Temporal Logic Robustness for General Signal Classes

ABSTRACT

In multi-agent systems, robots transmit their planned trajectories to each other or to a central controller, and each receiver plans its own actions by maximizing a measure of mission satisfaction. For missions expressed in temporal logic, the *robustness function* plays the role of satisfaction measure. Currently, a Piece-Wise Linear (PWL) or piece-wise constant fit is used at the receiver to reconstruct the continuous-time signal from the received samples. This allows an efficient robustness computation algorithm - a.k.a. *monitoring* - but is not adaptive to the signal class of interest, and does not leverage the compression properties of more general representations. When communication capacity is at a premium, this is a serious bottleneck. In this paper we first show that the robustness computation is significantly affected by how the continuous-time signal is reconstructed from the received samples, which can mean the difference between a successful control and a crash. We show that monitoring general spline-based reconstructions yields a smaller robustness error, and that it can be done with the same time complexity as monitoring the simpler PWL reconstructions. Thus robustness computation can now be adapted to the signal class of interest. We further show that the monitoring error is tightly upper-bounded by the L_∞ signal reconstruction error. We present a (non-linear) L_∞-based scheme which yields even lower monitoring error than the spline-based schemes (which have the advantage of being faster to compute), and illustrate all results on two case studies. As an application of these results, we show how time-frequency specifications can be efficiently monitored online.

KEYWORDS

Temporal Logic, Sampling, Robustness, Multi-Agent, Drones, Monitoring

ACM Reference Format:

. 2019. Temporal Logic Robustness for General Signal Classes. In *Proceedings of ACM HSCC conference (HSCC'2019).* ACM, New York, NY, USA, Article 4, 12 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Many Cyber-Physical Systems (CPS) require the exchange of information between components. As a first example, consider a multi-drone fleet tasked with a global mission (Fig. 1). Every drone periodically transmits its planned trajectory to its neighbors (in a distributed scheme) or to a central controller [19], and the receiver computes its next control actions based on how well the fleet's overall plan will meet the mission goals. As a second example, consider medical devices, like Insertable Cardiac Monitors, that must transmit the cardiac signals they record to a monitoring unit in a hospital [16]. The monitoring unit computes how far or close the current cardiac rhythm is from a 'normal' rhythm, and raises the alarm if long-term trends indicate a deterioration of the patient's condition. In both examples, the receiver determines whether the received signals satisfy or violate one or more specifications that are generally unknown to the transmitter.

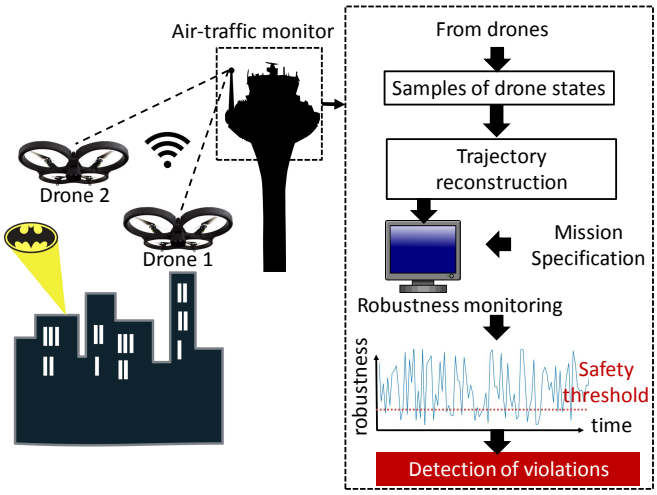


Figure 1: Remotely monitoring the safety of autonomous drones in an urban environment via a centralized air-traffic monitor.

In this work, the specification is formally captured in a Signal Temporal Logic (STL) formula, and the *robustness* of the STL formula quantifies how well the received signal satisfies the formula, or, conversely, how badly it violates it. It is therefore important to analyze how the robustness computation is affected by the *representation scheme* used for sampling the signal at the sender, and reconstructing it at the receiver. For an intuitive example, consider the signal shown in Fig. 2 with its various reconstructions. The signal measures the mutual separation between two autonomous quad-rotor drones, so a value of 0 indicates a collision. The STL formula for this data specifies, among other things, that the mutual separation should always be above a safety threshold, and below the communication range of the two drones most of the time. Clearly, if the robustness is computed on the black Piece-Wise Linear (PWL) reconstruction, it will suggest successful control, since the PWL signal satisfies the specification - but the purple cubic spline reconstruction reveals a specification violation - indeed, it shows a crash between the drones.

Of course, it is well-known that different bases have different signal reconstruction errors - but the effect of such errors on robustness computation have not been studied before, the choice of

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

HSCC'2019, April 2019, Montreal, Canada

© 2019 Copyright held by the owner/author(s).

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

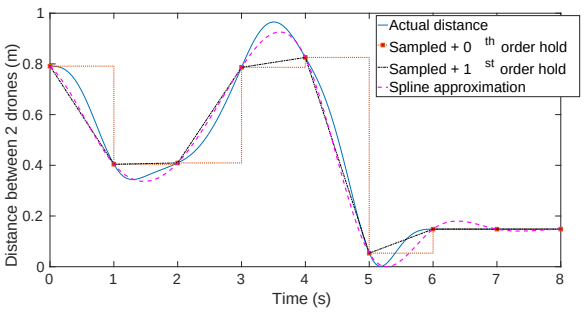


Figure 2: Distance between two quad-rotors, and its sampled/reconstructed versions.

appropriate bases has not been tackled, and is it not known whether it is possible to monitor general representation schemes efficiently. These questions are pervasive: they arise whenever a signal is compressed and transmitted to be monitored at the receiver, be it for control or verification. They are of a clear CPS nature, requiring an analysis of physical signal processing's effect on the robustness-guided digital controller or monitor. Answering these questions systematically is important, since a robustness value computed from the wrong basis can mean the difference between a crash and successful control, or between 'Normal' and 'Fatal' diagnoses. The answers are not a priori obvious - namely, better reconstruction does not necessarily mean a more accurate robustness value. Indeed, representation schemes seek to re-construct the entire signal (usually in the L_2 sense), while we only care for one summary value (the robustness). And of course, it is advantageous to use more compressive bases than, say, PWL used in [5], since simply increasing the number of samples is not an option.

Related work The example in Fig. 2 highlights the need for accurate computation of robustness. In practice, every analog signal is sampled to yield the sequence $(x(t_i))$. This raises the question of how to interpret an STL formula on the unavailable analog signal $\mathbf{x}$. In the literature, this has been addressed in one of two ways: either an explicit discrete-time (a.k.a. *pointwise*) semantics for the logic is used (as done in [4]), with some conditions to guarantee that the sampled sequence satisfies the formula only if the analog signal does [9]. These conditions are conservative, and impose formula-dependent restrictions on the sampling procedure (e.g., Assumption 2 in [9]), which is not desirable since the receiver might be monitoring multiple formulas, and the sender might not even know the formulas being monitored. They also require knowledge of certain quantities that might not be available, like the Lipschitz constants of the signals.

The second way this is approached in the literature is by using Piece-Wise Linear (PWL) interpolation to reconstruct an approximation $\tilde{\mathbf{x}}$ of $\mathbf{x}$ from the sample sequence, as done in the monitoring tool Breach [5].[1] PWL interpolation allows the development of an efficient monitoring algorithm. However, it might not be the best reconstruction scheme for the class of signals encountered in the application.

[1]Breach also supports piece-wise constant interpolation, which is a special case of PWL.

In [3], the related question of giving bounds on the robustness in online monitoring, given a priori bounds on signal values, is tackled. The notion of robustness is generalized in [12] to weaker algebraic structures and in [2] to include averaging over time, and the alternative notion in [15], developed for efficient control, does not capture the boolean truth value of the specification. The work in [21] accounts for missing samples via a statistical hypothesis test.

Contributions. We demonstrate empirically, on two case studies from drone fleets and cardiac monitoring, that the robustness computation is significantly affected by how the continuous-time signal is reconstructed from the received coefficients, which can change the decisions taken at the receiver in a meaningful manner (Section 3). We show that the monitoring error is tightly upper-bounded by the L_∞ signal reconstruction error, and present a (non-linear-filtering) L_∞-based representation scheme that yields the smallest monitoring error with today's most efficient monitor [5] for a given transmission size (Section 4). We then extend this monitor to handle spline-based representations beyond piece-wise linear. We show that monitoring these representations yields a smaller robustness error, while remaining in the same time complexity class as monitoring the PWL reconstructions (Section 5). As an application of these results, we show how time-frequency specifications can be efficiently monitored online (Section 6).

2 PRELIMINARIES: TEMPORAL LOGIC AND SIGNAL REPRESENTATIONS

A *signal* $\mathbf{x}$ is a function from $E \subseteq \mathbb{R}$ to $\mathbb{R}$. The signal's domain E is denoted $\mathbf{dom}\mathbf{x}$. The restriction of $\mathbf{x}$ to interval I is written $\mathbf{x}_{\restriction I}$. When E is countable we call $\mathbf{x}$ a *discrete-time signal*. Otherwise, if E is compact, we say $\mathbf{x}$ is a continuous-time, or *analog*, signal. Unless otherwise specified, all signals we use are analog. The first and second time derivatives are denoted $\mathbf{x}'$ and $\mathbf{x}''$ resp. The essential supremum and infinum are denoted ess sup and ess inf, resp. The sup norm of $\mathbf{x}$ is $\|\mathbf{x}\|_\infty := \operatorname{ess\,sup}_t x(t)$, and its p norm is $\|\mathbf{x}\|_p = (\int_{\mathbb{R}} |x(t)|^p)^{1/p}$, $1 \le p < \infty$. $L_p(E)$ is the space of real functions with domain E with finite p-norm, and $L_\infty(E)$ is the space of real functions on E with finite sup norm. When $E = \mathbb{R}$ we write L_p and L_∞. The inner product over a real function space is $\langle f, g\rangle := \int f(t)g(t)dt$.

2.1 Signal Temporal Logic and robustness

In verification, one wishes to determine whether the signal satisfies some specification like "Whenever $x > 2$, it stays there for at least 3 seconds". In control, one wishes to control the system producing the signal so that the latter satisfies the specification. Specification are formally expressed in *Signal Temporal Logic* (STL) [18], a language for expressing reactive temporal requirements, closely related to Metric Temporal Logic [13].

Let $M = \{\mu_1, \ldots, \mu_K\}$ be a set of Lipschitz *predicate* functions, $\mu_k : \mathbb{R} \to \mathbb{R}$, and let L be the largest Lipschitz constant of the set. Let $I \subset \mathbb{R}$ denote an interval, $\top$ the Boolean True, μ a predicate, $\neg$ and $\wedge$ the Boolean negation and AND operators, respectively. An STL formula ϕ is built recursively from the predicates as follows:

$$\phi := \top | \mu(x) \ge 0 | \neg\phi | \phi_1 \wedge \phi_2 | \phi_1 \mathcal{U}_I \phi_2$$

Informally, $\phi_1 \mathcal{U}_I \phi_2$ means that ϕ_2 must hold at some point in I, and *until* then, ϕ_1 must hold without interruption. The operators Always ($\Box$) and Eventually ($\Diamond$) can be derived from Until. Formally,

Definition 2.1 (STL semantics). *Let $E \subset \mathbb{R}$. The boolean truth value of ϕ w.r.t. signal $\mathbf{x} : E \to \mathbb{R}$ at time $t \in E$ is defined recursively.*

$$\begin{aligned}
(\mathbf{x}, t) &\models \top &\Leftrightarrow\quad & \top \\
\forall p_k \in AP, (\mathbf{x}, t) &\models p_k &\Leftrightarrow\quad & \mu_k(x_t) \geq 0 \\
(\mathbf{x}, t) &\models \neg\phi &\Leftrightarrow\quad & \neg(\mathbf{x}, t) \models \phi \\
(\mathbf{x}, t) &\models \phi_1 \wedge \phi_2 &\Leftrightarrow\quad & (\mathbf{x}, t) \models \phi_1 \text{ and } (\mathbf{x}, t) \models \phi_2 \\
\forall I \subset \mathbb{R}, (\mathbf{x}, t) &\models \phi_1 \mathcal{U}_I \phi_2 &\Leftrightarrow\quad & \exists t' \in (t + I) \cap E. (\mathbf{x}, t') \models \phi_2 \\
& & & \text{and } \forall t'' \in (t, t') \cap E,\ \ (\mathbf{x}, t'') \models \phi_1
\end{aligned}$$

We say $\mathbf{x}$ *satisfies* ϕ if $(\mathbf{x}, 0) \models \phi$. Otherwise we say $\mathbf{x}$ violates ϕ.

Designing a controller s.t. the closed-loop system satisfies the STL specification is not always enough. In a dynamic environment, where the system must react to unforeseen events, it is useful to have a margin of maneuverability by maximizing the *degree of satisfaction* of the specification. When unforeseen events occur, the system can react to them without violating the formula. This degree of satisfaction can be formally defined and computed using the *robust semantics* of temporal logic. In what follows, $a \sqcap b$ is the minimum of a and b, and $a \sqcup b$ is their maximum.

Definition 2.2 (Robustness[6, 8]). *The* robustness *of STL formula ϕ relative to $\mathbf{x} : E \to \mathbb{R}$ at time $t \in E$ is*

$$\begin{aligned}
\rho_\top(\mathbf{x}, t) &= +\infty \\
\rho_\mu(\mathbf{x}, t) &= \mu(x(t))\ \forall \mu \in M, \\
\rho_{\neg\phi}(\mathbf{x}, t) &= -\rho_\phi(\mathbf{x}, t) \\
\rho_{\phi_1 \wedge \phi_2}(\mathbf{x}, t) &= \rho_{\phi_1}(\mathbf{x}, t) \sqcap \rho_{\phi_2}(\mathbf{x}, t) \\
\rho_{\phi_1 \mathcal{U}_I \phi_2}(\mathbf{x}, t) &= \operatorname{ess\,sup}_{t' \in (t+I) \cap E} \Big(\rho_{\phi_2}(\mathbf{x}, t') \bigsqcap \\
&\qquad \operatorname{ess\,inf}_{t'' \in [t, t') \cap E} \rho_{\phi_1}(\mathbf{x}, t'') \Big)
\end{aligned}$$

When $t = 0$, we write $\rho_\phi(\mathbf{x})$ instead of $\rho_\phi(\mathbf{x}, 0)$.

The robustness[2] is a real-valued function of $\mathbf{x}$ with the following important properties.

Theorem 2.1. *[6, 8] For any $\mathbf{x} : E \to \mathbb{R}$ and STL formula ϕ, if $\rho_\phi(\mathbf{x}, t) < 0$ then $\mathbf{x}$ violates ϕ at time t, and if $\rho_\phi(\mathbf{x}, t) > 0$ then $\mathbf{x}$ satisfies ϕ at t. The case $\rho_\phi(\mathbf{x}, t) = 0$ is inconclusive.*

An algorithm that computes the robustness $\rho_\phi(\mathbf{x}, t)$ is called a *monitor*.

2.2 Signal decomposition and reconstruction

Notation. Let δ be the Dirac delta, $id : \mathbb{R} \to \mathbb{R}$ be the identity function $x \mapsto x$, and β_n be the order-n polynomial spline, defined recursively by:

$$\beta_0(t) = \begin{cases} 1 & , \quad |t| < 1/2 \\ 1/2 & , \quad t = 1/2 \\ 0 & , \quad |t| > 1/2 \end{cases}, \qquad \beta_n = \beta_{n-1} * \beta_0, n \geq 1 \qquad (1)$$

[2]Our definition has a slight technical difference with the usual definitions in that we use the *essential* supremum and infimum for the Until operator, instead of supremum and infimum. This is to match the definition of sup norm. For many signal classes of interest, like continuous signals, the two notions coincide.

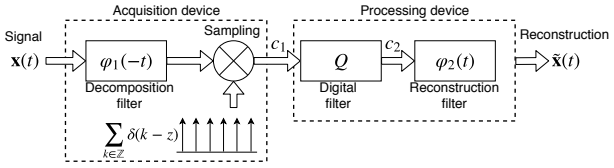


Figure 3: Filtering-based decomposition and reconstruction of signals [25]

Here, $f * g : y \mapsto \int_{-\infty}^{\infty} f(t) g(y-t) dt$ is the convolution of functions f and g. Note that $f * \delta = f$.

Filtering-based representations. When sending a signal $\mathbf{x}$, the transmitter must first convert the signal into a sequence of numbers, a process which we will refer to as *decomposition*. The receiver reconstructs an approximation $\tilde{\mathbf{x}}$ of $\mathbf{x}$ from the sequence. Linear filtering is widely used for both decomposition and reconstruction [25], as shown in Fig. 3. The signal is first convolved with the *decomposition* (or analysis, or acquisition) filter $\varphi_1(-t)$, is sampled, and the resulting coefficients ($c_1(k)$) are transmitted. At the receiver, a digital filter Q is applied to c_1 to yield ($c_2(k)$), then the approximation is constructed as

$$\tilde{\mathbf{x}}(t) = \sum_{k \in \mathbb{Z}} c_2(k) \varphi_2(t - k) \qquad (2)$$

(Note we assume sampling at the integers for simplicity, non-integer sampling times are handled by scaling the filters). Depending on the choice of filters φ_1, φ_2 and Q, different reconstructions with different properties are obtained. This paper considers three schemes. We present them in their special form that we use for robustness computation, their general form can be found in the cited references.

Default Scheme. This scheme uses $(\varphi_1, Q, \varphi_2) = (\delta, id, \beta_1)$. I.e. it simply samples the signal, and does a continuous Piece-Wise Linear (PWL) interpolation between the samples, a.k.a. *knots*, at the receiver.

L_2-optimal scheme [25]. In this case, $(\varphi_1, Q, \beta_n) = (\overset{o}{\beta}_n, id, \beta_n)$, where $\overset{o}{\varphi}$ is defined via its Fourier transform

$$\widehat{\overset{o}{\varphi}} := \frac{\widehat{\varphi}}{\sum_{k \in \mathbb{Z}} |\widehat{\varphi}(\omega + 2k\pi)|^2}$$

This scheme yields the reconstruction with minimum L_2 error.

Consistent Scheme [26]. In this case, $(\varphi_1, Q, \varphi_2) = (\delta, Q_{\text{int}}, \beta_n)$, with the filter's z-transform given by

$$Q_{\text{int}}(z) = \frac{1}{\sum_{k \in \mathbb{Z}} \beta_n(k) z^{-k}}$$

This scheme has the property that $\mathbf{x}(k) = \tilde{\mathbf{x}}(k)$ for all $k \in \mathbb{Z}$. It is used when one does not control the sender's choice of decomposition filter φ_1, and so the L_2 reconstruction error cannot be minimized. (E.g., here, δ has to be chosen). In this case, it is reasonable to require that at least the original and reconstructed signals yield the same measurements, and the digital filter Q ensures this.

We refer to these collectively as *filtering-based representation schemes*. In this paper we deal explicitly with scalar-valued signals only; the extension to higher dimensions follows straightforwardly from using separable decomposition and reconstruction bases.

2.3 Best uniform approximation

Because the Breach monitor uses PWL reconstruction from the knots $(t_i, \mathbf{x}(t_i))_i$, we propose the following scheme: for a maximum number $n_{\mathbf{x}}$ of transmitted knots,

1) *The transmitter* computes the PWL approximation $\tilde{\mathbf{x}}$ with smallest *uniform error* $\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty$ over $n_{\mathbf{x}}$ knots. Call this the Best Uniform approximation. This yields a sequence $(t_i, x(t_i))$ of knots, which are transmitted.

2) The receiver reconstructs $\tilde{\mathbf{x}}$ *perfectly* from the received knots by connecting them with lines, and evaluates $\rho_\phi(\tilde{\mathbf{x}}, \cdot)$ without error.

Thus the only error incurred is that of approximating $\mathbf{x}$ by $\tilde{\mathbf{x}}$ at the transmitter. The choice of uniform norm $\|\cdot\|_\infty$ to measure the error is theoretically justified in Section 4. For now, our goal is to introduce this scheme and evaluate its performance against the others, which we do in the next section.

3 THE EFFECT OF DIFFERENT REPRESENTATIONS ON ROBUSTNESS

We compare the accuracy of robustness computation for the different signal representation schemes from Section 2.2 in two case studies, one on autonomous drones and one on cardiac monitors.

In each case study, the input to Breach is the reconstructed $\tilde{\mathbf{x}}$. For a fair comparison, in all representation schemes, the number of transmitted coefficients $c_1(k)$ is the same, and equal to $1/20^{th}$ the length of the original signal. The ground truth for the comparison is the robustness computed on the original high-rate signal $\mathbf{x}$.

3.1 The two drone case study

3.1.1 Experimental setup. We collected data from simulations of two autonomous drones performing a joint reach-avoid mission (similar to [20]), while a centralized workstation is monitoring their mutual separation. The drones have to communicate with each other because they are performing a distributed optimization as part of their path planning algorithm. Therefore, they are allowed to go out of communication range (2m) for a limited duration at a time only (3 seconds). Moreover, they must always be at least $0.25m$ away from each other for safety. The specification is encoded in the STL formula:

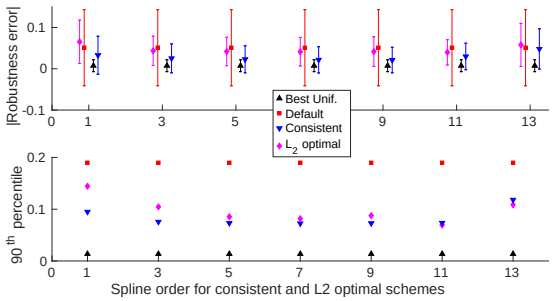
$$\phi_{\text{safe+comm}} = \Box((x \geq 2 \Rightarrow \Diamond_{[0,3]} x \leq 2) \wedge x \geq 0.25) \qquad (3)$$

We also monitor the following simpler spec starting at $t = 8s$ - i.e., this spec is relevant only after the initial take-off. It asks for a faster recovery to within communication range:

$$\phi_{\text{fast recovery}} = (x \geq 2 \Rightarrow \Diamond_{[0,1]} x \leq 2) \qquad (4)$$

Fifty traces were collected, with the drones starting from 50 randomly chosen initial positions. The original sampling rate is 20Hz, and the robustness of this signal is taken to be ground truth. For the signal decomposition in the schemes of Sec. 2.2, the signal is sampled, or the coefficients are computed, at 1Hz. For the Consistent and L_2-optimal schemes, we vary the spline order n over the odd integers between 1 and 13.

3.1.2 Results. For the specification $\phi_{\text{safe+comm}}$ of Eq. (3), Fig. 4 shows the mean and standard deviations of the absolute value of the robustness computation error, as well as the 90^{th} percentile of the absolute value of the robustness error. Across the different spline orders used in the *Consistent* and *L_2-optimal* schemes, the error is smaller than the error using the Default Scheme. The performance of the two schemes that rely on the β_n splines shows a slight improvement as the spline order n increases, until $n = 13$, where the error in the robustness value computed increases. Also shown in Fig. 4 are the mean, standard deviation, and 90^{th} percentile of the absolute error when using the Best Uniform scheme with the same number of knots as the filtering-based schemes. These errors are much smaller than those for the other schemes.



(a) Robustness computation errors

Best Unif. >	Consistent >	L_2-optimal >	Default

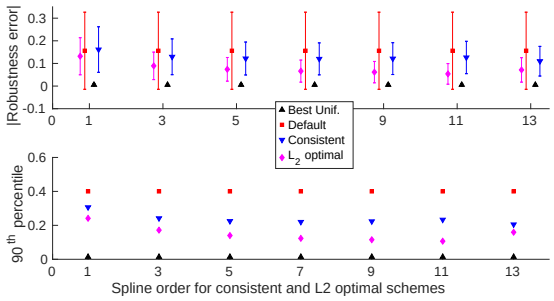
(b) Performance

Figure 4: Absolute error in computing the robustness of $\phi_{\text{safe+comm}}$ at time 0 (mean, standard deviation on top, 90^{th}-percentiles on bottom) for the different schemes with increasing order of splines. Default and Best Unif. performance is independent of spline order. Colors in digital copy.

Fig. 5 shows the mean, standard deviations and 90^{th} percentile of the absolute error in robustness computation for $\phi_{\text{fast recovery}}$, Eq. (4). Similar to the results for the more complex specification, the monitoring is more accurate when using *Consistent* and L_2-optimal than using Default Scheme. Of the three filtering-based schemes, *L_2-optimal* performs the best, and its performance improves with increasing order of the β_n spline. Finally, here too, Best Uniform performs best of all schemes.

3.1.3 Conclusions. These results clearly show that it is beneficial to use higher-order spline filters for signal representation and transmission, rather than simply relying on PWL interpolation between uniformly sampled signal values (the Default Scheme). We must emphasize that the choice of representation must be tied to the signal class one expects in the application, and there is no one-size-fits-all solution. *Our objective in the next section is to show that it is possible to efficiently monitor these more general signal reconstructions at the receiver - otherwise, there wouldn't be a substitute to PWL interpolation.*

The results also show that using the Best Uniform scheme outperforms all other schemes we studied, and so should be used if that is an option. (As we explain in Section 5, there might be compelling reasons forcing the use of a filtering-based scheme).



(a) Robustness computation errors

Best Unif. >>	L_2 optimal >	Consistent >	Default

(b) Performance

Figure 5: Absolute error in computing the robustness of $\phi_{\text{fast recovery}}$ at time 8 (mean, standard deviation on top, 90^{th}-percentiles on bottom) for the different schemes with increasing order of splines. Default and Best Unif. performance is independent of spline order. Colors in digital copy.

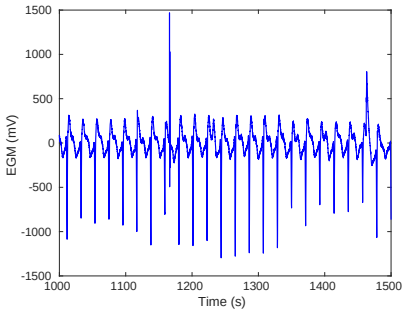


Figure 6: Cardiac electrogram during normal rhythm

3.2 The cardiac monitoring case study

3.2.1 Experimental Setup. Electrogram signals (Fig. 6) are used to diagnose heart conditions by physicians, and by implantable cardiac devices in a real-time manner [24]. While the analog signal may be acquired at a high rate by an Insertable Cardiac Monitor (ICM) - around 256Hz - it is compressed for periodic transmission. Weareable devices, which acquire a surface EGM-like signal, offer the potential to transmit these signals in real-time for remote collation and analysis, but because their measurements are much noisier, they are sampled at a low rate. For this case study, we envision such cases, where EGM signals, originally acquired at 1000Hz, are sampled and transmitted at 50Hz (or $1/20^{th}$ of the original data rate) to be monitored for a potentially fatal arrhythmia. Such a condition is captured in the following STL specification (x is measured in mV):

$$\begin{aligned}\phi_{EGM} = ((\Diamond_{[0,5]} x > 1000) \wedge \Box_{[0,20]}(x > 1000 \\ \Rightarrow \Diamond_{[1,5]} x > 1000)) \Rightarrow \Diamond_{[20,100]}(|x| < 250)\end{aligned} \tag{5}$$

This says that if the voltage exceeds 1V 5 times in the first 20 seconds, then it will settle below 0.25V in the 80 seconds following that.

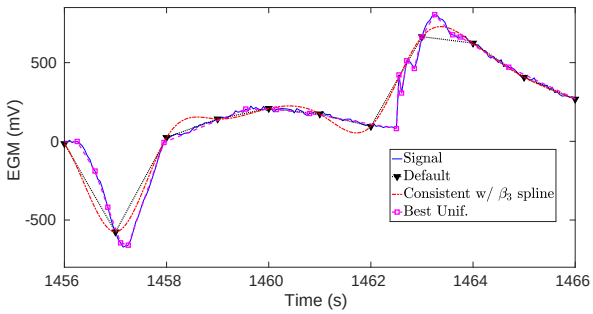
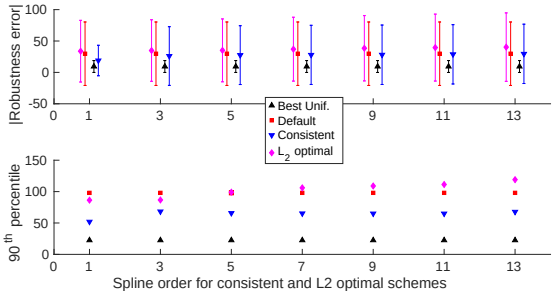


Figure 7: A zoomed-in view of 10 seconds of an EGM signal along with its reconstructions with the Default, Consistent β_3, and Best Uniform schemes with the same number of coefficients/samples. Colors in digital copy.



(a) Robustness computation errors

Best Unif. >>	Consistent >	Default ≥	L_2 optimal

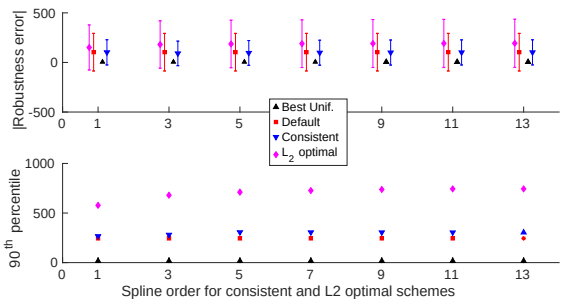
(b) Performance

Figure 8: Absolute error in computing the robustness of ϕ_{EGM} (mean, standard deviation on top, 90^{th}-percentiles on bottom) for the different schemes with increasing order of splines. Default and Best Unif. performance is independent of spline order. Colors in digital copy.

A violation suggests a sustained train of ectopic beats, indicative of disordered activity. We also monitor a simpler specification, which checks that there is activity in the EGM signal.

$$\phi_{\text{ no flatline}} = \Diamond(x \geq 5) \tag{6}$$

3.2.2 Results. We monitor the two formulas with the four representation schemes of Section 2.2, and at several spline orders for Consistent and L_2-optimal schemes. We measure the mean, standard deviation and 90^{th} percentile of the absolute error in robustness computation for each scheme. The ground truth is again provided by computing robustness on the high-rate original signal. To help interpret the results, we first provide Fig. 7, which shows a snapshot of an EGM signal and its approximations. At the signal peak around $t = 1464s$, the Default Scheme poorly approximates the peak, the Consistent Scheme does a better job, while the Best Uniform approximation (with the same number of samples as the other schemes) does best by using more samples to represent this part of the signal.



(a) Robustness computation errors

Best Unif.	>>	Consistent	≈	Default > L_2 optimal

(b) Performance

Figure 9: Absolute error in computing the robustness of $\phi_{\text{no flatfline}}$ (mean and std deviation on top, $90th$-percentiles on bottom) for the different schemes with increasing order of splines. Default and Best Unif. performance is independent of spline order. Colors in digital copy.

Fig. 8 gives the results for ϕ_{EGM}. First we note that the errors for the filtering-based schemes are significant. Also, unlike the Two Drone case, the L_2-*optimal* scheme performs similar to (or worse than) the *Default Scheme*. The *Consistent* scheme performs better than the other two filter-based schemes. Finally, Best Uniform produces the smallest robustness errors. This is because this formula is dependent on peak detection, and L_2 norm does not guarantee good approximation of narrow peaks.

This trend is more noticeable with the simpler $\phi_{\text{no flatline}}$, as shown in Fig. 9. The robustness of this specification is only dependent on the maximum value of the EGM signal. Working with the periodically sampled signal, as in the Default and Consistent schemes, results in similar 90^{th} percentile errors and means. The *consistent* scheme gives smaller standard deviation of the robustness computation error. Using Best Uniform (with the same number of samples) results in the best performance in this case as well (Fig. 9). This is expected, as it samples more around the peaks to give a tighter L_∞ approximation to the original signal.

3.3 Conclusions for the filtering-based schemes

The results for the Medical Monitoring case study do not follow the same trends as the Two Drone case study. This boils down to the different signal classes: smooth without large derivatives in Two Drones, and non-smooth with large derivatives and accelerations in Medical Monitoring. *This reinforces the need to choose the right representation scheme, and therefore the need for an efficient monitor that can handle a wider family of reconstructed signals, beyond PWL.* This is the subject of the next section.

3.4 Best Uniform approximation

Given the performance of Best Uniform in the previous section, here we study that performance as we vary the maximum allowed L_∞ reconstruction error, and compute the uniform PWL approximation $\tilde{\mathbf{x}}$ respecting this error bound with smallest number of knots,

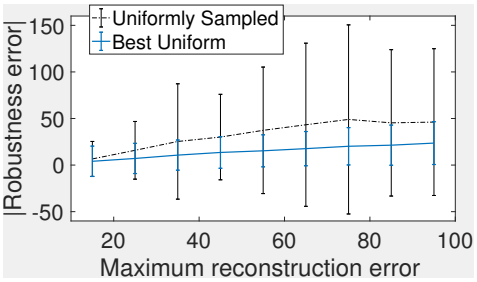


Figure 10: Error bars (mean ± std deviation) of robustness computation on the Medical Monitoring case study, vs. maximum allowed reconstruction errors. Colors in digital copy.

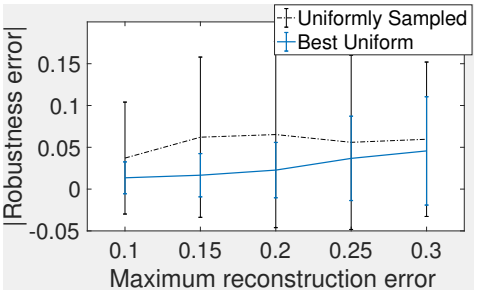


Figure 11: Error bars (mean ± std deviation) of robustness computation on the Two Drone case study, vs. maximum allowed reconstruction errors. Color in digital copy.

using a modified version of Dunham's dynamic program [7] (see Appendix E). We use formula ϕ_{EGM}. As a comparison basis, for each signal, we also evaluate the robustness of the PWL approximation whose knots are obtained by uniform sampling - call this $\mathbf{x}_u$. The "true" robustness for each signal was computed by sampling the signal at 1000Hz and feeding these samples to Breach. We repeat this experiment for various values of $\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty$, and compare the errors $|\rho_\phi(\mathbf{x}) - \rho_\phi(\tilde{\mathbf{x}})|$ and $|\rho_\phi(\mathbf{x}) - \rho_\phi(\mathbf{x}_u)|$.

Fig. 10 shows the error bars for robustness against the allowed reconstruction error for the EGM data set. As expected, using Best Uniform yields a lower average robustness error over the data set. Note also that Best Uniform yields a smaller variance, and that it is less sensitive to the allowed reconstruction error. The same conclusions hold over the Two Drone data set, shown in Fig. 11.

4 MONITORING THE BEST UNIFORM APPROXIMATION

In this section we explain the performance of Best Uniform, observed in the previous sections. Consider the signal $\mathbf{x}_n$ in Fig. 12, which approximates $\bar{0}$, the constant zero signal on [0,1], and the formula $\square(x \leq 1)$. The L_2 error $\|\bar{0} - \mathbf{x}_n\|_2 = 1/\sqrt{n} \to 0$, but the robustness error is $|\rho(\mathbf{x}_n) - \rho(\bar{0})| = |(1 - \sqrt{2}) - 1| = \sqrt{2}$ for all n. Thus controlling the L_2 signal reconstruction error does not always yield a control of the robustness computation error. The next result, a generalization of [1, Thm. 4.1], shows that we should control for

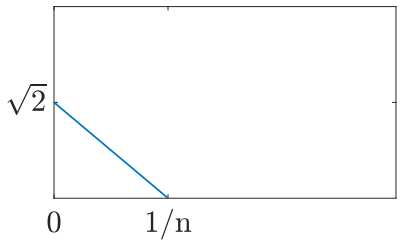


Figure 12: Small L_2 reconstruction error does not imply small robustness computation error.

the sup norm of the reconstruction error. Recall L is the largest Lipschitz constant of all predicates μ, and so is known.

Theorem 4.1. *Given two signals* $\mathbf{x}$ *and* $\tilde{\mathbf{x}}$ *and their difference* $e := \mathbf{x} - \tilde{\mathbf{x}}$*, it holds that*

$$\forall t, \forall \phi \in STL, |\rho_\phi(\mathbf{x}, t) - \rho_\phi(\tilde{\mathbf{x}}, t)| \leq L\|e\|_\infty$$

The bound is tight - i.e., for every signal there exists a formula where it holds with equality.

Proof. See Appendix. □

Because the bound is tight, and the transmitter does not know what formula is monitored at the receiver, the best one can do is to minimize $\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty$ by using Best Uniform.

The number of segments in the Best Uniform approximation of $\mathbf{x}$ over the interval $[a, b]$ behaves asymptotically as $c/\sqrt{\|e\|_\infty}$, where $c = 0.25 \int_a^b \sqrt{|\mathbf{x}''(t)|}dt$ [11]. This estimate helps choose the trade-off between robustness error and number of transmitted coefficients. This scheme naturally works for any reconstruction used at the receiver, e.g., piece-wise polynomial, *as long as there exists an efficient monitoring procedure for the resulting signal class* (and a procedure for computing the approximation at the transmitter).

5 MONITORING FILTERING-BASED APPROXIMATIONS

The results of the previous section demonstrate that filtering-based approximation schemes, based on the architecture of Fig. 3, outperform simple PWL (the Default Scheme), but are consistently out-performed by a uniform approximation. Nonetheless, it is important to study filtering-based schemes, because

- such schemes are pervasive in communication systems, and it might not be an option to modify existing infrastructure just for the purposes of robustness computation;
- the received signal can have multiple uses at the receiver, some of which require preserving the *shape* of the original signal, and
- these uses might require controlling the L_2 error specifically.

For example, in the multi-drone use case, a drone receives the planned trajectories of neighboring drones. In addition to computing their robustnesses (when coupled with its own plan), the receiving drone also does motion planning in continuous space. For this planning, it is important to preserve the shape of the others' trajectories. A PWL approximation of those trajectories is too inaccurate for this purpose. If the drone also uses robust Model Predictive Controller (MPC) for tracking, the robust MPC formulation can handle efficiently L_2-bounded measurement errors, so minimizing the L_2 reconstruction error matters.

This section explains how to monitor (i.e., compute the robustness of) signal reconstructions generated through filtering-based approximations at the same complexity as monitoring Default Scheme approximations, after a cheap pre-processing step. We require φ_2 to be a polynomial spline ($\varphi_2 = \beta_n, n > 1$), and place no restrictions on φ_1 and Q. Thus, this includes the Default, Consistent and L_2-optimal schemes as special cases. Then it is shown how to bound the L_∞ reconstruction error by the L_2 error for a wide class of signals, thus giving a partial explanation for why filtering-based schemes perform better than the Default Scheme.

5.1 Monitoring spline representations

First, an overview of the relevant elements of Breach's monitor is needed. Let $f : (a, b) \to \mathbb{R}$ be the signal to monitor. To simplify the exposition, as in [5], it is assumed that the predicates are all of the form $f(t) \geq a_p$. The monitor receives a finite sequence of n_f *knots* $(t_i, f(t_i), f'(t_i))_{i=1}^{n_f}$, and returns $\rho_\phi(\tilde{f}, \cdot) : (a, b) \to \mathbb{R}$, the *robustness signal* of the reconstruction $\tilde{f}$ relative to ϕ. In [5], it is assumed that $\tilde{f}$ is a PWL interpolation of the received sequence, which corresponds to using the Default Scheme.

Example 5.1. We first give an example of the monitor's recursive operation: if $\phi = \Diamond(p \wedge q)$, then the monitor first computes the robustness signals $y_1 = \tilde{f} - a_p$ and $y_2 = \tilde{f} - a_q$, which clearly are PWL. This is the first level of computation. It then computes the signal $y_3 = y_1 \sqcap y_2$, also PWL (but possibly with more knots generated at the intersections of y_1 and y_2). This is the second level. At the third and final level, it evaluates $\rho_\phi(\tilde{f}, t) = y_4(t) = sup_{s \geq t} y_3(s)$, which is also PWL. It is shown in [5] that every intermediate robustness signal y_k thus computed (at every level) is itself PWL. From one level of computation to the next, the algorithm carries forward the symbolic representation of the PWL approximation, namely its knots $(t_i, y(t_i), y'(t_i))_{i=1}^{n_y}$.

We now describe formally how the robust semantics of conjunction ($\wedge$) and untimed Eventually ($\Diamond$) operators are evaluated, which will be sufficient to explain how the computations can be efficiently generalized to polynomial spline reconstructions. See [5] for other operators. The computation is recursive.

Conjunction $\phi \wedge \psi$. Let $y = \rho_\phi(f, \cdot) \equiv (t_i, y(t_i), y'(t_i))_{i=1}^{n_y}$. and $w = \rho_\psi(f, \cdot) \equiv (s_j, w(s_j), w'(s_j))_{j=1}^{n_w}$. Both y and w are PWL as shown in [5]. We want to compute $z = y \sqcap w$. First, compute the sequence of times $(r_i)_{i=1}^{n_z} \in (a, b)$ which contains (t_i), (s_j), and the times at which y and w intersect - see Fig. 13. Note that 1) intersections can be efficiently computed for PWL signals, and 2) the new sequence (r_i) has length bounded as $n_z \leq 4 \max\{n_y, n_w\}$. Finally, z is given by $(z(r_i), z'(r_i)) = \min\{(y(r_i), y'(r_i)), (w(r_i), w'(r_i))\}$ (in lexicographical order).

Untimed eventually $\Diamond\phi$**.** If y is the robustness signal of ϕ and z that of $\Diamond\phi$, then for all $s < t$, $z(s) = z(t) \sqcup \sup_{[s,t)} y(s)$. Evaluating this last expression proceeds from the end of the signal backwards, initialized at $z(t_{n_y}) = -\infty$, iterating over the knots of $y \equiv (t_i, y(t_i), y'(t_i))_{i=1}^{n_y}$. Given the already-computed value $z(t_{i+1})$, there are 4 cases

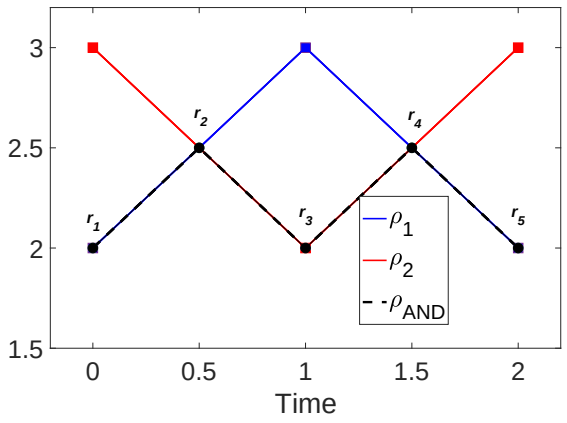


Figure 13: Monitoring a conjunction with PWL reconstruction. $\{r_i\}$ is the sequence of knot times for $\rho_{\mathbf{AND}} = \rho_1 \sqcap \rho_2$.

a) if $y(t_i) \leq y(t_{i+1})$ then $\forall s \in [t_i, t_{i+1}), z(s) = z(t_{i+1}) \sqcup y(t_{i+1})$,
b) if $y(t_i) > y(t_{i+1}) \geq z(t_{i+1})$, then $\forall s \in [t_i, t_{i+1}), z(s) = y(s)$
c) if $z(t_{i+1}) \geq y(t_i) > y(t_{i+1})$, then $\forall s \in [t_i, t_{i+1}), z(s) = z(t_{i+1})$
d) $y(t_i) > z(t_{i+1}) > y(t_{i+1})$, there exists $t^* \in [t_i, t_{i+1})$ s.t. $z(s) = y(s)$ on $[t_i, t^*)$ and $z(s) = z(t_{i+1})$ on $[t^*, t_{i+1})$.

The time complexity of any operator (i.e., at any level of the computation) is linear in the number of knots at that level, and the time complexity of the overall algorithm is given by the following.

Proposition 5.2. *[5] The time complexity of the Breach robustness monitor is $O(|\phi| \cdot d^{h(\phi)} n_f)$ where $d \leq 4$, $h(\phi)$ is the height of the formula's parse tree and $|\phi|$ is the number of nodes in the tree.*

Generalization to splines. The key observation, which the reader can make by studying the above two operators, is that piecewise linearity of the (reconstructed) signal is not essential for the monitor in [5] to be efficient. We have:

Proposition 5.3. *Let $\mathcal{F}$ be a class of signals supported on (a, b), and $\kappa : f \to (t_i, f(t_i), f'(t_i))^{n_f}$ be a map which returns a knot sequence for every $f \in \mathcal{F}$. If $\mathcal{F}$ and κ satisfy the following properties:*

1. *every $f \in \mathcal{F}$ admits a symbolic representation on every interval $[t_i, t_{i+1}]$ from its knot sequence $\kappa(f)$*
2. *every $f \in \mathcal{F}$ is monotone on every interval $[t_i, t_{i+1}]$ of its knot sequence*
3. *the number of intersections of any two signals y and w from $\mathcal{F}$ is at most a constant multiple of $n_y + n_w$.*
4. *these intersection points are computed in $O(1)$ time in $n_w + n_z, |\phi|, d$ and $h(\phi)$.*

Then the Breach monitor will compute the robustness of any signal in $\mathcal{F}$ with the same time complexity given in Prop. 5.2.

In particular, the time complexity at every level will still be linear in the number of knots at that level.

Therefore, we now exhibit such $\mathcal{F}$ and κ. Let $\mathcal{F}$ be the class of polynomial splines of order n, $f = \sum_k c_2(k)\beta_n(x - k)$, and let κ return the *inflection points* of f: points where it changes convexity and/or monotonicity, e.g. from convex decreasing to convex increasing, or from convex decreasing to concave decreasing, etc. These are the points where $f'(t)f''(t) = 0$. This choice satisfies properties (1)-(4). Indeed, it satisfies (1) as splines admit a symbolic representation, property (2) by construction of (t_i), and property (3) because on every refined interval $[t_i, t_{i+1}] \cap [s_j, s_{j+1}]$, two convex/concave monotone signals intersect at most twice. If they do intersect twice, at least one intersection is at an endpoint of the interval. It also satisfies property (4): finding the intersection points requires finding the zeros of the difference spline $y - w$. For splines up to order $n = 4$, this can be pre-solved analytically in terms of the spline coefficients and stored for online evaluation; for higher orders, one can use a zero-finding scheme that converges quadratically from any starting point [17]. Either way, this is $O(1)$.

Complexity. The complexity result Prop. 5.2 is in terms of the number of knots of the reconstruction, $n_{\tilde{\mathbf{x}}}$, but we seek to compare monitoring runtimes for the same number $T_{\mathbf{x}}$ of transmitted values. For the Default Scheme, $T_{\mathbf{x}}$ equals $n_{\tilde{\mathbf{x}}}$. For more general filtering-based schemes, that is not necessarily the case, since we transmit $T_{\mathbf{x}}$ spline coefficients, and the knots are inflection points of the reconstruction $\tilde{\mathbf{x}}$. We now prove the following.

Theorem 5.4. *The time complexity of the Breach monitor for polynomial spline reconstructions is $O(|\phi| \cdot d^{h(\phi)} T_{\mathbf{x}})$.*

Proof. We show that for spline reconstructions, $n_{\tilde{\mathbf{x}}}$ is at most linear in $T_{\mathbf{x}}$. Consider the reconstruction with $T_{\mathbf{x}}$ transmitted coefficients: $\tilde{\mathbf{x}} = \sum_{k \in K : |K| = T_{\mathbf{x}}} c_2(k)\beta_n(x - k)$ (a transmission is necessarily of finite duration). The inflection points of $\mathbf{x}$ are solutions of $\tilde{\mathbf{x}}'(t)\tilde{\mathbf{x}}''(t) = 0$, so there are at most as many inflection points as there are zeros of either $\mathbf{x}'$ or $\mathbf{x}''$: $n_{\tilde{\mathbf{x}}} \leq Z(\mathbf{x}') + Z(\mathbf{x}'')$. Clearly, $\tilde{\mathbf{x}}$ belongs to a spline space of dimension at most $T_{\mathbf{x}}$, and taking the derivative decreases the dimension by 1, so by a well-known property of splines [22]

$$Z(\mathbf{x}') + Z(\mathbf{x}'') \leq (T_{\mathbf{x}} - 1) - 1 + (T_{\mathbf{x}} - 2) - 1 = 2T_{\mathbf{x}} - 5$$

Finally,

$$n_{\tilde{\mathbf{x}}} \leq 2T_{\mathbf{x}} - 5$$

From this, and Props. 5.2 and 5.3, the result follows. □

In practice, there is an overhead due to computing the knots of $\tilde{\mathbf{x}}$, which requires finding the zeros of $\tilde{\mathbf{x}}'$ and $\tilde{\mathbf{x}}''$. This is a pre-processing step upon receipt of the spline coefficients, to which applies the above discussion on zero-finding methods.

Implementation. We created a (non-optimized) implementation of the modified robustness monitor in Matlab, and illustrate its operation in Fig. 14. The monitored signals, s_{12} and s_{23}, capture the pair-wise mutual separation between 3 drones, (we don't consider s_{13}). The formula is $\phi = \Diamond(s_{12} > \delta \wedge s_{23} > \delta)$, where δ is in meters. Fig. 14 shows the robustness signals for the predicates, $\rho_1 := \rho_{s_{12} > \delta}(\mathbf{x}, \cdot)$ and $\rho_2 := \rho_{s_{23} > \delta}(\mathbf{x}, \cdot)$, the intermediary signal ρ_{AND} from evaluating the AND, and the final robustness signal $\rho_{\text{Ev}} = \rho_{\phi}(\mathbf{x}, \cdot)$. It is clear that the final signal is in the same signal class as the monitored signals, which means it is significantly more accurate than a signal whose structure is imposed to be PWL or piece-wise constant.

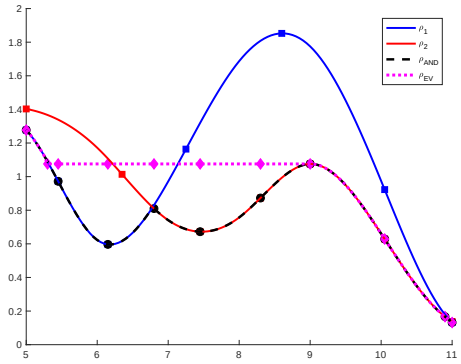


Figure 14: The modified monitor returns robustness signals that are in the same class as the monitored signals. Shown are the steps of computing $\rho_\phi(\mathbf{x}, \cdot)$ for mutual separation formula $\phi = \Diamond(s_{12} > \delta \wedge s_{23} > \delta)$. Final robustness signal is in purple (colors in digital copy). The knot points are also marked on each signal.

5.2 Bounding the L_∞ error in filtering-based schemes

When forced to use an existing filtering-based representation scheme, the results of the previous section showed that the user can at least use the reconstruction basis that is most appropriate for the signal class that appears in the application, without worrying about the robustness monitoring overhead. Filtering representations are evaluated based on the L_2 norm of the reconstruction error. As we saw in Thm. 4.1, it is the sup reconstruction error that matters when computing robustness. Yet we saw empirical evidence in Section 3 that some filtering-based schemes nonetheless perform well. By way of explaining this, in this section, we give conditions under which the L_2 norm of the reconstruction error upper-bounds its L_∞ norm over bounded intervals. The constants that appear in the bound are signal-dependent; it is well-known that, in general, no universal constants (that work for all signals) can exist (e.g., consider the signal family $e_n(x) = \sqrt{2n}(-nx + 1)$ over $[0, 1/n]$ and $e_n(x) = 0$ otherwise.).

THEOREM 5.5. *1. Let $\mathbf{x} \in L_\infty$, (a, b) be a bounded interval with $a \geq 0$, and set $e(t) := [\mathbf{x}(t) - \tilde{\mathbf{x}}(t)]_{\restriction (a,b)}$. Then there exists a t^* in (a, b) s.t. $\|e\|_\infty = \limsup_{t \to t^*} |e(t)|$.*
2. If $e \in L_\infty(a, b)$ and is continuous at t^* then there exists a constant C, dependent on $\mathbf{x}$, s.t. $\|e\|_\infty \leq C\|e\|_2$.

PROOF. See Appendix. □

The constant C appearing in the bound is not known a priori, and the empirical evaluation of the closeness between the two norms very much depends on the data set one uses. Thus this theorem provides only a partial explanation, but is nonetheless a start.

6 APPLICATION: ONLINE MONITORING OF TIME-FREQUENCY SPECIFICATIONS

The ability to monitor reconstructions with more general bases gives a more accurate robustness computation for the same number of transmitted values, as shown in previous sections. We now illustrate how to leverage other properties of more general bases, specifically, the localization of wavelet bases. The application is motivated by the following example, which uses the wavelet transform. See Appendix A for a review of the wavelet transform.

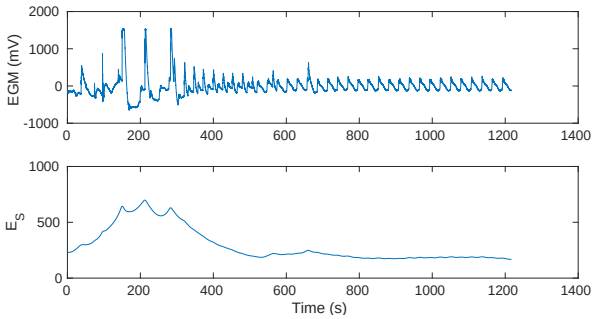


Figure 15: EGM during ventricular tachycardia (top) and its wavelet domain energy contained in a range $[s_1, s_2]$ (bottom)

Example 6.1. Consider the ventricular EGM of Fig. 15. In analyzing an EGM, its instantaneous amplitude in the time domain, $|\mathbf{x}(t)|$, and its instantaneous rate, are key features to determine the nature of the rhythm.[3] A *simplified* prototypical specification that one monitors for EGMs takes the form: "If instantaneous rate is above a threshold at some time t, then the amplitude of the signal is also above a threshold around that time". Violation of this specification might indicate ventricular fibrillation (a disorganized and fast rhythm with low amplitude), which causes the person to collapse within seconds [24]. The instantaneous rate is well-captured in the wavelet domain, because wavelets allow a precise measurement of the frequency content of the signal around specific points in time via the magnitude of the signal's wavelet transform $X(s, t)$ [10] (see Appendix). Namely, we can examine $E_S(t) = (\int_S |X(s, t)|^2 ds)^{1/2}$: this is the energy of the signal, measured in the wavelet domain, contained in the (scale) interval $S \subset \mathbb{R}$ where frequencies of interest (typical of fibrillation) occur[4]. Fig. 15 (bottom) shows E.

Thus we might formalize the above specification as

$$\phi_{VF} := \Box(|E_S| > b \implies \Diamond_{[0,\tau]}|x| > a)$$

The predicates in formula ϕ *carry over both the time and wavelet domains*. Formally, we have a 2D signal $(\mathbf{x}(t), E_S(t))$, and the predicates μ_k are functions of this 2D state.

If we are restricted to the Default Scheme, then we transmit samples $(x(t_i))$.[5] The receiver must then compute $X(s, t)$ and $E_S(t)$ to monitor. To compute $X(s, t) = \int \mathbf{x}(u)\psi_s(u - t)du$ at the larger scales s, we have to wait until most samples have been transmitted to perform the integration. This delays the start of calculation and decisions at the receiver.

Can we do online monitoring *at the receiver* - that is, start computing robustness as soon as the first few values arrive - by using a different representation scheme? We can. First we note two facts:

[3] The instantaneous rate is the rate computed over short windows whose size is dictated by the application, and is in general different from the instantaneous frequency.
[4] See [23, Appendix A] for scale-to-frequency mapping
[5] One could argue that we should transmit samples $(x(t_i), E_S(t_i))$. But that's an even larger number of values to transmit in the Default scheme.

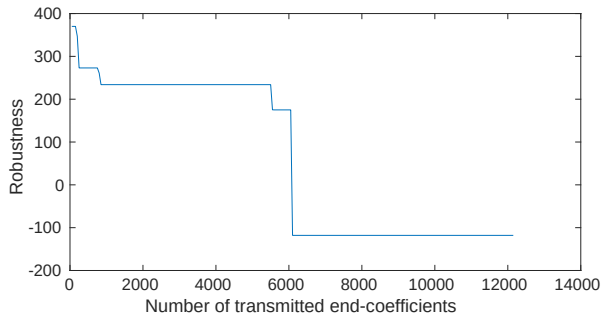


Figure 16: Robustness $\rho_{VF}(\mathbf{x}, 0)$ vs the number of transmitted coefficients $X(S, t)$, for the signal in Fig. 15.

• a wavelet reconstruction filter φ_2 fits into the general filtering-based scheme $(\varphi_1, Q, \varphi_2)$ of Section 2.2, so given an efficient way of finding the inflection points of a wavelet reconstruction, we can use the modified monitor of Section 5.1.
• As shown in Appendix B, all of Breach's computations can be evaluated forwards, processing the signal from time 0 to T.

Assume a finite-length signal $\mathbf{x}$ is to be transmitted, supported over $[0, T]$, and that we want to compute $\rho_\phi(\mathbf{x}, 0)$. The proposed scheme is as follows: At the transmitter, compute the wavelet transform X for all relevant scales $s \in S \subset \mathbb{R}$, and transmit these values as they're computed: $X(\cdot, 0), X(\cdot, \Delta t), X(\cdot, 2\Delta t) \ldots$, etc.[6] At the receiver, the monitor can *immediately* start computing $E_S(t)$, *and* reconstructing $\mathbf{x}_{\uparrow[0,\tau]}$ via the inverse wavelet transform; indeed, since wavelets are compactly supported, only $X(s, t'), t' - A_s \leq t \leq t + A_s, s \in S$, is needed to compute $\mathbf{x}(t)$, where $[-A_s, A_s]$ is the support of ψ_s. *This is not possible with the Euclidian basis used in the Default Scheme.* A fortiori, the monitor can start computing the robustness $\rho_\phi(\mathbf{x}, 0)$, *without waiting for the rest of the transmission.* Finally, note that in general, a far smaller number of wavelet coefficients needs to be transmitted for a good reconstruction of $\mathbf{x}$, than the number of samples $\mathbf{x}(t_i)$.

Fig. 16 shows the results of this scheme with the Bump wavelet used to analyze the signals in Fig. 15. With the first coefficients being received, the monitor starts computing robustness of the time-frequency specification ϕ_{VF}, and finally settles on a value.

7 CONCLUSIONS

We showed that when a signal is transmitted to be monitored at the receiver, the decomposition and reconstruction scheme has a significant effect on the accuracy of the computed robustness. We studied the performance of various schemes in terms of monitoring error experimentally on two data sets, then provided theoretical explanations for the empirical observations. We also demonstrated that we can compute the robustness for more general spline schemes without any increase in the complexity of the monitor, thus opening the way to a significantly more accurate monitoring of robustness that is adapted to the signal class of the application. The natural next step is to study how knowledge of the formula-to-monitor might guide the choice of representation basis and varying resolution along the signal.

[6] We ignore discretization questions relating to the wavelet transform. These are well-studied in the signal processing literature and are outside the scope of this paper. We could also use the discrete-time wavelet transform.

REFERENCES

[1] H. Abbas and G. Fainekos. 2013. Computing Descent Direction of MTL Robustness for Non-Linear Systems. In *American Control Conference*.
[2] Takumi Akazaki and Ichiro Hasuo. 2015. Time Robustness in MTL and Expressivity in Hybrid System Falsification. In *Computer Aided Verification*, Daniel Kroening and Corina S. Păsăreanu (Eds.). Springer International Publishing, Cham, 356–374.
[3] Jyotirmoy V. Deshmukh, Alexandre Donzé, Shromona Ghosh, Xiaoqing Jin, Garvit Juniwal, and Sanjit A. Seshia. 2015. Robust Online Monitoring of Signal Temporal Logic. In *Runtime Verification*, Ezio Bartocci and Rupak Majumdar (Eds.). Springer International Publishing, Cham, 55–70.
[4] A. Dokhanchi, B. Hoxha, and G. Fainekos. 2014. Online Monitoring for Temporal Logic Robustness. In *Proc. of Runtime Verification*.
[5] Alexandre Donzé, Thomas Ferrère, and Oded Maler. 2013. Efficient Robust Monitoring for STL. In *Computer Aided Verification*, Natasha Sharygina and Helmut Veith (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 264–279.
[6] Alexandre Donzé and Oded Maler. 2010. Robust Satisfaction of Temporal Logic over Real-valued Signals. In *Proceedings of the International Conference on Formal Modeling and Analysis of Timed Systems*.
[7] James George Dunham. 1986. Optimum Uniform Piecewise Linear Approximation of Planar Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8 (1986), 67–75.
[8] G. Fainekos and G. Pappas. 2009. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science* (2009).
[9] Georgios E. Fainekos and George J. Pappas. 2007. Robust Sampling for MITL Specifications. In *Formal Modeling and Analysis of Timed Systems*, Jean-François Raskin and P. S. Thiagarajan (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 147–162.
[10] Michael W. Frazier. 1999. *A Linear Algebra Introduction to Wavelets.* Springer.
[11] C.L. Frenzen, Tsutomu Sasao, and Jon T. Butler. 2010. On the number of segments needed in a piecewise linear approximation. *J. Comput. Appl. Math.* 234, 2 (2010), 437 – 446. https://doi.org/10.1016/j.cam.2009.12.035
[12] Stefan Jaksic, Ezio Bartocci, Radu Grosu, and Dejan Nickovic. 2018. An Algebraic Framework for Runtime Verification. arXiv:cs.LO/1802.03775
[13] R. Koymans. 1990. Specifying Real-Time Properties with Metric Temporal Logic. *Real-Time Systems* 2, 4 (1990), 255–299.
[14] Alois Kufner and Lars-Erik Persson. [n. d.]. *Weighted Inequalities of Hardy Type.* World Scientific.
[15] L. Lindemann and D. V. Dimarogonas. 2017. Robust motion planning employing signal temporal logic. In *2017 American Control Conference (ACC)*. 2950–2955. https://doi.org/10.23919/ACC.2017.7963399
[16] Medtronic. 2018. http://www.medtronic.com/us-en/patients/treatments-therapies/heart-monitors/our-monitors/reveal-linq-icm.html
[17] Knut Mørken and Martin Reimers. 2007. An unconditionally convergent method for computing zeros of splines and polynomials. *Math. Comp* 76, 258 (January 2007), 845–865.
[18] Dejan Nickovic and Oded Maler. 2007. AMT: A Property-Based Monitoring Tool for Analog Systems. In *FORMATS (LNCS)*, Vol. 4763. Springer, 304–319.
[19] Yash Vardhan Pant, Houssam Abbas, and Rahul Mangharam. 2017. Smooth operator: Control using the smooth robustness of temporal logic. In *Control Technology and Applications (CCTA), 2017 IEEE Conference on*. IEEE.
[20] Yash Vardhan Pant, Houssam Abbas, Rhudii A Quaye, and Rahul Mangharam. 2018. Fly-by-logic: Control of multi-drone fleets with temporal logic objectives. In *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*. IEEE Press.
[21] Nima Roohi, Ramneet Kaur, James Weimer, Oleg Sokolsky, and Insup Lee. 2018. Parameter Invariant Monitoring for Signal Temporal Logic. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (Part of CPS Week) (HSCC '18)*. ACM, New York, NY, USA, 187–196. https://doi.org/10.1145/3178126.3178140
[22] Larry L. Schumaker. 2007. *Spline functions: basic theory.* Cambridge University Press.
[23] E. Sejdic, I. Djurovic, and L. Stankovic. 2008. Quantitative Performance Analysis of Scalogram as Instantaneous Frequency Estimator. *IEEE Transactions on Signal Processing* 56, 8 (Aug 2008), 3837–3845. https://doi.org/10.1109/TSP.2008.924856
[24] Roland X. Stroobandt, S. Serge Barold, and Alfons F. Sinnaeve. 2009. *Implantable Cardioverter - Defibrillators Step by Step.* Wiley.
[25] M. Unser. 2000. Sampling-50 years after Shannon. *Proc. IEEE* 88, 4 (April 2000), 569–587. https://doi.org/10.1109/5.843002
[26] M. Unser and A. Aldroubi. 1994. A general sampling theory for nonideal acquisition devices. *IEEE Transactions on Signal Processing* 42, 11 (Nov 1994), 2915–2925. https://doi.org/10.1109/78.330352

A WAVELETS PRIMER

Let $\{\psi_s\}_{s>0}$ be a family of functions, called *wavelets*, which are obtained by scaling and dilating a so-called *mother wavelet* $\psi(t)$: $\psi_s(t) = \frac{1}{\sqrt{|s|}}\psi\left(\frac{t}{s}\right)$. The *Continuous Wavelet Transform* (CWT) X of signal $\mathbf{x} : \mathbb{R}_+ \to \mathbb{R}$ is the two-parameter function:

$$X(s,\, t) = \int_{-\infty}^{+\infty} \mathbf{x}(\tau)\overline{\psi_s}(\tau - t)\, d\tau \tag{7}$$

An example choice of ψ is the n^{th} derivative of a Gaussian, that is: $\psi(t) = \frac{d^n}{dt^n}G(t)$.

Parameter s in the wavelet ψ_s is known as the *scale* of the analysis. At small scales $s < 1$, the mother wavelet is *compressed*, so that only values close to $\mathbf{x}(t)$ influence the value of $X(s,t)$ (see Eq. (7)). Thus, at smaller scales, the wavelet coefficient $X(s,t)$ captures *local* variations of $\mathbf{x}$ around t, and these can be thought of as being the higher-frequency variations, i.e., variations that occur over a small amount of time. At larger scales $s > 1$, the mother wavelet is *dilated*, so that $X(s,t)$ is affected by values of $\mathbf{x}$ far from t as well. Thus, at larger scales, the wavelet coefficient captures low-frequency variations of $\mathbf{x}$, occurring over large periods of time. See [23, Table I] for some concrete scale-to-frequency mappings.

Fig. 17 shows a Normal Sinus Rhythm EGM and its CWT magnitude $|X(s,t)|$. Brighter colors indicate larger values of coefficient magnitudes $|X(s,t)|$. It is possible to see that early in the signal, mid- to low-frequency content is present (bright colors mid- to top of spectrogram), followed by higher-frequency variation (brighter colors at smaller scales), and near the end of the signal, two frequencies are present: mid-range frequencies (the bright colors near the middle), and very fast, low amplitude oscillations (the light blue near the bottom-right).

B COMPUTATION DIRECTION IN ROBUSTNESS MONITORING

Breach [5] computes the robustness signal $\rho_\phi(\mathbf{x}, t)$ by working from time 0 onwards (left to right) for some operators, and from time T backwards (right to left) for the others, as shown in this table:

Forward	Backward
$\neg$	$\Diamond$
$\wedge$	$\mathcal{U}$
$\Diamond_I$	

In fact, all computations can be performed in either direction. Indeed, $\neg$ and $\wedge$ are not temporal and so can be run in either direction - they just operate on the instantaneous value of the signal.

The timed Eventually operator, $\Diamond_I$, is evaluated by executing Lemire's running maximum algorithm on the sequence $(x(t_i))$ with window $t + I$. But the maximum of a sequence of m numbers $(a_1, \ldots, a_m)$ is the same as the maximum of $(a_m, \ldots, a_1)$, and an inspection of Lemire's algorithm can be executed backwards, by running the window from the end of the sequence to its beginning, yielding the same result and with the same complexity.

The untimed Eventually $\Diamond$ is evaluated on every two consecutive points $x(t_i), x(t_{i+1})$ by a case analysis. It is easy to see that this too can be computed forwards (in which case we incrementally compute the maximum over values from 0 to current time), again because the maximum of a sequence equals the maximum of the reversed sequence. (However note that we pay the price in memory as we must store the running maxima potentially for every t_i - if we only care about robustness at a specific point in time, then we only need to store one value).

Finally, the untimed Until operator is evaluated using a combination of $\wedge$, $\vee$ and untimed $\Diamond$, which we have shown can be evaluated in either direction.

C PROOF OF THM. 4.1

The proof is standard and proceeds by structural induction on the formula. The base cases $\phi = \top$ and $\mu \geq 0$ are immediate. The case $\phi \wedge \Psi$ is also standard and proceeds as follows: let $\delta = \|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty$. Let $a = \rho_\phi(\mathbf{x}, t)$, $b = \rho_\Psi(\mathbf{x}, t)$, $\tilde{a} = \rho_\phi(\tilde{\mathbf{x}}, t)$, and $\tilde{b} = \rho_\Psi(\tilde{\mathbf{x}}, t)$. Then $|\rho_{\phi\wedge\Psi}(\mathbf{x}, t) - \rho_{\phi\wedge\Psi}(\tilde{\mathbf{x}}, t)| = |a \sqcap b - \tilde{a} \sqcap \tilde{b}|$. By induction hypothesis, $|a - \tilde{a}| \leq L\delta$ and $|b - \tilde{b}| \leq L\delta$. Thus $a \sqcap b \leq (\tilde{a} \sqcap \tilde{b}) + L\delta$ and $a \sqcap b \geq (\tilde{a} \sqcap \tilde{b}) - L\delta$, and the conclusion follows. The Until case follows by applying the And and Or cases (the latter being proven similarly to AND).

D PROOF OF THM. 5.5

A set of zero Lebesgue measure is a *null set*. In what follows 'almost everywhere', meaning everywhere except possibly on a null set, is abbreviated as a.e.

1. By definition

$$\operatorname{ess\,sup}_{t\in(a,b)}|e(t)| = \inf\{M \mid |e(t)| \leq M \text{ a.e. in } (a,b)\}$$

Take an increasing sequence $M_i \nearrow M$. For every i there exists a non-null set $\Omega_i \subset (a,b)$ s.t. $|e(t)| > M_i$ a.e.; since $|e(t)| \leq M$ a.e. on Ω_i, then

$$M_i < e(t) \leq M \text{ a.e. on } \Omega_i \tag{8}$$

For every i, let Ω_i be the largest such set measure-wise. Note that Ω_i is a union of intervals, being non-null in $\mathbb{R}$. Since $M_1 < M_2$, $\Omega_2 \subset \Omega_1$, and in general, $\Omega_{i+1} \subset \Omega_i$. We can also take every Ω_i to be closed, since we can add a countable number of endpoints to any non-closed interval in Ω_i without changing its measure or the property that $|e(t)| > M_i$ a.e. on it. Thus every Ω_i is compact. Now every Ω_i is non-empty so define a sequence of points (t_i) by selection from Ω_i: $t_i \in \Omega_i$. This is an infinite sequence in Ω_1 which is compact and so it converges to a finite limit $t^* \in \Omega_1$.

From (8), it obtains that $M_i < \limsup_{t\to t_i}|e(t)| \leq M$, so as $i \to \infty$, $M_i \nearrow M$ and $t_i \to t^*$, and so $\limsup_{t\to t^*}|e(t)| = M$.

2. We will use a Hardy-type inequality which we now introduce. A *weight function* $w : \mathbb{R} \to \mathbb{R}$ is a measurable function positive a.e. in the interval (a,b) with $0 \leq a < b \leq \infty$. The w-weighted s-norm of function $f : (a,b) \to \mathbb{R}$, with $0 < s < \infty$ is:

$$\|f\|_{s,w} := \left(\int_a^b |f(t)|^s w(t)dt\right)^{1/s}. \tag{9}$$

The *Hardy operator* H is defined by $(Hf)(x) = \int_a^x f(t)dt$. Then, if f is non-negative, p, q satisfy $1 < p \leq q < \infty$, and u, v are weight functions on (a,b), the following two statements, known as a Hardy inequality, are equivalent [14]:

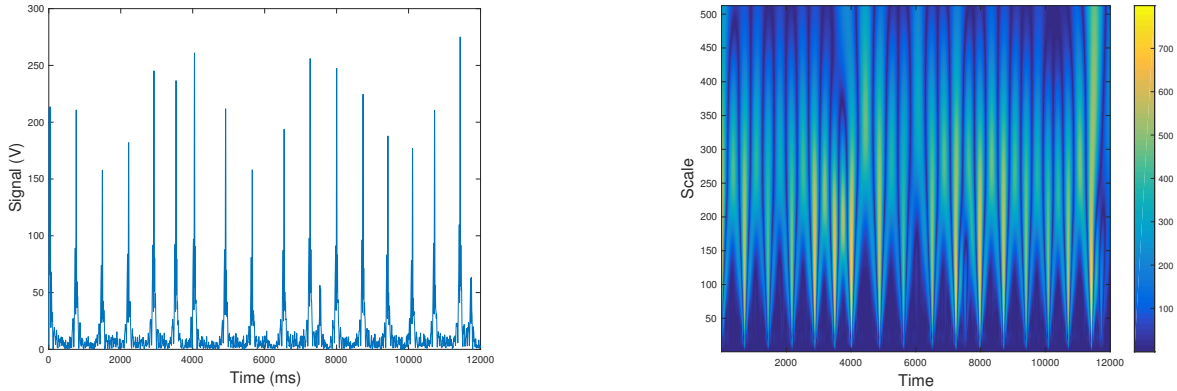


Figure 17: Rectified EGM during normal rhythm (left) and its CWT spectrogram (right)

- there exists a constant $C > 0$ s.t.

$$\|Hf\|_{q,u} \leq C \cdot \|f\|_{p,v}$$

- The following quantity is finite

$$A = \sup_{a<x<b} \left(\int_x^b u(t)dt\right)^{(1/q)} \left(\int_a^x v^{1-p'}(t)dt\right)^{1/p'} < \infty$$

where, $p' = \frac{p}{p-1}$

The proof proceeds as follows. Let t^* be as in part 1 of the theorem, and write $\overline{e} = \|e\|_\infty$. Assume that $\overline{e} > 0$ (otherwise the bound holds trivially). Since e is continuous at t^*, there exists $\varepsilon > 0$ s.t. $|e(t)| > 0$ over $t^* \pm \varepsilon$, so $\inf_{[t^*-\varepsilon, t^*+\varepsilon]} |e(t)| := \underline{e} > 0$. Write $t^- = t^* - \varepsilon, t^+ = t^* + \varepsilon$. We now assume without loss of generality that $0 < t^-$; if not, we can shift e to the right to make it so (possibly increasing b in the process), as this doesn't change the value of the norms we are computing. Therefore we have $0 \leq a \leq t^-$. Then there exists a constant $K > 0$ s.t. $e(t) \geq \underline{e} \geq K\overline{e}$ over $[t^-, t^+]$, and so

$$\int_{t^-}^x |e(t)|\, dt \geq K\overline{e}(x - t^-), \quad x \leq t^* + \varepsilon \tag{10}$$

Denote by χ_I the characteristic function of interval I. Take $f(t) = \chi_{[t^-,t^+]}(t)|e(t)|$: it is measurable and non-negative. Then

$$(Hf)(x) = \begin{cases} 0, & x < t^- \\ \int_{t^*-\varepsilon}^x |e(t)|dt, & t^- \leq x \leq t^+ \\ \text{a constant}, & x > t^+ \end{cases} \tag{11}$$

and by (10),

$$(Hf)(x) \begin{cases} = 0, & x < t^- \\ \geq K\overline{e}(x - t^* + \varepsilon), & t^- \leq x \leq t^+ \\ \geq 2\varepsilon K\overline{e}, & x > t^+ \end{cases} \tag{12}$$

So combining Eq. 9 and Eq. 12,

$$\begin{aligned} \|Hf\|_{q,u}^q &\geq \int_{t^-}^{t^+} K^q\overline{e}^q|x - t^-|^q u(x)dx + \int_{t^+}^b (2\varepsilon K)^q\overline{e}^q u(x)dx \\ &\geq \overline{e}^q \underbrace{\left(\int_{t^-}^{t^+} K^q|x - t^-|^q u(x)dx + \int_{t^+}^b (2\varepsilon K)^q u(x)dx\right)}_{B_\varepsilon} \\ &= B_\varepsilon \overline{e}^q \end{aligned}$$

Take v to be identically 1 over (a, b), $u(t) = e^{-t}$, and $p = q = 2$. Then

$$\begin{aligned} A &= \sup_{a<x<b} \left(\int_x^b e^{-t}dt\right)^{1/2} \left(\int_a^x 1dt\right)^{1/2} \\ &= \sup_{a<x<b} \sqrt{-e^{-b} + e^{-x}}\sqrt{x - a} < \infty \end{aligned}$$

Therefore, by Hardy's inequality, there exists a constant C s.t.

$$B_\varepsilon^{1/q}\overline{e} \leq \|Hf\|_{q,u} \leq C \cdot \|f\|_{p=2,v\equiv 1} \leq C\|e\|_2$$

Or, re-arranged,

$$\overline{e} = \|e\|_\infty \leq \frac{C}{B_\varepsilon^{1/2}}\|e\|_2$$

This completes the proof.

E COMPUTING THE BEST UNIFORM APPROXIMATION

Dunham's dynamic program [7] computes the uniform piece-wise linear approximation with the smallest number of segments given a maximum error ϵ. Let $F(u)$ be the minimal number of segments needed to approximate the data $(x(i))_{i\leq N}$ from $x(0)$ to $x(u)$ within ϵ. The essential observation is that F obeys the optimality principle $F(u) = \min_{v\in V(u)} 1 + F(v)$ where $V(u)$ is the set of all points $v \leq u$ such that the sequence $x(v), \ldots, x(u)$ can be approximated with one segment within error ϵ. A standard DP can then find the optimal solution.