fold down-sampling channels) of multi-rate FIR and IIR filtering, of a single template for a  $(1.4 + 1.4) M_\odot$  binary for initial, Advanced LIGO and the Einstein Telescope. See text in Sec. III D. Here computational costs for each type of filtering and for different sampling channels are calculated using Eqs. (47)–(50), with numerical values quoted in units of MFLOPS or  $10^6$  FLOPS. The minimum overlap is 0.99.

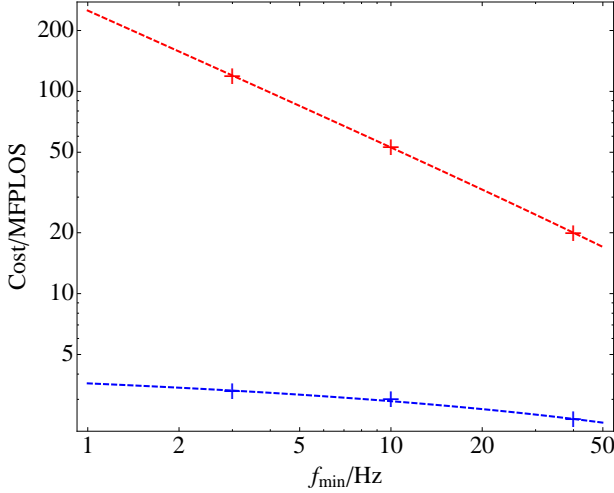


FIG. 1: Theoretical (dashed curves) and numerical (labeled by “+”s) scaling of the computational cost with  $f_{\text{min}}$  for the FIR (red color) and IIR (blue color) method for one template, fixing  $f_{\text{max}} = 2000$  Hz. The theoretical scaling is based on Eqs. (52) and (54) (see Sec. III E 2), numerical values are taken from Table II, column 15.

we have

$$N_{\text{FIR}} \geq 2N_{\text{cyc}} \sim \int_{t_{\text{ini}}}^{t_c} \Omega dt = \int \frac{\Omega}{\dot{\Omega}} d\Omega \sim f_{\text{min}}^{-5/3}. \quad (51)$$

Converting the summation Eq. (50) into integral, we obtain:

$$\mathcal{C}_{\text{FIR}} \sim \int \Omega dN_{\text{FIR}} \sim f_{\text{min}}^{-2/3}. \quad (52)$$

For IIR filters, during a dephasing time of  $T = \sqrt{2\epsilon/\dot{\Omega}}$ , we use one filter  $\Delta N_{\text{IIR}} = 1$  and  $\Delta\Omega = \dot{\Omega}T = \sqrt{2\epsilon\dot{\Omega}}$ ,

which leads to

$$N_{\text{IIR}} = \int d\Omega \frac{dN_{\text{IIR}}}{d\Omega} = \int \frac{d\Omega}{\sqrt{2\epsilon\dot{\Omega}}} \sim f_{\text{min}}^{-5/6}. \quad (53)$$

The computational cost of IIR filtering is

$$\mathcal{C}_{\text{IIR}} \sim \int \Omega dN_{\text{IIR}} \sim f_{\text{max}}^{1/6} - f_{\text{min}}^{1/6}. \quad (54)$$

Note that for IIR filtering, the positive power law means the computational cost scales predominantly with the higher cut-off frequency, instead of the lower cut-off frequency — we therefore expect the computational cost not to increase dramatically when  $f_{\text{min}}$  is lowered, if we already have  $f_{\text{max}} \gg f_{\text{min}}$ .

## 2. Numerical Estimates

More detailed constructions for Advanced LIGO and Einstein Telescope (ET) have been carried out, following Sec. III D, assuming  $f_{\text{min}} = 10$  Hz for Advanced LIGO and 3 Hz for ET. Assuming the same successive  $2\times$  down-sampling strategy, we evaluate the single template computational cost for  $(1.4 + 1.4)M_\odot$  binaries for both FIR and IIR filtering. As it turns out, using the same  $\epsilon = 0.269$ , but  $(\alpha, \zeta) = (2.5, 4.25)$  for Advanced LIGO and  $(\alpha, \zeta) = (2.25, 4.5)$  for ET, will still give us match above 0.99.

The number of filters in each down-sampling band, as well as computational cost break-down for a single template are shown in the second and third tiers of Table II, for Advanced LIGO and ET, respectively. We also compare our numerical values with scaling laws predicted in Eqs. (52) and (54), which are plotted in dashed curves in Fig. 1. [We determined the normalization of the theoretical formulas using numerical values of computational

cost at  $f_{\min} = 40 \text{ Hz}$ .] The agreement is remarkable, especially considering that our successive 2-fold down-sampling is not continuous, and therefore rather non-ideal.

As we can see from Table II and Fig. 1, the IIR reduces computational cost from (multi-rate) FIR filtering by factor of 8 for initial LIGO. As we move to lower starting frequencies, the saving factor increases to 18 and 40, respectively. The single-template cost, even when we *extrapolate*  $f_{\min}$  to the rather unlikely 1 Hz, stays at several MFLOPS.

#### IV. INTERPOLATION BETWEEN IIR FILTERS OF DIFFERENT INSPIRAL WAVEFORMS

In order to search for all possible kinds of compact binary coalescence, we must match the signal with a *family of templates* parametrized continuously by the parameters of the binary, e.g., their masses. In practice, although maximization of match over certain parameters (e.g., initial orbital phase of the binary) can be done analytically, for the rest of the parameters, we must sample them discretely, and build a template bank — and match the signal with each member of the bank. The density of the discretization is usually determined by imposing that each member of the continuous family can be approximated well enough by at least one member of the bank, with mismatch less than a maximum tolerable value,  $\varepsilon_{\max}$ .

For advanced detectors, the number of templates can be as large as  $10^5$  [14, 15] posing a significant computational challenges. Interpolation strategies have therefore been conceived (e.g., [16–19]) to *reduce* the number of templates, based on the fact that signal-to-noise ratio is a continuous function of the parameters being searched over. More specifically, if we refer to the bank constructed by imposing the maximum tolerance of mismatch  $\varepsilon_{\max}$  as the *fine bank*, then the hope is that even if match is *calculated* for a *coarse bank* in which parameters are less densely populated, the signal-to-noise ratio of the fine bank can still be *recovered by interpolation*, in such a way that the total cost of *computing* coarse-bank SNRs plus *interpolating* fine-bank SNRs is less than the cost of directly computing fine-bank SNRs.

Motivated by Refs. [7, 8, 11, 12], our interpolation method also divides each coarse-bank template into several sub-templates in the frequency (thus time) domain, and recovers fine-bank SNRs using SNRs from the sub-templates. We will show that, although the division into sub-templates increases the cost of recombination, it allows a much coarser bank — and finally decreases the computational cost by a large factor.

#### A. Template banks in general

To develop a scheme to discretize the parameter space without losing detection efficiency, we must know how much the SNR is reduced by using a template whose parameter values differ from those of the signal. We define the mismatch between two normalized templates of different sets of parameters as

$$\varepsilon \equiv 1 - \langle u(\boldsymbol{\lambda}) | u(\boldsymbol{\lambda}') \rangle. \quad (55)$$

The template  $u$  is specified by a parameter vector  $\boldsymbol{\lambda}$ . If  $\boldsymbol{\lambda}'$  is near to  $\boldsymbol{\lambda}$ , we can Taylor expand  $\varepsilon$  at  $\boldsymbol{\lambda}$  and have the approximation to second order of  $\Delta\boldsymbol{\lambda} \equiv \boldsymbol{\lambda}' - \boldsymbol{\lambda}$  as

$$\varepsilon \simeq \frac{1}{2} \frac{\partial^2 \varepsilon}{\partial(\Delta\lambda_i) \partial(\Delta\lambda_j)} \Big|_{\Delta\boldsymbol{\lambda}=0} \Delta\lambda_i \Delta\lambda_j, \quad (56)$$

from which we define a (positive definite) metric in the parameter space

$$\gamma_{ij} \equiv \frac{1}{2} \frac{\partial^2 \varepsilon}{\partial(\Delta\lambda_i) \partial(\Delta\lambda_j)} \Big|_{\Delta\boldsymbol{\lambda}=0}. \quad (57)$$

Equations (56) and (57) indicates that mismatch between neighboring points in the parameter space can be viewed as distance measured by metric  $\gamma$ .

Suppose we would like to place a template bank in a  $D$ -dimensional parameter space, with a mismatch no higher than  $\varepsilon$ , then the most straightforward strategy would be laying down a cubic grid with *proper side length*  $dl$  measured by the metric  $\gamma_{ij}$ , such that template placed at each grid point will be able to cover a cube whose vertices are centers of neighboring cubes. This means we have

$$D(dl/2)^2 = \varepsilon. \quad (58)$$

The volume spanned by each cube (according to metric  $\gamma_{ij}$ ) is therefore

$$\Delta V = dl^D = (2\sqrt{\varepsilon/D})^D. \quad (59)$$

The total number of templates in the bank would be the total volume of the parameter space divided by the volume of each cell, or

$$\mathcal{N} = \frac{V_{\text{tot}}}{\Delta V} = \frac{\int d^D \boldsymbol{\lambda} \sqrt{\det \|\gamma_{ij}\|}}{(2\sqrt{\varepsilon/D})^D} \quad (60)$$

#### B. Newtonian Chirps

Through the Stationary-Phase Approximation [20], the Fourier Transform of a Newtonian Chirp can be written as

$$\tilde{u}(f; A, t_c, \phi_c) \propto f^{-7/6} e^{i(Af^{-5/3} + 2\pi f t_c + \phi_c)}, \quad f > 0, \quad (61)$$

and  $\tilde{u}(f) = \tilde{u}^*(-f)$  for  $f < 0$ . The mismatch between two neighboring templates with parameters  $(A, t_c, \phi_c)$  and  $(A + \Delta A, t_c + \Delta t_c, \phi_c + \Delta \phi_c)$  can be written as

$$\varepsilon(\Delta A, \Delta t_c, \Delta \phi_c) = 1 - \frac{\int_{f_{\min}}^{f_{\max}} \frac{f^{-7/3} \cos \Delta \Phi}{S_h(f)} df}{\int_{f_{\min}}^{f_{\max}} \frac{f^{-7/3}}{S_h(f)} df} \quad (62)$$

where

$$\Delta \Phi = f^{-5/3} \Delta A + 2\pi f \Delta t_c + \Delta \phi_c \quad (63)$$

Expanding Eq. (62) up to second order in  $\Delta \Phi$ , we obtain by comparing with Eqs. (56) and (57) the metric

$$\|\gamma_{ij}\| = \begin{bmatrix} I(-\frac{17}{3}) & I(-3) & I(-4) \\ * & I(-\frac{1}{3}) & I(-\frac{4}{3}) \\ * & * & I(-\frac{1}{3}) \end{bmatrix}, \quad (64)$$

where “\*” indicates terms obtainable by symmetry, and

$$I(\beta) = \frac{1}{2} \left[ \int_{f_{\min}}^{f_{\max}} df \frac{f^\beta}{S_h(f)} \right] / \left[ \int_{f_{\min}}^{f_{\max}} df \frac{f^{-7/3}}{S_h(f)} \right], \quad (65)$$

and we have used  $i = 1, 2, 3$  to label  $\Delta A$ ,  $2\pi \Delta t_c$  and  $\Delta \phi_c$ , respectively. Note the metric depends on the frequency division and noise spectral density only.

Here among the three parameters, search over  $\phi_c$  is done analytically, as discussed in Sec. III C, while search over  $t_c$  is carried out systematically at the sample rate — the only parameter left to discretize is  $A$ . Therefore,  $A$  is an intrinsic parameter as described previously. The correct way to place templates along intrinsic parameter directions is to “project out” the intrinsic parameters, as discussed, e.g., by Owen and Sathyaprakash [15].

In our case, the projected metric along direction  $A$  is one dimensional given by

$$g_{11} = \gamma_{11} - \frac{\gamma_{13}^2 \gamma_{22} - 2\gamma_{12} \gamma_{13} \gamma_{23} + \gamma_{12}^2 \gamma_{33}}{\gamma_{22} \gamma_{33} - \gamma_{23}^2} \quad (66)$$

which depends on  $f_{\min}$ ,  $f_{\max}$  and the noise curve  $S_h$  through  $I(\beta)$ . Following Eq. (60), the number of templates required to achieve a mismatch  $\varepsilon$  is then

$$\mathcal{N} = \frac{\sqrt{g_{11}}(A_{\max} - A_{\min})}{2\sqrt{\varepsilon}}, \quad (67)$$

where  $A_{\min}$  and  $A_{\max}$  are the minimum and maximum values of  $A$ . Here we can be more specific about template placement along the  $A$  direction. Given any  $A$ , which is associated with a member of the template bank, and suppose its mismatch with a neighboring template with  $A \pm \Delta A$  is  $\varepsilon_{\max}$ , or

$$g_{11}(\Delta A)^2 = \varepsilon_{\max} \quad (68)$$

then neighboring templates should be placed at  $A \pm 2\Delta A$ , therefore we have

$$\mathcal{N} = \frac{A_{\max} - A_{\min}}{2\Delta A} \quad (69)$$

which recovers Eq. (67).

Here we give the noise spectral density we use for initial LIGO, Advanced LIGO, and the Einstein Telescope. For the initial LIGO [21], we have  $x = f/(150 \text{ Hz})$  and

$$S_h(f) = 9 \cdot 10^{-46} [(4.49x)^{-56} + 0.16x^{-4.52} + 0.52 + 0.32x^2] \text{ Hz}^{-1}. \quad (70)$$

For Advanced LIGO [22], we have  $x = f/(215 \text{ Hz})$  and

$$S_h(f) = 10^{-49} \left[ x^{-4.14} - 5x^{-2} + 111 \frac{1 - x^2 + \frac{1}{2}x^4}{1 + \frac{1}{2}x^2} \right] \text{ Hz}^{-1}. \quad (71)$$

Note this is different from what is used in [8]. As a result, two methods are dealing with different number of templates for the same parameter space. Note also that we used Newtonian waveforms. These should be taken into account when we compare the computational cost of the two methods. For the Einstein Telescope, we adopt ET<sub>B</sub> — a broad-band configuration using conventional technologies [23, 24], for which  $x = f/(100 \text{ Hz})$  and

$$\sqrt{S_h(f)} = 10^{-25} \left( 2.39 \times 10^{-27} x^{-15.64} + 0.349x^{-2.145} + 1.76x^{-0.12} + 0.409x^{1.10} \right) \text{ Hz}^{-0.5}. \quad (72)$$

Applying Eqs. (67) and (68) to these three detectors, we can show that the number of templates increase by a factor of 3.9 when we upgrade from initial to Advanced LIGO, and another factor of 4.4 when we upgrade from Advanced LIGO to the Einstein Telescope. These numbers are listed in Table I, column 5.

## C. Subtemplates

### 1. General Discussion

Now suppose we divide our entire signal frequency interval,  $(f_{\min}, f_{\max})$  into  $M$  segments of

$$[f_0, f_1], [f_1, f_2], \dots, [f_{M-1}, f_M], \quad (73)$$

with  $f_0 = f_{\min}$  and  $f_n = f_{\max}$ . (When we later apply this to IIR filter chains,  $M$  will be much less than the total number of filters,  $N$ .) For any template  $u$ , we define sub-template  $u_J$ ,  $J = 1, \dots, M$ , to have the same value as template  $u$  within the frequency interval  $[f_{J-1}, f_J]$  but have zero values elsewhere,

$$\tilde{u}_J(f) = \begin{cases} \tilde{u}(f), & f_{J-1} \leq f \leq f_J, \\ 0, & \text{otherwise.} \end{cases} \quad (74)$$

Now let us consider two neighboring templates,  $u$  and  $v$ , their  $J^{\text{th}}$ -sub-innerproduct can be naturally defined as an integral over frequency segment  $J$ :

$$\langle u|v \rangle_J \equiv \langle u_J|v_J \rangle = 4\text{Re} \left[ \int_{f_{J-1}}^{f_J} df \frac{\tilde{u}^*(f) \tilde{v}(f)}{S_h(f)} \right]. \quad (75)$$

This sub-innerproduct can also be regarded as the contribution to the full inner product  $\langle u|v \rangle$  from segment  $J$  [Cf. Eq. (5)], and

$$\langle u|v \rangle = \sum_{J=1}^M \langle u|v \rangle_J \quad (76)$$

We denote  $u$  and  $u + \Delta u$  as neighboring templates, and we also define their  $J^{\text{th}}$ -sub-mismatch specific to interval  $J$ , in the intrinsic parameter space, as

$$\varepsilon_J \equiv 1 - \frac{\langle u|u + \Delta u \rangle_J}{\sqrt{\langle u|u \rangle_J \langle u + \Delta u|u + \Delta u \rangle_J}}, \quad (77)$$

which is equal to the “ordinary” mismatch between  $u_J$  and  $u_J + \Delta u_J$  as defined in Eq. (55). Up to second order in  $\Delta u$ , we can show that the total mismatch and the  $J^{\text{th}}$ -sub-mismatch are

$$\varepsilon = \frac{1}{2} \frac{\langle \Delta u|\Delta u \rangle}{\langle u|u \rangle} \quad (78)$$

$$\varepsilon_J = \frac{1}{2} \frac{\langle \Delta u|\Delta u \rangle_J}{\langle u|u \rangle_J} \quad (79)$$

Using Eq. (76), we can show that

$$\varepsilon = \sum_{J=1}^M \varepsilon_J \frac{\langle u|u \rangle_J}{\langle u|u \rangle}. \quad (80)$$

Since

$$\sum_{J=1}^M \frac{\langle u|u \rangle_J}{\langle u|u \rangle} = 1, \quad (81)$$

the overall mismatch is therefore a weighted average of the sub-mismatches. This means to achieve an overall mismatch of  $\varepsilon$ , we only need to make sure the sub-mismatches  $\varepsilon_J$  average to  $\varepsilon$ . This has dramatic implications in the sense that it allows the overall mismatch to be maintained by (1) dividing the frequency band into several frequency intervals with non-uniform sub-mismatches, (2) reducing the size of frequency intervals to allow larger step size for intrinsic parameters. These lay the foundation for our template interpolation method.

To qualitatively understand the reason that the grid size for intrinsic parameter placement can be enlarged when we restrict ourselves to smaller frequency intervals, we first note that in the frequency domain, it is the phase that we need to match, while the amplitude as a function of frequency is the same for all parameters. We note that the phase of  $\tilde{u}(f)$ , which we denote by  $\Phi(f)$ , is determined by  $A$ , as well as  $t_c$  and  $\phi_c$  (Eq. (61)). In Fig. 2, we plot the phase  $\Phi(f)$  for a particular set of parameters  $(A, t_c, \phi_c)$  in blue and also for a neighboring set of parameters  $(A + \Delta A, t_c, \phi_c)$  in red. If we were to use the template with parameter  $A$  to search for a signal with parameter  $A + \Delta A$ , we could shift  $\phi_c$  and  $t_c$  used in the search, which corresponds to shifting the blue curve by a

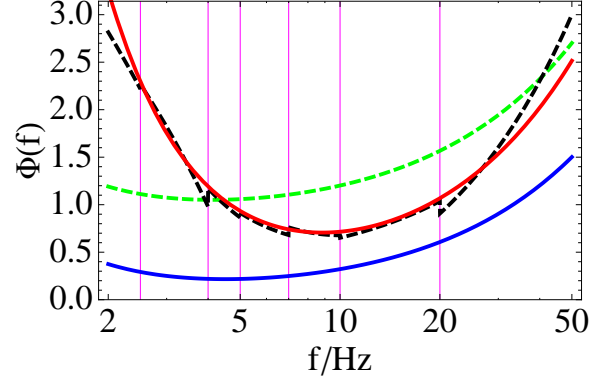


FIG. 2: Illustration of the phase function  $\Phi(f)$  vs frequency for the presumed parameter  $A$  (blue solid line) and its neighboring parameter  $A + \Delta A$  (red line), the linear shift of the blue line to match the red line (green dashed line), and a piecewise approximation (black dashed line) of the red line by shifting segments from the blue line. It shows that with smaller frequency intervals, it is easier to match phases arising from different intrinsic parameters.

linear function in frequency. The green dashed line illustrates a reasonably optimal attempt — yet the difference between the green curve and the red curve cannot be reconciled very well due to the fact that linear functions do not correct for curvature. However, if we divide the frequency range into several intervals, and allow different values of  $\Delta t_c$  and  $\Delta \phi_c$  to be applied to each interval, then sub-templates with  $A$  can achieve rather low sub-mismatches with signal with  $A + \Delta A$ . This corresponds to the fact that a curve can be better approximated by straight lines when divided into smaller intervals.

## 2. Newtonian Chirp in the Frequency Domain

Let us now focus on a particular frequency segment  $J$ , with  $f_{J-1} \leq f \leq f_J$ , and work out the relation between  $\Delta A$  and  $\varepsilon_J$ , as  $\Delta \phi_c$  and  $\Delta t_c$  are allowed to readjust their values (to be different from other segments). This simply requires us to repeat the procedure in Sec. IV B for each segment: with  $\Delta A$ ,  $\Delta t_c$  and  $\Delta \phi_c$ , we have the  $J^{\text{th}}$ -sub-mismatch of

$$\varepsilon_J = \begin{bmatrix} \Delta A & 2\pi\Delta t_c & \Delta \phi_c \end{bmatrix} \gamma^J \begin{bmatrix} \Delta A \\ 2\pi\Delta t_c \\ \Delta \phi_c \end{bmatrix} \quad (82)$$

with

$$\gamma^J \equiv \begin{bmatrix} I_J(-\frac{17}{3}) & I_J(-3) & I_J(-4) \\ * & I_J(-\frac{1}{3}) & I_J(-\frac{4}{3}) \\ * & * & I_J(-\frac{1}{3}) \end{bmatrix}. \quad (83)$$

and

$$I_J(\beta) \equiv \frac{1}{2} \left[ \int_{f_{J-1}}^{f_J} \frac{df f^\beta}{S_h(f)} \right] / \left[ \int_{f_{J-1}}^{f_J} \frac{df f^{-7/3}}{S_h(f)} \right]. \quad (84)$$

Note that the above are identical to Eqs. (62)–(64), except with integrations restricted to the interval of  $[f_{J-1}, f_J]$ .

The next step is similar to the “projection” process described by Owen and Sathyaprakash, but restricted to interval  $J$ . With Eq. (82), we ask the following question: if we are allowed to freely re-adjust individually the values of  $\Delta t_c$  and  $\Delta \phi_c$  for interval  $J$  of the template (i.e., the  $J^{\text{th}}$ -subtemplate), what would be the  $J^{\text{th}}$ -sub-mismatch achievable for  $\Delta A$ , and what should the corresponding  $\Delta \phi_c$  and  $\Delta t_c$  be.

The answer to the question is readily obtainable by a maximization of the mismatch  $\varepsilon$  over  $\Delta t_c$  and  $\Delta \phi_c$ , fixing  $\Delta A$ . This results in adjustments of

$$\begin{bmatrix} 2\pi\Delta t_c^J \\ \Delta \phi_c^J \end{bmatrix} = - \begin{bmatrix} \gamma_{22}^J & \gamma_{23}^J \\ \gamma_{32}^J & \gamma_{33}^J \end{bmatrix}^{-1} \begin{bmatrix} \gamma_{12}^J \\ \gamma_{13}^J \end{bmatrix} \Delta A \quad (85)$$

which result in the  $J^{\text{th}}$  sub-mismatch of

$$\varepsilon_J = g_{11}^J (\Delta A)^2, \quad (86)$$

with

$$g_{11}^J \equiv \gamma_{11}^J - \frac{[\gamma_{13}^J]^2 \gamma_{22}^J - 2\gamma_{12}^J \gamma_{13}^J \gamma_{23}^J + [\gamma_{12}^J]^2 \gamma_{33}^J}{\gamma_{22}^J \gamma_{33}^J - [\gamma_{23}^J]^2}. \quad (87)$$

Following Eq. (80), we have the total mismatch

$$\varepsilon = g_{11}^{\text{eff}} (\Delta A)^2, \quad (88)$$

where

$$g_{11}^{\text{eff}} = \sum_J \frac{g_{11}^J \langle u|u \rangle_J}{\langle u|u \rangle} \quad (89)$$

is an *effective metric coefficient* for any division of the frequency band. More specifically,  $g_{11}^{\text{eff}}$  describes the mismatch achievable by individually adjusting  $\Delta \phi_c^J$  and  $\Delta t_c^J$ , for each interval of the division. It can be shown that in general a finer division of frequency intervals always gives a smaller  $g_{11}^{\text{eff}}$ . As a consequence, if we define

$$\Delta A_{\text{cb}} = \sqrt{\frac{\varepsilon_{\text{max}}}{g_{11}^{\text{eff}}}}, \quad (90)$$

with the subscript “cb” indicating coarse bank, then  $\Delta A_{\text{cb}}$  will be greater than  $\Delta A$  given by Eq. (68) where  $g_{11}$  is evaluated using the full frequency band. In order to make a distinction, we shall rewrite that same equation as

$$\Delta A_{\text{fb}} = \sqrt{\frac{\varepsilon_{\text{max}}}{g_{11}}}, \quad (91)$$

but adding a subscript “fb” to indicate the fine bank. In order to maximize  $\Delta A_{\text{cb}}$  for a maximum mismatch  $\varepsilon_{\text{max}}$ , we should simply minimize  $g_{11}^{\text{eff}}$  globally, over all possible frequency division schemes. Because a template at  $A$  in the fine bank covers  $(A - \Delta A_{\text{fb}}, A + \Delta A_{\text{fb}})$ , the ratio of the number of templates in the coarse bank to that of the fine bank is,

$$\frac{\mathcal{N}_{\text{cb}}}{\mathcal{N}_{\text{fb}}} = \sqrt{\frac{g_{11}^{\text{eff}}}{g_{11}}} \quad (92)$$

In summary, given a required maximum mismatch  $\varepsilon_{\text{max}}$  with a particular frequency subdivision, by adjusting  $\Delta \phi_c^J$  and  $\Delta t_c^J$  individually, a single template at  $A$  can cover the region  $(A - \Delta A_{\text{cb}}, A + \Delta A_{\text{cb}})$ . For a signal with  $|\Delta A| \leq \Delta A_{\text{cb}}$ , the  $J^{\text{th}}$ -sub-template for parameter  $A + \Delta A$  can be constructed by adjusting  $\Delta t_c^J$  and  $\Delta \phi_c^J$  of the sub-template of template  $A$  using Eq. (85). The interpolated template of parameter  $A + \Delta A$  is therefore the sum of the constructed sub-templates from a coarse-bank template  $A$

$$\begin{aligned} \tilde{u}(f; A + \Delta A, t_c, \phi_c) &= \sum_{J=1}^M \tilde{u}_j(f; A, t_c + \Delta t_c^J(\Delta A), \phi_c + \Delta \phi_c^J(\Delta A)) \\ &= \sum_{J=1}^M \tilde{u}_j(f; A, t_c, \phi_c) e^{i2\pi f \Delta t_c^J(\Delta A) + i \Delta \phi_c^J(\Delta A)}. \end{aligned} \quad (93)$$

It is straightforward to establish the following properties of the effective metric: (i)  $g_{11}^{\text{eff}}$  always becomes smaller when we insert one or more dividing frequencies into an existing division of  $[f_{\text{min}}, f_{\text{max}}]$ , (ii) if we continue to decrease the maximum size of intervals, we can decrease  $g_{11}^{\text{eff}}$  indefinitely [in fact, for small intervals,  $g_{11}^J$  scales as  $(\Delta f)^5$ , which means  $g_{11}^{\text{eff}}$  should scale as  $(\Delta f)^4$ , and hence  $\Delta A$  scales as  $(\Delta f)^{-2}$ ]. Furthermore, for template families with *more than one parameter*, it is straightforward to generalize our result to

$$g_{ab}^{\text{eff}} = \sum_J \frac{g_{ab}^J \langle u|u \rangle_J}{\langle u|u \rangle} \quad (94)$$

with the number of templates in the coarse bank given by

$$\frac{\mathcal{N}_{\text{cb}}}{\mathcal{N}_{\text{fb}}} = \sqrt{\frac{\det \|g_{ab}^{\text{eff}}\|}{\det \|g_{ab}\|}} \quad (95)$$

#### D. Application to IIR filtering technique

In this section, we will apply the formalism developed in the previous subsection and discuss how we can implement IIR filter chains only for a much coarser bank of templates — while still obtaining SNRs for the entire

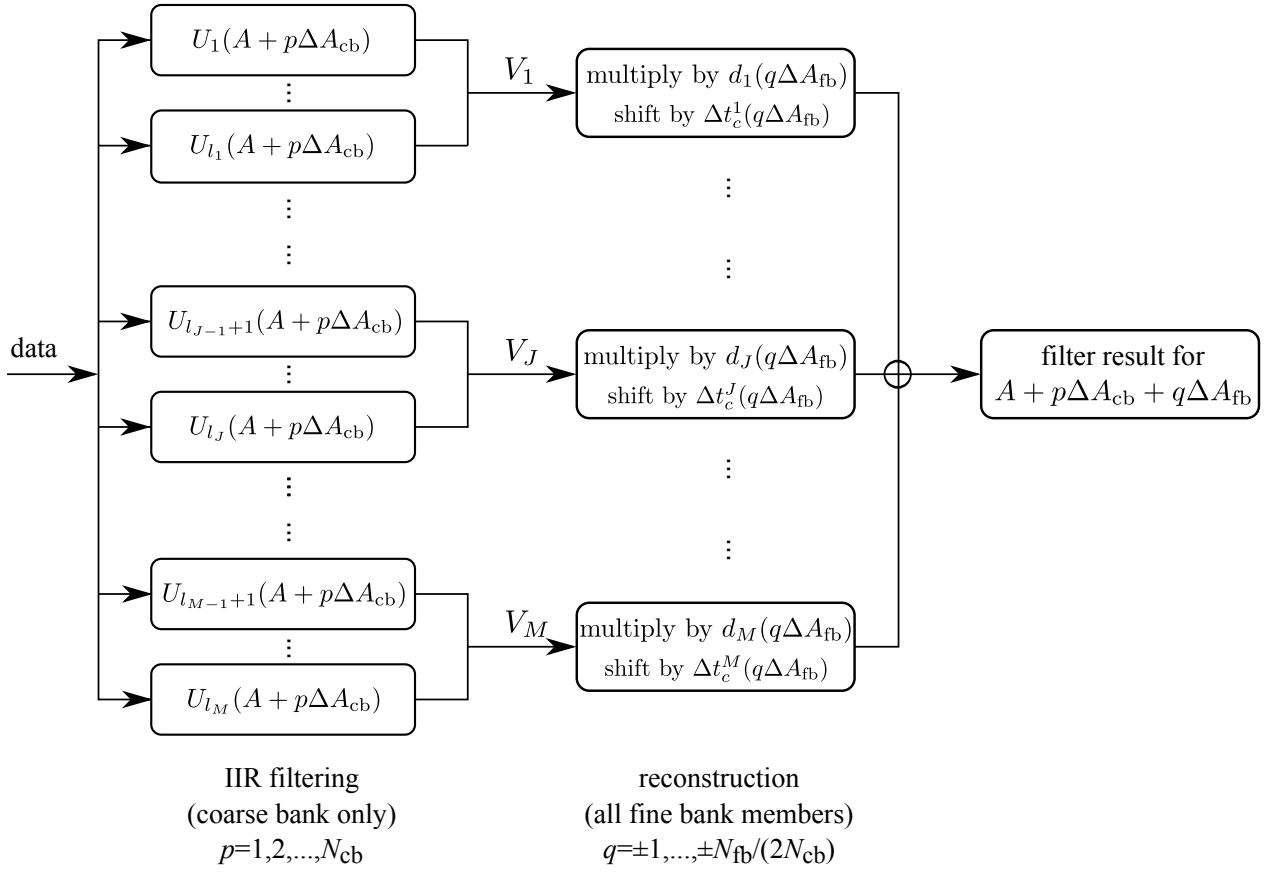


FIG. 3: Schematic diagram of the IIR filtering process for a template with parameter  $A + p\Delta A_{cb} + q\Delta A_{fb}$ . The first part is the IIR filtering for a member of the coarse bank,  $A + p\Delta A_{cb}$ , which produces a range of filter outputs, labeled by  $U_1 \dots U_{l_M}$ . These are grouped into  $M$  groups of summed IIR results  $V_1, \dots, V_M$ . The result for  $A + p\Delta A_{cb} + q\Delta A_{fb}$  is obtained by combining these  $V_J$ 's after each one is multiplied by  $d_J(q\Delta A_{fb})$  and shifted by  $\Delta t_c^J(q\Delta A_{fb})$ . The entire data analysis process still computes  $N_{fb}$  filter results, by including  $N_{cb}$  possible  $p$ 's and  $N_{fb}/N_{cb}$  possible  $q$ 's for each  $p$ . [In the special case of  $q = 0$ , the  $V_J$ 's are directly summed without having to go through multiplications and time shifts.] The downsampling or upsampling process is not shown.

fine template bank. Discussions made in the previous sections, although strictly speaking only apply to sharp divisions in the signal frequency band, still qualitatively apply to IIR filters that work in time-domain. The trick is to replace *frequency intervals* in the previous section by *groups of IIR filters*. This approach will work as long as we include enough number of filters in each “group”, so that overlaps between different groups are relatively unimportant. We note that, as is the case for the construction of IIR filter chains, the construction of the interpolation scheme by itself does not justify its efficiency — a separate test of achievable match will be carried out explicitly after the interpolation scheme is constructed.

To be more specific, we re-group the entire chain of  $N$  IIR filters into  $m$  sub-groups, with group  $J$  including those whose oscillation frequency lies within the frequency interval  $J$  defined in Sec. IV C. In other words,

group  $J$  of IIR filters can be written as

$$\begin{aligned}
 & V_J(t; A, t_c, \phi_c) \\
 &= \sum_{\frac{\Omega_l}{2\pi} \in [f_{J-1}, f_J]} U_l(t; A, t_c, \phi_c) \\
 &\equiv \sum_{l=l_{J-1}+1}^{l_J} U_l(t; A, t_c, \phi_c), \quad J = 1, \dots, M, \quad (96)
 \end{aligned}$$

where we have  $l_0 = 0$ . We will treat  $V_J$  as corresponding to the  $\tilde{u}_J(f; A, t_c, \phi_c)$  of Sec. IV C. As a consequence, from Eqs. (85) and (93), signal  $u(t; A + \Delta A, t_c, \phi_c)$  can be interpolated by the IIR filters constructed for

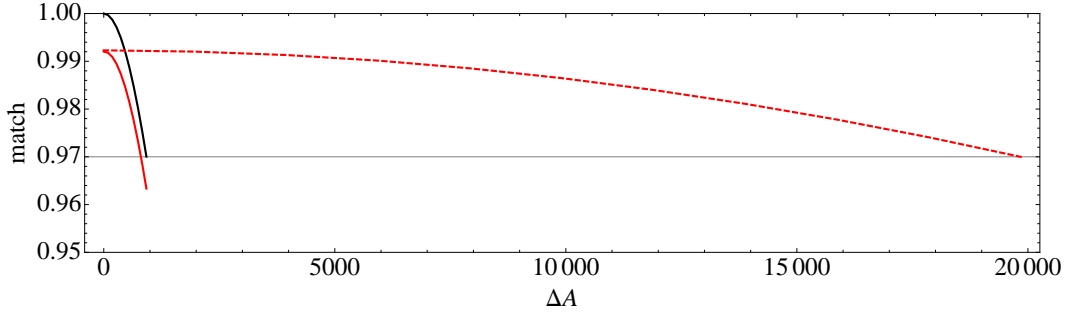


FIG. 4: Matches achievable with a Newtonian-Chirp signal at  $A + \Delta A$ , by various templates built for  $A$ , using initial LIGO noise spectral density for a  $(1.4+1.4)M_\odot$  NS-NS binary. Black solid curves corresponds to the result for a Newtonian-Chirp template, therefore the match is equal to unity at  $\Delta A = 0$ . Red solid curve corresponds to that using IIR filters, while red dashed curve corresponds to the interpolated match that can be recovered by using 6 filter subgroups.

$u(t; A, t_c, \phi_c)$

$$\begin{aligned} & u(t; A + \Delta A, t_c, \phi_c) \\ & \simeq \sum_{J=1}^M V_J(t; A, t_c + \Delta t_c^J, \phi_c + \Delta \phi_c^J) \\ & = \sum_{J=1}^M e^{i\Delta \phi_c^J} V_J(t; A, t_c + \Delta t_c^J, \phi_c). \end{aligned} \quad (97)$$

Here  $\Delta t_c^J$  and  $\Delta \phi_c^J$  should be computed from  $\Delta A$  using Eq. (85).

In practice we can easily generalize the coefficients in front of  $V_J$  to further reduce the overall mismatch, by using a slightly more general reconstruction formula:

$$u(t; A + \Delta A, t_c, \phi_c) \simeq \sum_{J=1}^m d_J V_J(t; A, t_c + \Delta t_c^J, \phi_c), \quad (98)$$

where  $d_J$  are complex coefficients that depend on  $\Delta A$ , given by

$$d_J = \sum_K T_{JK}^{-1} \langle V_K(A, t_c, \phi_c) | u(A + \Delta A, t_c, \phi_c) \rangle \quad (99)$$

with the matrix  $\mathbf{T}$  given by

$$T_{JK} = \langle V_J(A, t_c, \phi_c) | V_K(A, t_c, \phi_c) \rangle \quad (100)$$

### E. Full computational cost

Fig. 3 illustrates the procedure of obtaining the outputs from IIR filter chain for fine-bank coverage by interpolating coarse-bank filter outputs described previously. Upon obtaining outputs from subgroups of IIR filters for the *coarse bank*, we need to reconstruct outputs for all members of the *fine bank*. We hereby estimate the cost for reconstruction. Let's assume that a member of the fine bank that is *not* a member for the coarse bank is  $\Delta A$  away from a coarse-bank template  $A$ . For this  $\Delta A$ , we need to go through each group  $J$  of filters, take the total

output of this group (which corresponds to filtering by  $V_J$ ), multiply it by the complex number  $d_J$  (6 floating-point operations) and shift in time by  $\Delta t_c^J$ , and then add it to the sum (2 floating-point operations). The output eventually yields the SNR corresponding to the member of the fine bank. Note that both  $d_J$  and  $\Delta t_c^J$  are functions of  $\Delta A$ , but they do not need to be recalculated for each time step.

Assuming our frequency division is made in a way such that each filter group has the same sample rate ( $S_J$  for group  $J$ ), then the total recombination cost is

$$C_{\text{recom}} = \sum_J 8S_J. \quad (101)$$

In language of Sec. III D, if we assume there are  $N_{\text{group},k}$  IIR filter groups for each down-sampling channel, then the recombination cost can also be written as

$$C_{\text{recom}} = \sum_k 8S_k N_{\text{group},k}. \quad (102)$$

As a consequence, assuming that  $\Delta A_{\text{cb}} = R\Delta A_{\text{fb}}$ , we have a total cost of

$$\begin{aligned} C_{\text{total}} &= \mathcal{N}_{\text{fb}} \left[ \frac{C_{\text{IIR}}}{R} + \left(1 - \frac{1}{R}\right) C_{\text{recom}} \right] \\ &\approx \mathcal{N}_{\text{fb}} \sum_k \left[ \frac{12N_{\text{IIR},k}}{R} + 8N_{\text{group},k} \right] S_k, \end{aligned} \quad (103)$$

with the approximation valid when  $R \gg 1$ . In this case, we can have a good estimate of the computational cost of IIR filtering with interpolation. For a coarse bank with density  $1/R$  the fine bank, filtering cost naturally decreases to  $1/R$  of the cost of conventional IIR filtering without interpolation. The cost of recombination can be estimated with a simple rule: for each sample rate, the cost of recombination is about  $2/(3\bar{n}_k)$  times that of conventional IIR filtering, where

$$\bar{n}_k \equiv \frac{N_{\text{IIR},k}}{N_{\text{group},k}} \quad (104)$$

	$S_k \text{ (s}^{-1}\text{)}$	32	64	128	256	512	1024	2048	4096	8192	Cost
	$f/\text{Hz}$	4 – 8	8 – 16	16 – 32	32 – 64	64 – 128	128 – 256	256 – 512	512 – 1024	1024 –	Total
iLIGO	$N_{\text{group},k}$	0	0	0	1	2	1	1	0	1	
	$\{f_J\} \cap (f_{k-1}, f_k]$				52.9	71.0, 97.3	141	244		2000	0.10
	$\mathcal{C}_{\text{recomb},k}$				0.002	0.008	0.008	0.016		0.066	
aLIGO	$N_{\text{group},k}$	0	1	3	2	2	1	0	0	1	
	$\{f_J\} \cap (f_{k-1}, f_k]$		12.9	16.8, 22.1 29.6	40.1, 55.2	78.5, 122	228			2000	0.090
	$\mathcal{C}_{\text{recomb},k}$		0.0005	0.003	0.004	0.008	0.008			0.066	
ET <sub>B</sub>	$N_{\text{group},k}$	2	2	2	1	1	1	0	0	1	
	$\{f_J\} \cap (f_{k-1}, f_k]$	5.1, 6.9	9.3, 12.8	17.8, 25.3	37.4	60.3	122			2000	0.083
	$\mathcal{C}_{\text{recomb},k}$	0.0005	0.001	0.002	0.002	0.012	0.008			0.066	

TABLE III: Break-down of recombination cost required for obtaining one fine-bank template using the interpolation method, for initial, Advanced LIGO and the Einstein Telescope — assuming a successive two-fold down-sampling and ignoring the cost of down- and up-sampling. The IIR filter information is listed in Table II. For each down-sampling channel, we list the number of filter groups, as well as each of their upper-bound frequency (i.e.,  $f_J$  for group  $J$ ), and the computational cost as computed by Eq. (102). Computational cost here is measured by MFLOPS, or  $10^6$  FLOPS.

is the average number of IIR filters in groups at the  $k$ -th sample rate. As a consequence, the total cost of the IIR filtering with interpolation scheme including recombination can be lowered significantly if we achieve a balance of  $R \gg 1$  and  $\bar{n} \gg 1$ . Note larger  $R$  means larger coarse-bank grid size  $\Delta A_{cb}$  for a fixed  $\Delta A_{fb}$ . This is achieved by introducing finer frequency intervals. On the other hand, finer frequency intervals means more IIR groups  $N_{\text{group}}$  or smaller  $\bar{n}$  within each down-sampling channel.

The computational cost for performing down- or up-sampling is implementation-dependent (see discussions in [8]). They are not included in our calculation for simplicity. We only need to perform data downsampling once for all templates, so the cost should be negligible compared to the total cost. The upsampling process is needed at least for each coarse-bank template, but only for filter group outputs. Note the number of filter groups is much smaller than the total number of the IIR filters. Depending on the type of upsampling filters, the upsampling cost can be negligible compared to the total cost, but can also be in similar orders as the recombination cost. This requires further investigation.

#### F. Implementation for initial, Advanced LIGO and Einstein Telescope

We first investigate the case of initial LIGO to demonstrate the feasibility of our interpolation method. Taking into account the fact that even the optimal match between IIR filter and the real signal is not unity, we need to place the fine-bank IIR template a little denser than that from theoretical waveform. Theoretically for the Newtonian waveform, we have  $\Delta A_{fb} = 923$  (in units of  $\text{s}^{-5/3}$ ) to have a minimum match of 0.97 for templates based on the signal waveform. For the IIR filter bank, we need a smaller spacing of  $\Delta A_{fb}^{\text{IIR}} = 800 \text{ s}^{-5/3}$  in order for the bank to achieve the same match between an IIR tem-

plate and the signal at  $A + \Delta A_{fb}^{\text{IIR}}$ . Fig. 4 shows numerically calculated match as a function of template spacing  $\Delta A$  for templates from the signal waveform (black solid line) and for the IIR filters (red solid line) for the case of  $(1.4+1.4) M_\odot$  binary. Note that the numbers of fine-bank templates here are slightly different from those given in Table I, as we use slightly different overlap and also we use numerically evaluated matches here, instead of ones computed analytically assuming high match (in Sec. IV B).

To test the coarse-bank template placement, for simplicity, we restrict ourselves with the case of subdividing the frequency band into a total of six segments (or equivalently, six IIR filter groups in the time domain). According to the idealized theoretical calculations in frequency domain (Sec. IV C 2), the optimal frequency subdivision predicts  $\Delta A_{cb}/\Delta A_{fb} \approx 26$  for a minimum match of 0.97. This calculation has assumed high match, and divides signals into parts that are strictly localized within separate frequency bands. On the other hand, the numerical result using interpolation method on the IIR filter groups in the time domain (as prescribed in Sec. IV D) reveals that we can relax the coarse-bank spacing up to  $\Delta A_{cb}^{\text{IIR}} = 19845 \text{ s}^{-5/3}$  (dashed curve in Fig. 4), meaning

$$\frac{\Delta A_{cb}^{\text{IIR}}}{\Delta A_{fb}^{\text{IIR}}} \approx 25. \quad (105)$$

This is in very good agreement with the idealized prediction. Fig. 4 shows in dashed line the numerical result of the match as function of  $\Delta A$  for the interpolated IIR filtering method.

We can now evaluate the total computational cost of the entire filtering-reconstruction process. For filtering, since we only have

$$\mathcal{N}_c = (A_{\text{max}} - A_{\text{min}})/(2\Delta A_{cb}^{\text{IIR}}) = 92 \quad (106)$$

templates in the coarse bank [34], and the cost for each full filtering is 2.4 MFLOPS (see Table II), the cost of



IIR filtering is  $\mathcal{C}_{\text{IIR}}^{\text{bank}} = 221$  MFLOPS. Since the number of templates in the fine bank is

$$\mathcal{N}_f = (A_{\text{max}} - A_{\text{min}}) / (2\Delta A_{\text{fb}}^{\text{IIR}}) = 2281, \quad (107)$$

while the reconstruction cost for each member is 0.10 MFLOPS, the total cost for reconstruction (for members in the fine bank but not already in the coarse bank) is 228 MFLOP. Therefore the total cost for searching for Newtonian Chirps in initial LIGO is 449 MFLOPS, or 0.5 GFLOPS.

We carry out the same procedure for Advanced LIGO and ET<sub>B</sub>, with frequency division information listed in Table III, and interpolation factor as well as break-down of filtering and recombination costs listed in Table IV. As we can read from Table IV, the computational power required for a real-time search of Newtonian Chirps, using IIR filters and interpolation, in initial, Advanced LIGO and ET are 0.5 GFLOPS, 1.2 GFLOPS and 4.4 GFLOPS, respectively. The scaling of cost with  $f_{\text{min}}$  is rather mild as expected, and the cost, even for ET, seems very manageable.

In summary, it seems possible that to search for tens to hundreds of thousands of fine-bank templates for advanced LIGO or ET, we can have the entire search done with a few desktop computers and fewer if other acceleration technique such as the Graphics Processing Unit [25, 26] can be adopted. While our result is based on the Newtonian chirp, this outcome should be applicable to Post-Newtonian (PN) cases. Note the low-latency pipeline LLOID with the FIR scheme in combination with downsampling and SVD technique [8, 11] also predicts manageable computing power for Advanced LIGO. MBTA method [7], on the other hand, can already perform network analysis to search for inspiral signals using PN waveforms with a few CPUs for the initial LIGO. How it scales with advanced detectors while maintaining low latency remains to be investigated (see also Sec. V for a comparison of frequency vs time domain method). The integration of the time-domain IIR filtering method with the infrastructure of the LLOID pipeline is currently under way. Preliminary result for the application of the IIR filterbank method to PN waveforms can be found in [27] and [28].

## V. TIME DOMAIN VS FREQUENCY DOMAIN APPROACH

### A. General consideration

In terms of template interpolation, the ideas to divide the template into segments in the time or frequency domain are equivalent in mathematics – both trying to represent the template by the superposition of a complete basis of the continuous real-value function space on real axis. The functions in the basis are much simpler than the template, and thus easier to deal with. We can improve the computational efficiency by processing the ba-

sis functions first and then superpose them in the right way to get the result for a template. Given that the data we get from the detector is in the time domain, the advantage of working in the time domain is that we can avoid procedures required to transform the data into frequency domain (e.g., data accumulation in Fourier transformation) and easily achieve low time latencies. On the other hand, working in frequency domain allows us to easily combine the algorithm with down sampling technique and reduce the number of templates.

The frequency-domain template interpolation technique, e.g., that used in MBTA[7], usually uses Heaviside function to cut the template. So the template can be superposed smoothly in the frequency domain while in the time domain the joint of different basis functions can be quite crude. This means that those methods with this technique could easily take advantages of working in the frequency domain, but not both in the time and frequency domain without substantial additional cost in computation.

Our algorithm, with IIR filters working in the time domain and template interpolation designed from the frequency domain, takes advantages of the benefits from both the time and frequency domain approach. Because we use a relatively smooth cut in both domains, we can both achieve low latency in the time domain and reduce the total number of templates while taking advantages of the down sampling technique.

### B. Comparison of computational efficiency

When latencies of the analysis are not in concern, the frequency domain implementation of the cross correlation of data with templates (Eq. 13) is probably the most computationally efficient approach. This is due to the use of Fast Fourier Transform technique that has  $O(N \log N)$  operation count ( $N$  is the number of data points) as compared to the  $O(N^2)$  operation count for the FIR method described previously. On the other hand, the operation count of the IIR filterbank method is  $O(N)$  but multiplied with a coefficient directly related to the possibly large number of filters needed to achieve a desired match to the chirp signal. Here we take latencies into consideration and compare the computational efficiency of the FFT-based method with the proposed IIR method.

To obtain low-latencies for the FFT-based matched filtering prescribed in Eq. (13), the most straightforward approach is to analyze data in overlapping segments. We consider the analysis of equal-length segments of duration  $T_{\text{stretch}}$  as shown in Fig 5 with the duration of overlap equal to that of the longest signal, and the rest termed  $T_{\text{latency}}$ , that is,

$$T_{\text{stretch}} = T_{\text{longest}} + T_{\text{latency}}. \quad (108)$$

Here we assume the same strategy as in the current GW search pipeline where FFTs are performed with fixed length that accommodates the longest signal to ensure

	$\Delta A_{\text{fb}}^{\text{IIR}}$	$\mathcal{N}_{\text{fb}}$	$\Delta A_{\text{cb}}^{\text{IIR}}$	$\mathcal{N}_{\text{cb}}$	$\mathcal{C}_{\text{IIR}}$	$\mathcal{C}_{\text{IIR}}^{\text{total}}$	$\mathcal{C}_{\text{recomb}}$	$\mathcal{C}_{\text{recomb}}^{\text{total}}$	$\mathcal{C}_{\text{total}}$	$\mathcal{C}_{\text{total}}/\mathcal{C}_{\text{total}}^{\text{fb}}$
iLIGO	800	2281	19845	92	2.4	221	0.10	228	449	0.082
aLIGO	255	7156	10500	174	3.0	522	0.090	628	1150	0.054
ET	70	26069	2713	673	3.3	2221	0.083	2108	4329	0.050

TABLE IV: Break-down of total computational cost in MFLOPS in searching for Newtonian Chirps in initial LIGO, Advanced LIGO and ET, assuming interpolation for inspirals of 1–3  $M_{\odot}$  individual masses. Here we list numbers of templates in both the fine ( $\mathcal{N}_{\text{fb}}$ ) and coarse banks ( $\mathcal{N}_{\text{cb}}$ ), the computational cost for each full IIR chain ( $\mathcal{C}_{\text{IIR}}$ , taken from Table II), as well as the recombination cost for each template ( $\mathcal{C}_{\text{recomb}}$ , taken from Table III). We then give the total IIR filtering cost ( $\mathcal{C}_{\text{IIR}}^{\text{total}}$ ), the total recombination cost ( $\mathcal{C}_{\text{recomb}}^{\text{total}}$ ), and the grand total cost. We also list the ratio  $\mathcal{C}_{\text{total}}/\mathcal{C}_{\text{total}}^{\text{fb}}$ , in which  $\mathcal{C}_{\text{total}}^{\text{fb}}$  represents computational cost for the full bank without using interpolation.

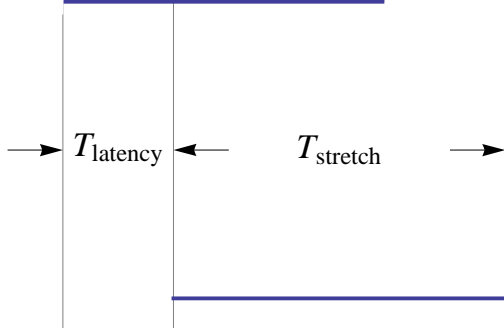


FIG. 5: Analysis with overlapping data segments. The two horizontal lines represent two adjacent data stretches used for FFT. The lower data segment starts data accumulation with a delay of  $T_{\text{latency}}$  relative to the upper one. The duration of the overlap between the two stretches is that of the longest signal in the template bank (see text in Sec. VB).

the coverage of signals of all possible duration. Note in practice, longer  $T_{\text{stretch}}$  might be needed to take into account of the windowing effect of the FFTs and issues like the sharp notch filter problems due to lines in the noise power spectrum [29]. For each data stretch, the output of Eq. (13) has also the duration  $T_{\text{stretch}}$ , but due to the wrap-around effect of FFTs, only outputs (for signals with ending time) within the last  $T_{\text{latency}}$  are valid. This means that to obtain a valid output of duration  $T_{\text{latency}}$ , a data stretch of at least  $T_{\text{longest}} + T_{\text{latency}}$  needs to be processed. The requirement to perform filtering in real-time implies that the entire analysis needs to be completed within  $T_{\text{latency}}$  seconds. The minimum total number of real multiplications and real additions for the FFT algorithm is about  $4N \log_2 N$  [30, 31]. Therefore the minimum computational cost in terms of FLOPS for each template for a real-time FFT-based matched filtering is at least,

$$\mathcal{C}_{\text{FFT}} = \frac{4S \cdot T_{\text{stretch}} \log_2(S \cdot T_{\text{stretch}})}{T_{\text{latency}}}, \quad (109)$$

where  $S$  is the data sampling rate. Here we assume a uniform sampling rate.

In the FFT method, the actual delay  $T_{\text{delay}}$  between the end time of a GW signal and the event triggering

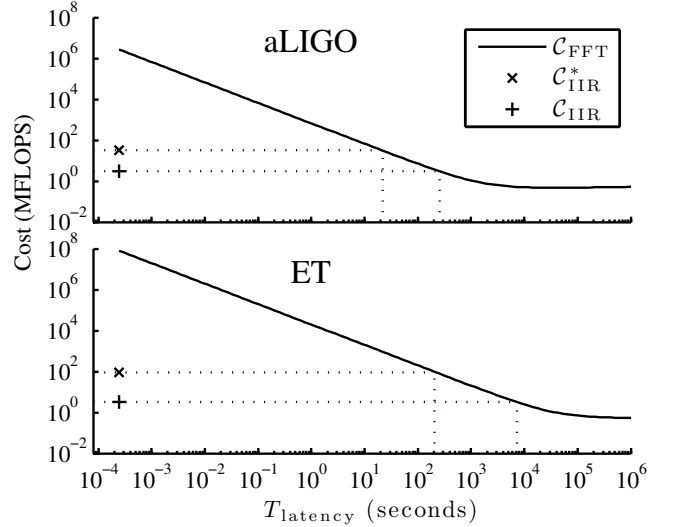


FIG. 6: Computational cost as a function of  $T_{\text{latency}}$  for a straightforward FFT analysis with overlapping data segments (solid line) and for the IIR filter method with downsampling technique (“+” symbols) and without (“x” symbols) for real-time filtering with one template of a (1.4+1.4)  $M_{\odot}$  binary. The upper panel shows the cost for aLIGO and the lower one for ET. The dotted lines illustrate the equal cost between the FFT and IIR method and the corresponding latencies. The computational cost of the FFT method is calculated from Eq. (109) with the longest template taken to be that of (1+1)  $M_{\odot}$  binary and sampling rate  $S=4096\text{Hz}$ . The IIR data are from Table II (column 15) (with downsampling) and Eq. (112) (without downsampling).

depends on where the signal lies in the data stretch. The longest delay occurs when the ending time of a signal lies ( $T_{\text{latency}} - dt$ ) before the end of a data stretch where  $dt \equiv 1/S$  is the sampling interval. In this case, after the signal ends, it takes the segment further ( $T_{\text{latency}} - dt$ ) time to finish accumulating data, and then another  $T_{\text{latency}}$  to be processed, resulting in a delay of,

$$T_{\text{delay}}^{\text{FFT, worst}} = 2T_{\text{latency}} - dt \approx 2T_{\text{latency}} \quad (110)$$

The shortest latency is achieved when the ending time of a signal lies just at the end of the data stretch, in which case the waiting time for the data to be analyzed is zero and the delay time of obtaining the trigger is simply the

analysis time,

$$T_{\text{delay}}^{\text{FFT,best}} = T_{\text{latency}}. \quad (111)$$

Therefore, for the FFT method, the delay time between the end of the signal and the event triggering is about 1–2 times of  $T_{\text{latency}}$ . Although in previous LIGO inspiral search pipelines  $T_{\text{latency}}$  is usually chosen so that adjacent data stretches are overlapped by 50%, it can be chosen so that  $T_{\text{latency}}$  is much smaller, meaning data segments are analyzed with larger overlaps and higher computational cost.

In comparison, for the IIR method, every new data point will be processed immediately when it is available. The delay time between the end of the signal and the triggering time can therefore in principle be as small as the data sampling interval. For real-time processing, the analysis time of the IIR filters at each time step should also be within one sampling interval,  $dt$ . As discussed previously, for each output of an IIR filter in (19), a total of 12 floating point operations are needed. Hence to produce the IIR filter bank output in real time without downsampling for one template requires the floating point operation per unit time of

$$C_{\text{IIR}}^* = 12S \cdot N_{\text{IIR}}, \quad (112)$$

and the delay

$$T_{\text{delay}}^{\text{IIR}} = dt. \quad (113)$$

Here asterisk is used to indicate the computational cost of the IIR filter method without the downsampling technique.

Fig. (6) shows the computational cost of one template for the FFT method as a function of  $T_{\text{latency}}$  when searching for a GW from a  $(1.4+1.4) M_{\odot}$  NS-NS binary and its comparison to that of the IIR filter method with and without downsampling technique. It shows that the computational cost of the FFT based method increases as latencies decreases, the increase is particularly significant at latencies less than hundreds to thousands of seconds (Eq (109)), whereas IIR methods (Eq.(47), Eq. (112)) have an inherent latency of the sampling interval (i.e **not** a function of latency). It is clear that the IIR filter method presented in this paper has significant advantage over the FFT method in computational efficiency when low latencies are in demand. In particular, for Advanced LIGO, the IIR method can be much more efficient at latencies less than a few  $\times 10^2$  seconds. For the Einstein Telescope, IIR filter method can be much more efficient at latencies less than a few  $\times 10^3$  seconds.

It should be mentioned that we compare only the core computational cost for the IIR and the FFT method for one template. We purposely leave out the cost of whitening or the cost to take care of other FFT effect such as windowing effect as they are very much implementation-dependent. We also do not include template interpolation method for both methods as they are very much

implementation-dependent. In practice, both methods require that the raw data be conditioned, transported, pre-whitened before they are ready to be analyzed. These are expected to cause additional latencies on the order of tens of seconds.

## VI. CONCLUSION

In this paper, we show that a time-domain search algorithm, with the flexibility of being able to detect a (non-precessing) compact-binary coalescence even before the final merger, is not only feasible for advanced and even future ground-based gravitational-wave detectors — but in fact can be realizable by a small number of state-of-the-art personal computers.

In addition to employing the multi-rate technique for time-domain filtering, we have developed two additional key techniques in order to bring down the computational cost into the realm of feasibility: (i) the conversion of a chirp signal into a chain of IIR filters, and (ii) an algorithm that allows the reconstruction of filtering results of a finely spaced template bank from a much coarser bank, when each template in the coarse bank is divided into sub-templates. In order to illustrate the main techniques, we have restricted ourselves to the Newtonian Chirp, but it is rather straightforward to generalize our algorithms into post-Newtonian templates.

Our main results on computational cost of the time-domain algorithm, for initial, advanced and future detectors, are summarized in Table IV. With a simple comparison, we also conclude that our time-domain algorithm should require less computational resources than the conventional frequency-domain approach, when a short latency of less than hundreds to thousands of seconds is required — as shown in Fig. 6.

Besides being computationally efficient at low (or even negative) latencies, the IIR filter bank method is also much simpler to implement than the FFT-based methods, making it ideal for parallel computing, e.g., with Graphics Processing Units [25].

Two further ingredients must be added into the search pipeline before we can set up an early-warning system for EM follow-ups of compact binary coalescence: (1) a reliable veto strategy, and (2) an efficient algorithm for sky localization. The fact that our numerical results for IIR filter groups agree so well with frequency-domain analytical estimates (Sec. IV C) assuming sharp divisions in frequency indicates that the sub-IIR-groups can be well-approximated as independent contributions to the SNR. This means a  $\chi^2$ -like test that compares relative SNR contributions from filter subgroups to their expectations would be a promising veto strategy (see also [32] for other strategies that might be applicable for further efficiency improvement.)

As for localization, we could in principle adopt the existing algorithm already in place in the LIGO/VIRGO pipeline, which is based on coincidences of SNRs among multiple detectors. Alternatively, the fact that IIR filters are separated in both time and frequency may pro-

vide a possibility of developing a *coherent* search pipeline with feasible computational cost. The reason for the high number of templates in a coherent search is directly due to the multiplication of the high number of templates along the direction of mass parameters and the high number of sky locations. However, as we divide each template into frequency segments, we find that in low frequencies, although there is a large number of cycles, and hence a requirement for a finer separation in mass parameters, the sky resolution of a detector network is low and there does not need a high number of sky patches; in high frequencies, we need a fine grid in the sky, but a coarse grid in mass parameters. As a consequence, we may need a much lower number of sub-templates are required for each frequency segment. This is currently being investigated.

## Acknowledgments

We thank Rana Adhikari for introducing us to the concept of IIR filters and for suggesting us to consider using IIRs for compact binary coalescence searches. We are grateful for inspiring discussions with Kipp Cannon, Chad Hanna, Drew Keppel, Alan Weinstein, Patrick Brady, Shin Kee Chung, David Blair, Benoit Mours, Leo Singer, Peter Shawhan, Bruce Allen, and Nickolas Fotopoulos. This work has been supported in part by NSF.PHY1068881, NSF Grants PHY-0601459, PHY-0653653, (LIGO) and CAREER Grant PHY-0956189 and the David and Barbara Groce start-up fund at Caltech, and by ARC Discovery Project and ARC Future Fellowship program.

- 
- [1] R. O'Shaughnessy, V. Kalogera, and K. Belczynski, *ApJ* **716**, 615 (2010), arXiv:0908.3635.
  - [2] J. Abadie, B. P. Abbott, R. Abbott, M. Abernathy, T. Accadia, F. Acernese, C. Adams, R. Adhikari, P. Ajith, B. Allen, et al., *Classical and Quantum Gravity* **27**, 173001 (2010), URL <http://stacks.iop.org/0264-9381/27/i=17/a=173001>.
  - [3] D. B. Fox, D. A. Frail, P. A. Price, S. R. Kulkarni, E. Berger, T. Piran, A. M. Soderberg, S. B. Cenko, P. B. Cameron, A. Gal-Yam, et al., *Nature* **437**, 845 (2005), arXiv:astro-ph/0510110.
  - [4] E. Nakar, *Phys. Rep.* **442**, 166 (2007), arXiv:astro-ph/0701748.
  - [5] L. S. Finn, *PRD* **46**, 5236 (1992), arXiv:gr-qc/9209010.
  - [6] C. Cutler and É. E. Flanagan, *PRD* **49**, 2658 (1994), arXiv:gr-qc/9402014.
  - [7] D. Buskulic, Virgo Collaboration, and LIGO Scientific Collaboration, *Classical and Quantum Gravity* **27**, 194013 (2010).
  - [8] K. Cannon, R. Cariou, A. Chapman, M. Crispín-Ortuzar, N. Fotopoulos, M. Frei, C. Hanna, E. Kara, D. Keppel, L. Liao, et al., *ArXiv e-prints* (2011), arXiv:1107.2665.
  - [9] L. R. Rabiner and B. Gold, *Theory and application of digital signal processing* (Englewood Cliffs, N.J., Prentice-Hall, Inc., 1975, 1975).
  - [10] J. Kanner, T. L. Huard, S. Márka, D. C. Murphy, J. Piscionere, M. Reed, and P. Shawhan, *Classical and Quantum Gravity* **25**, 184034 (2008), 0803.0312.
  - [11] K. Cannon, A. Chapman, C. Hanna, D. Keppel, A. C. Searle, and A. J. Weinstein, *PRD* **82**, 044025 (2010), arXiv:1005.0012.
  - [12] K. Cannon, C. Hanna, and D. Keppel, *ArXiv e-prints* (2011), arXiv:1101.4939.
  - [13] A. V. Oppenheim and R. W. Schaffer, *Digital signal processing* (1975).
  - [14] B. J. Owen, *PRD* **53**, 6749 (1996), arXiv:gr-qc/9511032.
  - [15] B. J. Owen and B. S. Sathyaprakash, *Phys. Rev. D* **60**, 022002 (1999).
  - [16] R. P. Croce, T. Demma, V. Pierro, I. M. Pinto, and F. Postiglione, *PRD* **62**, 124020 (2000), arXiv:gr-qc/0005012.
  - [17] R. P. Croce, T. Demma, V. Pierro, I. M. Pinto, D. Churches, and B. S. Sathyaprakash, *PRD* **62**, 121101 (2000), arXiv:gr-qc/0008059.
  - [18] S. Mitra, S. V. Dhurandhar, and L. S. Finn, *PRD* **72**, 102001 (2005), arXiv:gr-qc/0507011.
  - [19] S. E. Field, C. R. Galley, F. Herrmann, J. S. Hesthaven, E. Ochsner, and M. Tiglio, *Physical Review Letters* **106**, 221102 (2011), arXiv:1101.3765.
  - [20] B. S. Sathyaprakash and S. V. Dhurandhar, *PRD* **44**, 3819 (1991).
  - [21] B. P. Abbott, R. Abbott, R. Adhikari, P. Ajith, B. Allen, G. Allen, R. S. Amin, S. B. Anderson, W. G. Anderson, M. A. Arain, et al., *Reports on Progress in Physics* **72**, 076901 (2009), URL <http://stacks.iop.org/0034-4885/72/i=7/a=076901>.
  - [22] LIGO Scientific Collaboration, *Advanced LIGO anticipated sensitivity curves*, LIGO Document T-0900288-v2 (2009).
  - [23] The ET Science Team, *Einstein gravitational wave Telescope conceptual design study*, ET-0106B-10 (2011), URL <http://www.et-gw.eu/etsensitivities>.
  - [24] S. Hild, S. Chelkowski, and A. Freise, *ArXiv e-prints* (2008), 0810.0604.
  - [25] S. K. Chung, L. Wen, D. Blair, K. Cannon, and A. Datta, *Classical and Quantum Gravity* **27**, 135009 (2010), arXiv:0906.4175.
  - [26] S. K. Chung, L. Wen, D. Blair, and K. Cannon, in *American Institute of Physics Conference Series* (2010), vol. 1246 of *American Institute of Physics Conference Series*, pp. 207–210.
  - [27] S. Hooper, L. Wen, D. Blair, S. K. Chung, Y. Chen, and J. Luan, in *American Institute of Physics Conference Series* (2010), vol. 1246 of *American Institute of Physics Conference Series*, pp. 211–214.
  - [28] S. Hooper, S. K. Chung, J. Luan, D. Blair, Y. Chen, and L. Wen, *ArXiv e-prints* (2011), arXiv:1108.3186.
  - [29] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, *ArXiv General Relativity and Quantum Cosmology e-prints* (2005), arXiv:gr-qc/0509116.
  - [30] S. G. Johnson and M. Frigo, *IEEE Transactions on Signal Processing* **55**, 111 (2007).
  - [31] T. Lundy and J. van Busbirk, *Computing* **80**, 23 (2007).
  - [32] K. Cannon, C. Hanna, D. Keppel, and A. C. Searle, *PRD*

**83**, 084053 (2011), arXiv:1101.0584.

- [33] Because we start off from low enough frequencies, the estimate given here should be very accurate for the purpose of estimating computational cost.
- [34] Recall that since match between template at  $A$  and signal at  $A + \Delta A$  is already satisfactory, the next template needs to be placed at  $A + 2\Delta A$ .