AN ABSTRACT OF THE THESIS OF

Richard Pelphrey for the degree of Master of Science in Electrical and Computer Engineering presented on June 7, 2022.

Title: A Multiresolution Analysis of Temporal Logic

Abstract approved: ______________________________

Houssam Abbas

Is it possible to determine whether a signal violates a formula in Signal Temporal Logic (STL), if the monitor only has access to a low-resolution version of the signal? We answer this question affirmatively by demonstrating that temporal logic has a multiresolution structure, which parallels the multiresolution structure of signals. A formula in discrete-time Signal Temporal Logic (STL) is equivalently defined via the set of signals that satisfy it, known as its language. If a wavelet decomposition $x = y + d$ is performed on each signal x in the language, we end up with two signal sets Y and D, where Y contains the low-resolution approximation signals y, and D contains the detail signals d needed to reconstruct the x's. This thesis provides a complete computational characterization of both Y and D using a novel constraint set encoding of STL, s.t. x satisfies a formula if and only if its decomposition signals satisfy their respective encoding constraints. We obtain a sequence of lower-resolution formulas $\phi_{-1}, \phi_{-2}, \phi_{-3}, ...$ which thus constitute a multiresolution analysis of ϕ. This work lays the foundation for multiresolution monitoring in distributed systems. One potential application of these results is a multiresolution monitor that can detect specification violation early by simply observing a low-resolution version of the signal to be monitored. We illustrate these results with experiments on synthetic signals.

A Multiresolution Analysis of Temporal Logic

by

Richard Pelphrey

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 7, 2022
Commencement June 2022

Master of Science thesis of Richard Pelphrey presented on June 7, 2022.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Richard Pelphrey, Author

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Houssam Abbas for his experience and mentorship.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Algorithms

Chapter 1: Introduction

Consider a drone or drone swarm operating with the goal of carrying out some task, whether it be package delivery, building inspection, agricultural functions, art, or some other objective. During operation the drones would be sending data streams back to a central controller or monitor. This could include image/video, telemetry, or environmental data from an array of sensors. It may be that a human is simply watching the video feed from a drone under their control while an inspection is taking place. Or it could be that some central monitor is verifying all drones performing crop dusting services are spraying the correct fields and avoiding populated areas. Each of these tasks, and indeed many more, require the sending and receiving of signals.

Signals, whether continuous- or discrete-time, have a multi-resolution structure: a signal x in an appropriate signal space can be decomposed into the sum $x = x_{-1} + d_{-1}$ where x_{-1} is a low resolution approximation of x, and d_{-1} contains the detailed features that are missing from x_{-1}. By iterating this decomposition, we obtain a multiresolution sequence of approximations $x_{-1}, x_{-2}, \ldots, x_{-j}$, and so on. As $j \to \infty$, x_{-j} becomes lower and lower resolution. This is like walking away from a painting: as we walk further away from it, its details become blurred and the resolution of viewing decreases. This *multiresolution analysis*, obtained by wavelet decomposition, has been heavily leveraged in Signal Processing: for signal de-noising, enhancement, compression and decompression [24], and more recently to explain the capabilities of generative neural networks [17]. An example of a decomposition of an image is shown in Fig. 1.1 [18].

In most communication systems, the sender will first decompose its signal x using a wavelet basis. Rather than send the full-resolution x (which would require more bits), it sends first a low-resolution approximation x_{-J} (for some J). It then progressively sends more and more components of the signal so the receiver can progressively build higher and higher resolution approximations $x_{-J+1}, x_{-J+2}...$ until it has the full x. The advantage of such an incremental scheme is that if the connection drops at any point, the receiver has some version of the signal which is hopefully a good approximation of that signal and sufficient for its task.

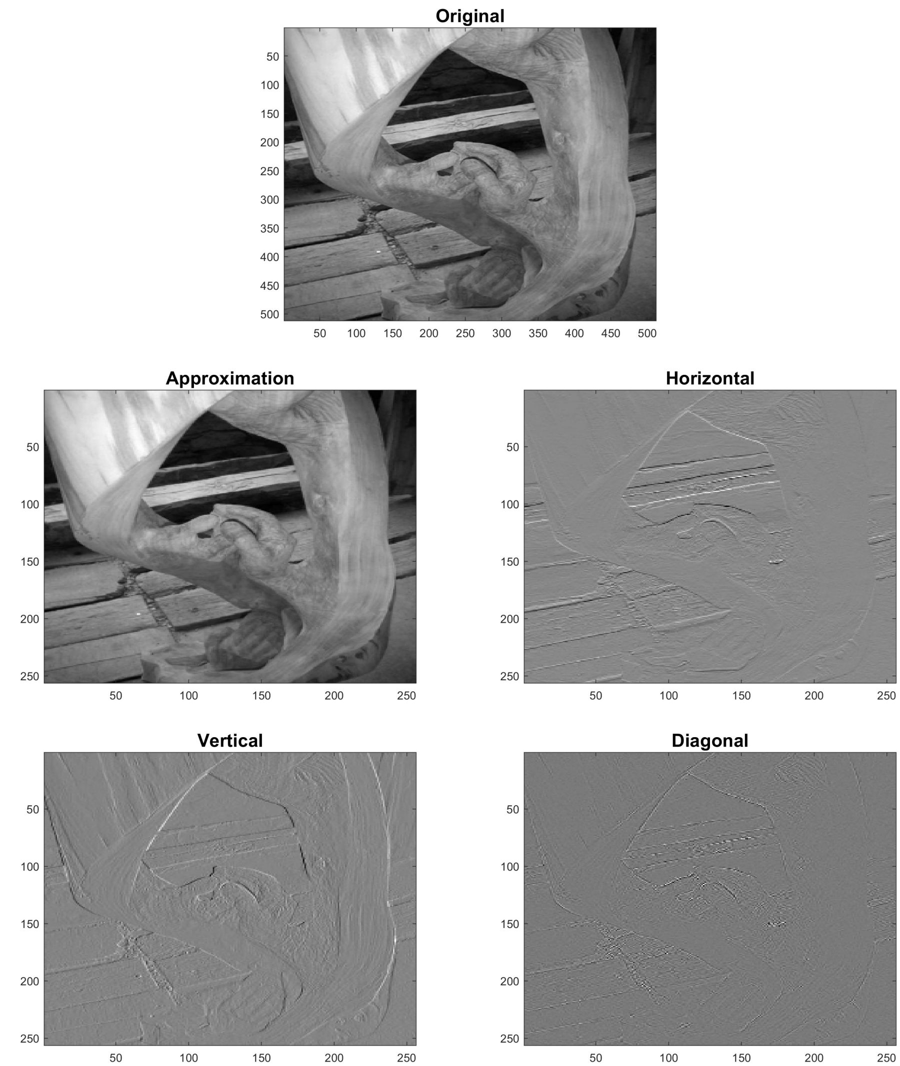

Figure 1.1: 2D Image Wavelet Decomposition

The task of interest in this thesis is runtime monitoring and it is motivated by distributed applications. An example of a distributed system is the drone swarm presented at the beginning of this section. In runtime monitoring of distributed systems, geographically dispersed nodes, such as the drones, send their signals to a monitoring node that determines whether the signals satisfy a specification in some temporal logic.

The field of Runtime Verification (RV) is concerned with verifying system behavior during operation using formal specifications. System behavior is typically defined in terms of some specification, such as "The car must not exceed 70 mph" or "The unmanned aerial vehicle must be at least 10 ft away from any structures at all times." Verifying these in real time hopefully allows for correction of undesirable or unsafe system states.

Up until now, Runtime Verification has not leveraged multiresolution approximations. We typically start with a signal in the time domain, and at most we assume the signal is missing samples [25, 15] or is corrupted by stochastic noise [23]. Issues of filtering, compression and de-compression, wireless transmission, and choice of representation, have not played a prominent role in Runtime Verification, despite the fact that they affect any signal we can practically measure, and therefore pose an a priori problem to the accuracy of runtime monitoring. Because these operations rely heavily on non-time-domain representations, most notably the Fourier and Wavelet domains, we must integrate the languages of signal processing and runtime verification to arrive at a consistent and practical formalism in which to pose and solve these problems. This paper focuses on the wavelet-based multiresolution analysis, and its implications for runtime verification.

Specifically, suppose the receiver has a formula ϕ in discrete-time Signal Temporal Logic (STL), which formalizes a system specification. Its task is to determine whether the received signal x violates it. A number of questions naturally arises when we start thinking about the multiresolution structure of x. First, if the monitor only has access to a lower resolution approximation x_{-j}, can it still determine whether x violates ϕ? More generally, if $\mathcal{L}(\phi)$ is the set of signals that satisfy ϕ (known as the *language* of ϕ) is there a computational characterization of its approximation $\mathcal{L}_{-j}$ at scale $-j$? In particular, is there a *logical* characterization of $\mathcal{L}_{-j}$ - i.e. can the lower-resolution set of signals $\mathcal{L}_{-j}$ itself be approximated as the language of some formula?

This thesis gives positive answers to the above questions: it is possible to soundly detect a violation from coarse approximations of x; the low-resolution approximation $\mathcal{L}_{-j}$ of the language of ϕ can be encoded with mixed-integer constraints obeying certain properties; and $\mathcal{L}_{-j}$ is over-approximated by the language of a certain formula which we derive.

These answers have important implications for runtime verification: first, the multiresolution analysis $(\mathcal{L}_{-j})_{j\in\mathbb{N}}$ serves as a new concept of approximation for temporal logic. This notion of approximation is more refined than classical entailment, does not require user input, and is a priori distinct from weakness [7] and entropy [5]. We hope to leverage this for more efficient monitoring and control synthesis. The logical characterization of approximate languages will also allow a tighter integration of runtime monitoring in signal processing chains, as it allows us to interpret, in the same logical formalism, the effects of certain wavelet-domain operations. In particular, many feature extractors, including certain deep neural networks, use wavelet domain features, and it would be interesting to provide an interpretation of these features in logic.

In this thesis, defined is an immediate application of the result. In this application, we show how a monitor which is receiving successive approximations x_{-j} of x, can monitor each approximation against the formula at scale $-j$, and determine early (before receiving the entire signal) whether x violates ϕ.

The remainder of this work is as follows: chapter 2 provides a brief literature review of works that pertain to temporal logic and signal processing techniques, such as frequency analysis. Chapter 3 provides the necessary technical background on multiresolution analysis and STL. Chapter 4 contains the central contributions of this work. An application for low resolution monitoring is described in Chapter 5, after which experiments in monitoring are run on randomly generated signals. Lastly, Chapter 6 concludes the work with some final remarks and potential future work.

The appendices provide proofs for the theorems and lemma found in Chapter 4.

Chapter 2: Literature Review

Signal Temporal Logic (STL) [16], like other temporal logics, provides for unambiguous formal specifications which can be used to verify system behavior. It concerns itself only with temporal properties and the authors of this work are unaware of any work that applies STL to low-resolution signals.

There are works that consider run-time verification, using temporal logic formulas, of signals in which some samples are lost. The work [25] uses a Hidden Markov Model to determine the probability of satisfaction. The authors use the Hidden Markov Model to estimate the values of the lost samples, and therefore make a determination about the probability of satisfaction. The work [15] does not use state estimation, but considers directly whether a formula can be monitored if samples are lost. Crucially, it is concerned with only transient loss. The authors introduce an algorithm, which runs offline, to make this determination. Both of these examples consider chiefly the result of monitoring with missing samples, rather than monitoring low-resolution signals.

Some works do consider frequency-domain characterization of signals, but consider a new signal constructed using frequency information over time, rather than a low-resolution version of the signal. In [10] the authors introduce a variant of STL, called Time-Frequency Logic (TFL), to explicitly specify properties of the frequency spectrum of the signal - e.g. one formula might read "$|X(\omega, t)| > 5$ over the next three time units" (where $X(\omega, t)$ is obtained by a windowed Fourier transform). In [20], abnormal behaviors of signals are investigated using what the authors call parametric time-frequency logic (PTFL). PTFL is similar to TFL, with the exception that instead of applying a windowed Fourier transform, the authors utilize the continuous wavelet transform and use the resulting coefficients to create a signal containing both spectral and temporal information. Similar to [10], the work does not consider monitoring low-resolution signals, nor was that the intention of the work. The works [22] and [14] provide a general algebraic framework for the semantics of temporal logic based on semi-rings, opening the way to producing new semantics automatically by concretizing the semi-ring operations. The work [22], in particular, provides a direct mapping from temporal logic, specifically

Metric Temporal Logic and Linear Temporal Logic, to time-invariant filters, a very common signal processing technique. The recent [6] develops a systematic frequency analysis of temporal logic robustness, and demonstrates its use to design linear filters that can clean a signal without negatively impacting the accuracy of a temporal logic monitor.

Papers [2, 3] apply the continuous wavelet transform to the problem to detecting irregularities in heartbeats through peak detection. The authors craft peak detection algorithms using the CWT and apply this to a formal language, although not a temporal logic, which allows for efficient expression of numerical queries, such as counting, over data streams. In [1], the authors demonstrate how different representation and signal reconstruction schemes can impact Runtime Verification. Specifically, they demonstrate that the robustness, a measure of how well a signal satisfies a specification or by how much it violates it, computation can be negatively impacted by this choice.

Chapter 3: Technical Background

Notation. Let $\mathbb{R} = (-\infty, \infty), \mathbb{N} = \{0, 1, 2, \ldots\}$ and $\mathbb{Z} = \mathbb{N} \cup -\mathbb{N}$. With two integers a and b, $[a : b] = \{a, a+1, \ldots, b\} \subset \mathbb{Z}$. Given an integer interval $I \subset \mathbb{N}$ and $t \in \mathbb{N}$, $t + I := \{t' \mid \exists s \in I. t' = t + s\}$. Given a real interval $[a, b]$ and scalar c, $c[a, b]$ equals $[ca, cb]$ if $c \geq 0$, otherwise it equals $[cb, ca]$. Interval addition is defined the natural way: $[a, b] + [c, d] = [a + c, b + d]$.

The inner product between two elements x and y of an inner product space is written $\langle x, y \rangle$. E.g. in $\mathbb{R}^N$, $\langle x, y \rangle = \sum_n x(n)y(n)$. Suppose X is an inner product space and U, V are orthogonal subspaces of X (i.e., for every $u \in U$ and $v \in V$, $\langle u, v \rangle = 0$). Then the *direct sum* of U and V is the set $U \oplus V := \{u + v \mid u \in U, v \in V\}$. Note orthogonal subspaces only intersect at 0. A *basis* for an N-dimensional vector space X is a set of N independent vectors that span X. A basis is *orthonormal* if its vectors have unit norm and are pairwise orthogonal. The shift operator R_i on $\mathbb{R}^N$ is defined by $R_i x = (x(i), x(i+1), \ldots, x(N-1), x(0), \ldots, x(i-1))$. The upsample operator, U is defined by $Ux = (x(0), 0, x(1), 0, \ldots, x(N), 0)$. The convolution, denoted with an $*$, is $x * w(n) = \sum_{n=0}^{N-1} x(m-n)w(n)$.

A *boolean variable* takes values in $\{0, 1\}$.

3.1 Wavelets and Multiresolution Analysis in $\ell^2(\mathbb{Z}_N)$

All the material in this section is drawn from [11]. To maintain clarity of exposition, we develop this thesis' theory in what is perhaps the simplest setting allowing a multiresolution analysis. Namely, our signals are elements of $\ell^2(\mathbb{Z}_N)$, the set of periodic real sequences with period N

$$\begin{aligned} \ell^2(\mathbb{Z}_N) := \{z = (\ldots, z(0), z(1), z(2), \ldots, z(N-1), \ldots) \mid \\ z(j) \in \mathbb{R} \text{ and } z(j + kN) = z(j) \ \forall k \in \mathbb{Z}, 0 \leq j \leq N-1\} \end{aligned}$$

Thus if $z \in \ell^2(\mathbb{Z}_{20})$ then $z(-5) = z(15) = z(35)$. Such a sequence can be viewed as the result of uniform sampling of a periodic continuous-time signal. If the signals in a given application are not periodic we can take N sufficiently large so that the boundary values do not matter.

Because z is N-periodic, only its values over $0, 1, \ldots, N-1$ need to be looked at. So in what follows we treat an element of $\ell^2(\mathbb{Z}_N)$ as being an N-dimensional vector in $\mathbb{R}^N$ - this is a standard representation of periodic sequences in signal processing. (Formally, one can regard z as defined on the equivalence classes of $\mathbb{Z}$ mod N). As such, $\ell^2(\mathbb{Z}_N)$ is a normed vector space with the usual inner product: with $u, v \in \ell^2(\mathbb{Z}_N)$, $\langle u, v\rangle := \sum_{n=0}^{N-1} u(n)v(n)$.

We are interested in a signal decomposition that separates high-resolution signal details from low-resolution coarse features. Such a decomposition is captured mathematically through the concept of a *Multiresolution Analysis*, or MRA [11].

We consider a p^{th}-stage wavelet basis of $\ell^2(\mathbb{Z}_N)$. Wavelet basis are constructed by first considering two 1^{st}-level wavelet filters, $u_1,\ v_1 \in \ell^2(\mathbb{Z}_N)$. u_1 is primarily responsible for retrieving the approximation of a signal, while v_1 retrieves the details, or higher frequency components, of that signal. There are many filter systems known, such as the Haar and Daubechies' order 6 filter systems, both of which we make use of here and are covered in more detail in [11]. Fig. 3.1 shows time-centered u_1 and v_1 for both of these wavelets.

Under suitable conditions, u_1 and v_1 are already enough to construct a first stage wavelet basis.

$$B_1 = \{R_{2k}u_1\}_{k=0}^{N/2-1} \cup \{R_{2k}v_1\}_{k=0}^{N/2-1} \tag{3.1}$$

B_1 is orthonormal, so it follows that the sub-spaces

$$V_{-1} = span\{R_{2k}u_1\}_{k=0}^{N/2-1} \qquad W_{-1} = span\{R_{2k}v_1\}_{k=0}^{N/2-1}$$

are orthogonal and add back up to the full space: $\ell^2(\mathbb{Z}_N) = V_{-1} \oplus W_{-1}$. Thus every signal $x \in \ell^2(\mathbb{Z}_N)$ can be written as the sum of its orthogonal projections onto V_{-1} and W_{-1}:

$$x = \mathrm{Proj}(x, V_{-1}) + \mathrm{Proj}(x, W_{-1}) := x_{-1} + d_{-1} \tag{3.2}$$

By convention we set $x_0 := x$. An orthogonal projection is easily computed by projecting

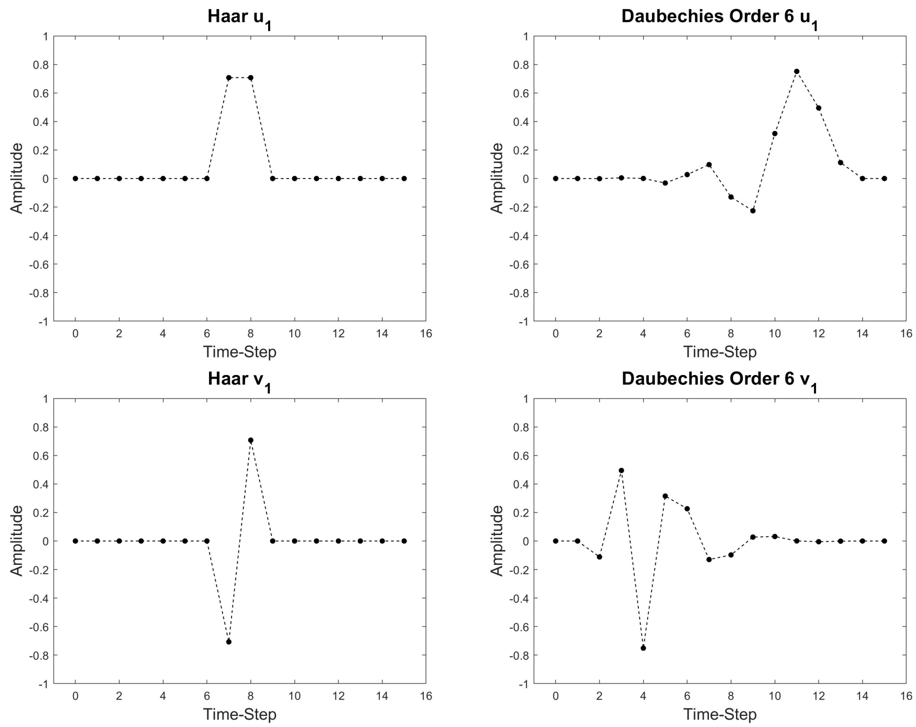

Figure 3.1: u_1 and v_1 for Haar and Daubechies' Order 6.

onto the basis vectors. For example, the projection of x onto W_{-1} is given by:

$$\mathrm{Proj}(x, W_{-1}) = \sum_{k=0}^{N/2-1} \langle z, R_{2k}v_1 \rangle R_{2k}v_1 \tag{3.3}$$

The inner products $\langle z, R_{2k}v_1 \rangle$ are known as the wavelet coefficients. The projection operator is extended to sets of signals in the natural way: $\mathrm{Proj}(S, W_{-1}) = \{\mathrm{Proj}(x, W_{-1}) \mid x \in S\}$.

We can further iterate this decomposition to achieve lower levels of resolution by considering filters $u_\ell, v_\ell \in \ell^2(\mathbb{Z}_{N/2^{\ell-1}})$ for higher levels of decomposition. These are constructed from u_1 and v_1 in the following manner:

$$u_\ell[n] := \sum_{k=0}^{2^{\ell-1}-1} u_1 \left[n + \frac{kN}{2^{\ell-1}} \right]$$

and similar for v_ℓ. Next it is necessary to define

$$f_\ell := g_{\ell-1} * U^{\ell-1}(v_\ell) \ ; \ g_\ell := g_{\ell-1} * U^{\ell-1}(u_\ell)$$

We also set $f_1 = v_1$, and $g_1 = u_1$. This then allows us to use the standard notation for the elements of a wavelet basis

$$\varphi_{-j,k} := R_{2^j k} g_j \ ; \ \psi_{-j,k} := R_{2^j k} f_j \tag{3.4}$$

in which the resulting $\varphi_{-j,k}$ and $\psi_{-j,k}$ are orthonormal at each scale j. This allows us to iterate this decomposition splitting $V_{-1} = V_{-2} \oplus W_{-2}$ using a 2^{nd}-stage wavelet basis:

$$\begin{aligned} B_2 &= \{\psi_{-2,k}\}_{k=0}^{N/2^2-1} \cup \{\varphi_{-2,k}\}_{k=0}^{N/2^2-1} \\ V_{-2} &= span\{\phi_{-2,k}\}_{k=0}^{N/2^2-1} \qquad W_{-2} = span\{\psi_{-2,k}\}_{k=0}^{N/2^2-1} \end{aligned}$$

Thus $x = d_{-1} + x_{-1} = d_{-1} + d_{-2} + x_{-2}$. After p iterations, we find the following full p^{th}-stage basis:

$$B_p = \{\varphi_{-p,k}\}_{k=0}^{N/2^p-1} \cup \{\psi_{-p,k}\}_{k=0}^{N/2^p-1} \cup \cdots \cup \{\psi_{-1,k}\}_{k=0}^{N/2-1} \tag{3.5}$$

and MRA of $\ell^2(\mathbb{Z}_N)$:

$$\ell^2(\mathbb{Z}_N) = W_{-1} \oplus W_{-2} \oplus \ldots \oplus W_{-p} \oplus V_{-p} \tag{3.6}$$

where, $V_{-j} = span\{\phi_{-j,k}\}_{k=0}^{N/2^j-1}$, $W_{-j} = span\{\psi_{-j,k}\}_{k=0}^{N/2^j-1}$, and $V_{-j} = V_{-(j+1)} \oplus W_{-(j+1)}$. It holds that

$$\ldots \subset V_{-(j+1)} \subset V_{-j} \subset \ldots V_{-1} \subset \ell^2(\mathbb{Z}_N) \tag{3.7}$$

This MRA is succinctly visualized in Figure 3.2, where arrows indicate how subspaces add up.

The negative index $-j$ is known as the *scale* of the subspaces. (Negative indices are used as in [11] for consistency of notation with the continuous-time case).

$$
\begin{array}{l}
\ldots \longrightarrow V_{-3} \longrightarrow V_{-2} \longrightarrow V_{-1} \longrightarrow \ell^2(\mathbb{Z}_N) \\
\ldots \nearrow W_{-3} \nearrow W_{-2} \nearrow W_{-1} \nearrow
\end{array}
$$

Figure 3.2: MRA of $\ell^2(\mathbb{Z}_N)$

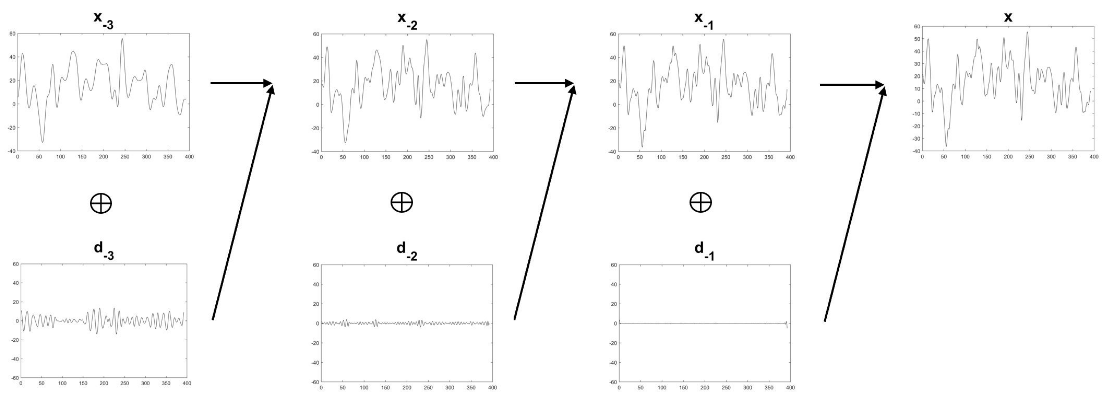

Figure 3.3: MRA of signal x [19].

So far we haven't said much about wavelets, or the basis B_1 in Eq. (3.1). The basis is such that x_{-1} contains the coarse (or low-resolution) features of x, while d_{-1} contains its detailed features. More generally, B_j is such that x_{-j-1} contains the coarse features of x_{-j}, while d_{-j-1} contains its detailed features. This is achieved by having basis vectors that are well-localized in both the time and Fourier domains, i.e. such that $|h(n)| << 1$ and $|\widehat{h}(\omega)| << 1$ for all but a few positions n and ω, for all vectors $h \in B_j$. (Here, $\widehat{h}$ is the Fourier transform of h). An example of this decomposition is provided in Fig. 3.3 to scale -3 using the Daubechies wavelet.

To finish off the wavelet background, we now quickly construct a first stage basis using the discrete Haar filters in $\ell^2(\mathbb{Z}_N)$ and apply the basis to a simple signal. If we take N=4 for simplicity, we have, at first level, $u_1 = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0]^T$ and $v_1 = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, 0]^T$.

Then, if we recognize, from Eqn. 3.4, that $\psi_{-1,k}$ and $\varphi_{-1,k}$ are simply rotations by 2 of v_1 and u_1 respectively, V_{-1} and W_{-1} are simply,

$$V_{-1} = span\left\{\begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\\0\\0\end{bmatrix}\begin{bmatrix}0\\0\\\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{bmatrix}\right\} \quad W_{-1} = span\left\{\begin{bmatrix}\frac{1}{\sqrt{2}}\\-\frac{1}{\sqrt{2}}\\0\\0\end{bmatrix}\begin{bmatrix}0\\0\\\frac{1}{\sqrt{2}}\\-\frac{1}{\sqrt{2}}\end{bmatrix}\right\}$$

Which creates a full orthonormal basis for $\ell^2(\mathbb{Z}_4)$. Now we want to find the first level decomposition of the toy signal

$$x = [0, 1, 1, 0]^T$$

First we find the approximation of the signal, or the projection onto V_{-1}. A quick calculation finds that

$$x_{-1} = [0.5, 0.5, 0.5, 0.5]^T$$

and the details resulting from projecting onto W_{-1} are then

$$d_{-1} = [-0.5, 0.5, 0.5, -0.5]^T$$

And then a quick sanity check reveals that

$$x_{-1} + d_{-1} = [0, 1, 1, 0]^T = x$$

So we have now deconstructed our signal, and then obtained the original signal by adding the approximation and the details together. This is a simple toy problem, but illustrates the basic idea behind a wavelet decomposition well.

3.2 Signal Temporal Logic over Finite Discrete-Time Signals

Signal Temporal Logic (STL) [16, 9] is a logic that allows the succinct and unambiguous specification of a wide variety of desired system behaviors over time, such as "The vehicle reaches its destination within 10 time units while always avoiding obstacles" and "While the vehicle is in Zone 1, it must obey that zone's velocity constraints." STL is defined over continuous-time signals.

We use a variant of STL which applies to discrete-time signals, such as the elements of $\ell^2(\mathbb{Z}_N)$. For simplicity in this thesis we work with scalar-valued signals. Formally, let $X \subset \mathbb{R}$ be the state-space. A *signal* x is an element of $\ell^2(\mathbb{Z}_N)$ s.t. $x(j) \in X$ for all j. We will write $\ell^2(\mathbb{Z}_N \to X)$ for the signal space. Let $\{\mu_1, \ldots, \mu_L\}$ be a set of real-valued functions of the state: $\mu_\ell : X \to \mathbb{R}$. Let $AP = \{p_1, \ldots, p_L\}$ be a set of atomic propositions.

Definition 1 (Discrete-time STL (DT-STL)) *The syntax of the logic is given by*

$$\phi := \top \mid p \mid \neg p \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathcal{U}_I \, \phi_2 \mid \phi_1 \mathcal{R}_I \phi_2$$

where $p \in AP$ and $I \subseteq \mathbb{N}$ is an integer interval. The semantics are given relative to signals as follows.

$$\begin{aligned}
(x,t) &\models \top \\
(x,t) &\models p_\ell \text{ iff } \mu_\ell(x(t)) \geq 0 \\
(x,t) &\models \neg p_\ell \text{ iff } (x,t) \not\models p_\ell \\
(x,t) &\models \phi_1 \wedge \phi_2 \text{ iff } (x,t) \models \phi_1 \text{ and } (x,t) \models \phi_2 \\
(x,t) &\models \phi_1 \vee \phi_2 \text{ iff } (x,t) \models \phi_1 \text{ or } (x,t) \models \phi_2 \\
(x,t) &\models \phi_1 \mathcal{U}_I \phi_2 \text{ iff } \exists t' \in t + I \;.\; (x,t') \models \phi_2 \text{ and} \\
&\quad \forall t'' \in [t : t'-1], \;\; (x,t'') \models \phi_1 \\
(x,t) &\models \phi_1 \mathcal{R}_I \phi_2 \text{ iff EITHER } \forall t' \in t + I \;.\; (x,t') \models \phi_2 \text{ OR} \\
&\quad \exists t' \in t + I.\; (x,t') \not\models \phi_2 \text{ and } \exists t'' \in [t : t'),\; (x,t'') \models \phi_1
\end{aligned}$$

The language of ϕ *is* $\mathcal{L}(\phi) := \{x \in \ell^2(\mathbb{Z}_N \to X) \mid (x,0) \models \phi\}$.

The Eventually and Always operators are derived from Until. Because we will often use them, we give their semantics explicitly. For eventually we have

$$(x,t) \models \Diamond_I \, \phi \text{ iff } \exists t' \in t + I \;.\; (x,t') \models \phi$$

The Always is given by $\Box_I \, \phi := \neg \Diamond_I \neg \phi$ and it's semantics are

$$(x,t) \models \Box_I \, \phi \text{ iff } \forall t' \in t + I \;.\; (x,t') \models \phi$$

We use the shorthand notation $\Diamond_n$ for $\Diamond_{[n,n]}$; when $n = 0$ then $\Diamond_0 \, \phi = \phi$.

We make the following assumptions. The state space X is bounded. The functions μ_ℓ are continuous and have bounded variation [13]. $\mu_\ell : [a, b] \rightarrow \mathbb{R}$ is of bounded variation iff $\sum_{i=0}^{n} |\mu_\ell(x_i) - \mu_\ell(x_{i-1})| \leq M$ for some $M > 0$ and for all $a < x_1 < x_2 < \ldots <_n < x_b$ [12].

Because of the periodicity of signals in $\ell^2(\mathbb{Z}_N \rightarrow X)$, an unbounded formula (with unbounded temporal intervals I) can always be re-expressed as a boolean combination of bounded formulas that constrain one period $(x(0), \ldots, x(N-1))$. So without loss of generality, we may consider that all formulas are bounded and constrain one period.

Discrete-time STL has been used extensively in control synthesis such as in [21], although it is usually referred to as (plain) STL in those papers. STL, however, is a logic over continuous-time or time-stamped sequences. Because in future work we will extend our MRA to continuous-time STL, in this thesis we explicitly say 'DT-STL'.

In what follows we will make extensive use of the following construct. Recall the definitions of spaces V_{-j} and W_{-j} from Section 3.1.

Definition 2 (Projected Language) *Given a DT-STL formula ϕ and an MRA of $\ell^2(\mathbb{Z}_N)$, define $\mathcal{L}_{-j} := Proj(\mathcal{L}(\phi), V_{-j})$ and $\mathcal{K}_{-j} := Proj(\mathcal{L}(\phi), W_{-j})$. Thus $\mathcal{L}(\phi) = \mathcal{L}_{-j} \oplus \mathcal{K}_{-j}$.*

In words, $\mathcal{L}_{-j}$ contains the scale-$(-j)$ coarse approximations of signals that satisfy ϕ, and $\mathcal{K}_{-j}$ contains the corresponding details.

Chapter 4: Multiresolution Analysis of a DT-STL Formula

Recall the three questions posed in the Introduction: If the monitor only has access to a lower resolution approximation x_{-j} of x, is it still possible to determine whether x violates ϕ? If $\mathcal{L}(\phi)$ is the language of ϕ, is there a computational characterization of its projection $\mathcal{L}_{-j}$ onto V_{-j}? In particular, can the lower-resolution set of signals $\mathcal{L}_{-j}$ itself be approximated as the language of some formula ϕ_{-j}? We answer these three questions in the following sections.

4.1 Logical Characterization of Projected Languages

Theorem 1 (Approximate MRA of DT-STL) *Fix an MRA of $\ell^2(\mathbb{Z}_N)$, and let $x \in \ell^2(\mathbb{Z}_N \to X)$ be a signal with decomposition $x = x_{-J} + d_{-J} + \ldots + d_{-1}$ in the given MRA. Let ϕ be a DT-STL formula. Then there exist DT-STL formulas ϕ_{-j} and δ_{-j}, given by Algorithm 1, s.t. $x \models \phi$ only if $x_{-j} \models \phi_{-j}$ and $d_{-j} \models \delta_{-j}$.*

The proof is in the Appendix. Thus application of the theorem yields an approximate MRA $(\phi_{-j})_j$ of ϕ.

The reason we call this an 'approximate' MRA is that the implication is one-sided: we can go from φ to φ_{-j} but not the other way around. In the next section we show a different characterization (using mixed-integer constraints) that gives a two-sided implication.

This theorem provides an answer to the first question above: it is possible to soundly detect a violation from coarse approximations of x. Namely, x_{-j} is monitored against ϕ_{-j}. By Thm. 1, if there exists j s.t. $x_{-j} \not\models \phi_{-j}$ or $d_{-j} \not\models \delta_{-j}$, then $x \not\models \phi$.

Secondly, the scale-$(-j)$ approximation of $\mathcal{L}(\phi)$, $\mathcal{L}_{-j}$, does have a logical characterization. Indeed, suppose $y \in \mathcal{L}_{-j}$. Then there exists an $x \in \mathcal{L}(\phi)$ s.t. $y = \mathrm{Proj}(x, V_{-j})$, and by the theorem, $y \models \phi_{-j}$. Thus $\mathcal{L}_{-j} \subseteq \mathcal{L}(\phi_{-j})$.

We now turn our attention to the algorithm MRA-DT-STL, which computes ϕ_{-j}, δ_{-j} and is recursive over the structure of the formula. In brief, the algorithm takes in V_{-j}, W_{-j}, a formula ϕ, a position n in which the ϕ is applied to, and state bounds. It then

Algorithm 1: MRA-DT-STL.

Data: A DT-STL formula ϕ, a j^{th}-level MRA V_{-j} and W_{-j}, orthonormal bases $\{\varphi_{-j,k}\}_{k=0}^{N/2-1}$ of V_{-j} and $\{\psi_{-j,k}\}_{k=0}^{N/2-1}$ of W_{-j}, state-space bounds $X = [-a, a]$ (i.e. for any signal, $|x(m)| \leq a$ for all m), a position n which ϕ is applied to

Result: (ϕ_{-j}, δ_{-j}): an ordered pair of DT-STL formulas s.t. $\text{Proj}(\mathcal{L}(\phi), V_{-j}) \subseteq \mathcal{L}(\phi_{-j})$ and $\text{Proj}(\mathcal{L}(\phi), W_{-j}) \subseteq \mathcal{L}(\delta_{-j})$

`// Base case 1`

1 **if** $\phi = \top$ **then**

3   Set $\phi_{-j} = \top$ and $\delta_{-j} = \top$. Return (ϕ_{-j}, δ_{-j})

`// Base case 2`

4 **else if** *$\phi = p$ for some atomic proposition $p = (\mu(x) \geq 0)$* **then**

5   Let $v_{m,\ell} := \sum_k \varphi_{-j,k}(\ell)\varphi_{-j,k}(m)$ and $w_{m,\ell} := \sum_k \psi_{-j,k}(\ell)\psi_{-j,k}(m)$ for $0 \leq \ell, m \leq N-1$

7   Express $\mathcal{L}(p) = \{x \in [-a, a] \mid \mu(x) \geq 0\}$ as the union of disjoint intervals $S^i, i = 1 \dots K$

9   Set $h = n \bmod 2^j$

10   **for** $i = 1 \dots K$ **do**

11   **for** $0 \leq m \leq N-1$ **do**

13   Let $S^i_{-j}[m+h, h] := v_{m+h,h} S^i + \sum_{\ell \neq h} v_{m+h,\ell}[-a, a]$

15   Let $D^i_{-j}[m+h, h] := w_{m+h,h} S^i + \sum_{\ell \neq h} w_{m+h,\ell}[-a, a]$

17   Define atomic proposition $s_{m+h,h} := x \in S^i_{-j}[m+h, h]$

19   Define atomic proposition $q_{m+h,h} := x \in D^i_{-j}[m+h, h]$

20   **end**

22   Set $\phi^i_{-j} = \wedge_{0 \leq m < N} \Diamond_m s_{m+h,h}$ and $\delta^i_{-j} = \wedge_{0 \leq m < N} \Diamond_m q_{m+h,h}$

23   **end**

25   Set $\phi_{-j} = \bigvee_i \phi^i_{-j}$ and $\delta_{-j} = \bigvee_i \delta^i_{-j}$. Return (ϕ_{-j}, δ_{-j})

27 **else if** $\phi = \neg p$ **then**

    `// Similar to case` $\phi = p$ `above`

29 **else if** $\phi = \Diamond_I \eta$ **then**

31   **for** $i = 0 \dots 2^j - 1$ **do**

33   Set $(\eta_{-j}(i), \alpha_{-j}(i)) = \text{MRA-DT-STL}(\eta, n = n + i)$`// Other inputs unchanged`

35   Set $I(i) = \{m \in I | m = i + 2^j y, y \in \mathbb{N}\}$

36   **end**

37   Return $\phi_{-j} = \bigvee_{0 \leq i < 2^j} \Diamond_{I(i)} \eta_{-j}(i)$ and $\delta_{-j} = \bigvee_{0 \leq i < 2^j} \Diamond_{I(i)} \eta_{-j}(i)$

39 **else if** $\phi = \Box_I \eta$ **then**

    `// Lines 31 - 35`

41   Return $\phi_{-j} = \bigwedge_{0 \leq i < 2^j} \Box_{I(i)} \eta_{-j}(i)$ and $\delta_{-j} = \bigwedge_{0 \leq i < 2^j} \Box_{I(i)} \eta_{-j}(i)$

43 **else if** *$\phi = \eta \ Op \ \xi$ for DT-STL formulas η, ξ and $Op \in \{\wedge, \vee\}$* **then**

44   Set $(\eta_{-j}, \alpha_{-j}) = \text{MRA-DT-STL}(\eta, n)$

45   Set $(\xi_{-j}, \beta_{-j}) = \text{MRA-DT-STL}(\xi, n)$

46   Return $\phi_{-j} = \eta_{-j} \ Op \ \xi_{-j}$ and $\delta_{-j} = \alpha_{-j} \ Op \ \beta_{-j}$

walks down the structure of the formula, starting from the outside and working towards the proposition, recursively returning the correct $\phi_{-j}(n)$ and $\delta_{-j}(n)$. In this manner it returns the scale$-j$ low-resolution formulas. The proof of Thm. 1 will establish the algorithm's correctness. Some remarks are in order:

1. The algorithm gives the scale-(-j) decomposition of the formula. It must be repeated for $j = 1 \dots J$ to build the full MRA of ϕ.

2. Line 7 simply expresses the fact that the set $\mathcal{L}(p)$ is necessarily the finite union of disjoint bounded closed intervals (see Appendix A).

3. Lines 13 and 15 use interval arithmetic: given an interval $[a, b]$ and scalar v, the set $v[a, b]$ equals $[va, vb]$ if $v \geq 0$ and $[vb, va]$ otherwise. Further, $[a, b] + [c, d] = [a + c, b + d]$.

4. When ϕ is an atomic proposition p, its scale-(-j) approximation p_{-j} (line 25) is *not* an atomic proposition - i.e. it is not a constraint on $x_{-j}(n)$ alone. A priori, p_{-j} constrains *every* time step $x_{-1}(m)$, as can be seen on Line 22. This is a result of the coarsening (or averaging) that happens when going down in scale $-j$. The extent to which this happens depends on how concentrated in time are the functions $\psi_{-j,k}$ and $\varphi_{-j,k}$: the more concentrated, the less the effect of constraint $x(n) \in S^i$ on far-away time steps $|n| >> 0$. Wavelets $\psi_{-j,k}$ are designed to be concentrated around $n = 2^j k$, which gives a handle on how quickly the effect of the constraint disappears.

5. Also in the case $\phi = p$, the formulas ϕ_{-j} and δ_{-j} are built from the new atoms $s_{m,n}, q_{m,n}$, not the atoms of ϕ (lines 17-19).

6. It may be helpful to think of ϕ^i_{-j} as an 'extended atomic proposition', in the sense that it is the most basic constraint induced by ϕ on the states of x_{-j}. Similarly for δ^i_{-j}.

7. In Lines 29-41 we recursively call MRA-DT-STL on each sub-formula of ϕ, each time setting $n = n + i$ as shown in Line 33. This reflects the need to place the extended proposition at the correct location and therefore the dependence of the sub-formulas on the positions to which they are being applied, as is shown in Appendix A.

This does establish that there is some multiresolution structure to DT-STL formulas. Meaning that given $x_{-j} \models \phi_{-j}$, we *can* find a ϕ_{-j-1} such that $x_{-j-1} \models \phi_{-j-1}$ and $d_{-j-1} \models \delta_{-j-1}$. We may visualize this idea with the following diagram, which should be compared to Fig. 3.2:

$$\begin{array}{ccccccccc} \ldots \longleftarrow & \phi_{-3} \longleftarrow & & \phi_{-2} \longleftarrow & & \phi_{-1} \longleftarrow & \phi \\ \ldots \swarrow & \delta_{-3} & \swarrow & \delta_{-2} & \swarrow & \delta_{-1} \swarrow & \end{array} \tag{4.1}$$

The direction of the arrows, right to left, indicates that given ϕ_{-j} we can obtain its lower-scale decomposition. However, we are not yet able to combine ϕ_{-j-1} and δ_{-j-1} to obtain ϕ_{-j}. I.e. given signals y and z that satisfy ϕ_{-j-1} and δ_{-j-1} respectively, it is not necessarily the case that $y + z$ satisfies ϕ_{-j}. We can think of ϕ as providing a logical representation of the set of signals $\mathcal{L}(\phi)$. In Section 4.2, using a different representation of $\mathcal{L}(\phi)$, we obtain the ability to traverse the diagram in both directions.

In the next two subsections, we provide examples of the construction of these extended propositions and low-resolution formulas.

4.1.1 Extended Propositions

In this section, provided are examples of these extended propositions, as defined by Eqn. A.7 in Appendix A.

When plotting the extended propositions, the bounds are plotted as points at each time-step. This allows for a more direct comparison between similar extended propositions and cleaner plots when considering monitoring examples, which will come later in the following chapter.

Figure 4.1 shows the extended propositions resulting from the proposition $p := x > 0.5$ created using both the Haar and Daubechies' first level wavelets. State bounds $[-1, 1]$ were used here, and this same choice will be made for all subsequent figures.

This figure illustrates a few important points. Firstly, and most importantly, it illustrates that the choice of wavelet matters. At every position of the two extended propositions, the ones created using the Haar wavelet result in tighter constraints than those created using the Daubechies' wavelet. Second, notice how the constraints resulting from the Haar wavelet are much more uniform than those of the Daubechies' wavelet.

Lastly, the figure demonstrates the necessity of either the over-approximation or the use of two extended propositions, both of which are described in Appendix A. $p_{-1}(0)$ is clearly not the same as $p_{-1}(1)$. Using the Haar wavelet, they look to differ in only a time shift, but simply shifting the Daubechies $p_{-1}(1)$ will not result in achieving exactly $p_{-1}(0)$. So it is necessary to use both in order to have the best chance of detecting violations.

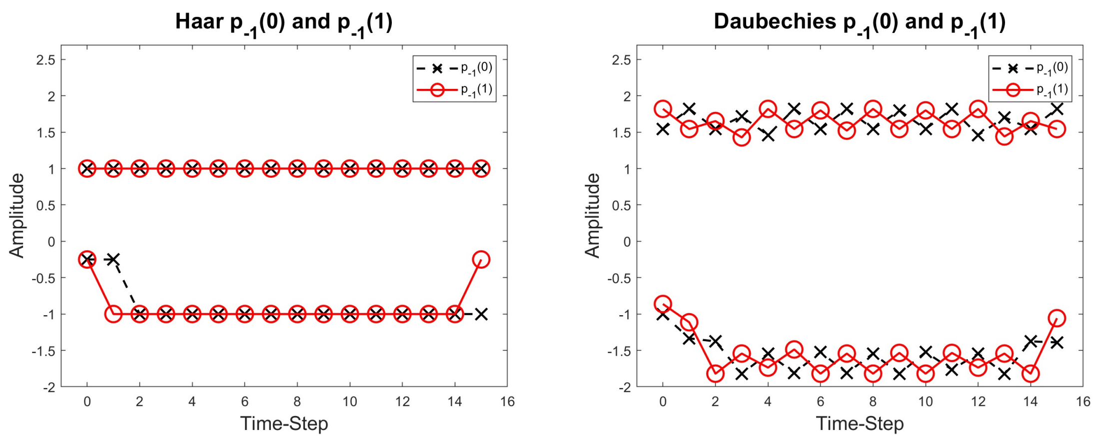

Figure 4.1: Extended propositions from $p := x > 0.5$

Now we turn our attention to Figure 4.2 and compare it to Figure 4.1. Figure 4.2 was created using the same wavelets, but the original proposition was $r := x < 0$. So the constraint due to r on the original signal is smaller than that induced by p, and in the opposite direction. Importantly, both $r_{-1}(0)$ and $r_{-1}(1)$, for either choice of wavelet reflect these differences. The atomic proposition making up r_{-1} are both more relaxed and in the opposite direction than those due to p_{-1}.

4.1.2 Low-Resolution Formulas

Here we give three example formulas and their corresponding low-resolution formulas to only a 1^{st} level of decomposition for simplicity. We will not be calculating the extended propositions here, but will assume that they are already known.

Let us start with $x, 0 \models \Box_{[0,5]}\, p$. The resulting low-resolution formula is then

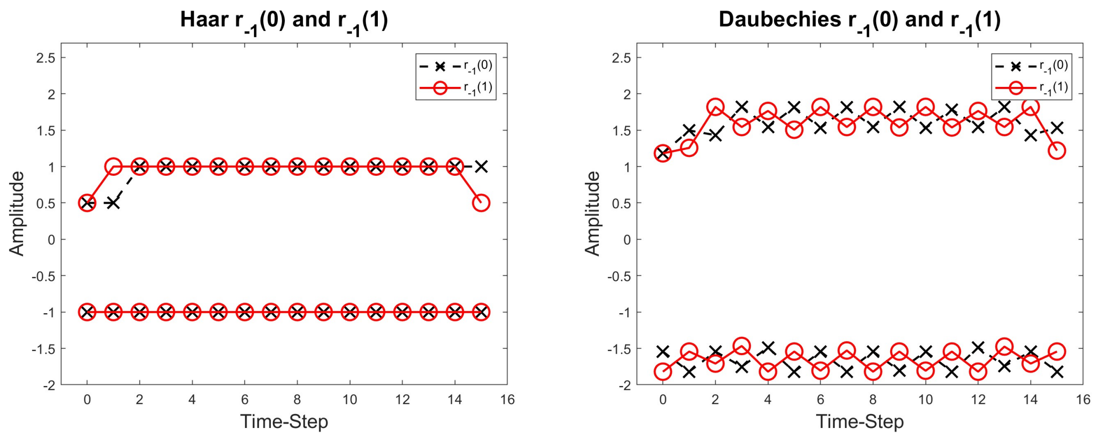

Figure 4.2: Extended propositions from $r := x < 0$

$$x_{-1}, 0 \models \phi_{-1}(0) = \Box_{\{0,2,4\}}\, p_{-1}(0) \wedge \Box_{\{1,3,5\}}\, p_{-1}(1)$$

From the result of Equation A.9. The way to interpret $\{0, 2, 4\}$ is not as an interval, but as containing each time-step the operator is concerned with. So $\Box_{\{0,2,4\}}\, p_{-j}(0)$ applies the extended proposition only to 0, 2, and 4 time-steps in the future. It does not check 1 or 3 time-steps in the future.

Similarly, from Equation A.8, if our formula is $x, 0 \models \Diamond_{[0,5]}\, p$, then we find

$$\phi_{-1}(0) = \Diamond_{\{0,2,4\}}\, p_{-1}(0) \vee \Diamond_{\{1,3,5\}}\, p_{-1}(1).$$

Now let's look at the case of nested operators, namely $x, 0 \models \Box_{[0,5]} \Diamond_{[1,3]}\, p$. We find that, defining $\eta = \Diamond_{[1,3]}\, p$ for clarity,

$$\phi_{-1}(0) = \Box_{\{0,2,4\}}\, \eta_{-1}(0) \wedge \Box_{\{1,3,5\}}\, \eta_{-1}(1)$$
$$= \Box_{\{0,2,4\}} \left(\Diamond_{\{0,2\}}\, p_{-1}(0) \vee \Diamond_{1}\, p_{-1}(1)\right) \wedge \Box_{\{1,3,5\}} \left(\Diamond_{\{0,2\}}\, p_{-1}(1) \vee \Diamond_{1}\, p_{-1}(0)\right)$$

Crucially, the result here is that $p_{-1}(0)$ is only applied to even time-steps and $p_{-1}(1)$ is only applied to odd. Each of the low-resolution formulas presented here are for monitoring x_{-1}. The corresponding low-resolution formulas for monitoring d_{-1}, which we call δ_{-1}, have the same structure which we have seen. We simply switch out p_{-1} for q_{-1}, which are the extended propositions for monitoring d_{-1}.

4.2 Mixed-Integer Encoding of DT-STL and Projected Languages

We now provide a different representation of the language $\mathcal{L}(\phi)$ using a novel mixed-integer constraint encoding of DT-STL semantics. It is then possible to easily obtain the encodings of $\mathcal{L}_{-1}$ and $\mathcal{K}_{-1}$ from that of $\mathcal{L}(\phi)$. The main benefit of this representation is that given arbitrary signals y and z that satisfy the constraints encoding of $\mathcal{L}_{-1}$ and $\mathcal{K}_{-1}$, respectively, it holds that $y+z$ satisfies the constraints encoding of $\mathcal{L}(\phi)$, and therefore satisfies ϕ. Thus we have a true multi-resolution analysis of $\mathcal{L}(\phi)$, completely paralleling the MRA of $\ell^2(\mathbb{Z}_N)$. We can traverse the MRA diagram in both directions. The disadvantage of this constraints representation is that it requires new monitoring algorithms - namely, monitoring reduces to solving the mixed-integer constraints. The logical representation of the previous section, of course, can be monitored using classical algorithms like those found in the toolboxes S-TaLiRo and Breach.

We first encode a formula as a set of Mixed Integer Constraints (MICs), i.e. a set of inequalities over real-valued and boolean variables. The MICs give an explicit characterization of $\mathcal{L}(\phi)$ which is easier to work with.

Definition 3 (Mixed Integer Constraints) *A* MIC set *M is a set of inequalities over real and boolean variables. We write $M.B$ for the boolean variables, $M.BC$ for the purely boolean constraints, and $M.C$ for the mixed constraints.*
An assignment *is a map $A : M.B \rightarrow \{0, 1\}$. A* feasible assignment *is one that respects $M.BC$. If every boolean b in $M.B$ is replaced by $A(b)$, we obtain a set of inequalities over real variables, denoted $M[A]$.*
Given a MIC set M over the elements of signal x and some booleans, x is said to satisfy *M if there exists a feasible assignment A to $M.B$ s.t. x satisfies the resulting inequalities $M[A]$. Assignment A is then said to be a* witness *for satisfaction. We write the satisfaction relation as $x \models M$ and refer to the* language *$\mathcal{L}(M)$ of M as being the set of all signals that satisfy M.*

E.g. the length-2 signal $x = (2, 2.7)$ satisfies the MIC $bx(0) + 1 \leq x(1)$ since setting the boolean $b = 0$ yields $1 \leq x(1)$ which is indeed satisfied by x. The set of all solutions is $\mathcal{L}(M) = \{x \in \ell^2(\mathbb{Z}_2) \mid x(0) + 1 \leq x(1) \text{ or } 1 \leq x(1)\}$.

Lemma 1 *Let ϕ be a DT-STL formula. Then there exists a set MIC_ϕ of mixed integer constraints s.t.*

1. (Equivalence) *a signal satisfies ϕ iff it satisfies MIC_ϕ - i.e., $\mathcal{L}(\phi) = \mathcal{L}(MIC_\phi)$.*
2. (State-Independence) *Every constraint that is not purely boolean is of the form $c_n \leq x(n) \leq d_n$ where c_n and d_n are expressions that do not involve $x(m)$ for any value of m. (They might, however, involve the boolean variables).*
3. (One-Per-n) *For any n, $x(n)$ appears in at most one mixed constraint.*

Proofs are in the Appendix. Previous work [21] also gave a mixed integer encoding of DT-STL semantics, but the two encodings are different because they serve different purposes: control synthesis in their case, wavelet decomposition in ours. Whereas they only needed their encoding to have the Equivalence property, we need ours to also have the State Independence and One-Per-n properties. As we will see, *these two properties make every formula, no matter how complicated, look like a conjunction of atomic propositions*:

$$c_0 \leq x(0) \leq d_0,\ c_1 \leq x(1) \leq d_1,\ \ldots, c_{N-1} \leq x(N-1) \leq d_{N-1}$$

This will be key to deriving the MIC for $\mathcal{L}_{-j}$.

Next, we show that a MIC encoding of a formula can be decomposed into two scale-(-1) MICs (the analysis step), and that the scale-(-1) MICs can be re-composed to yield the encoding (the synthesis step).

Theorem 2 *Given a DT-STL formula ϕ and a signal x that satisfies it (equivalently, satisfies MIC_ϕ), then*

1. *(Analysis) there exist two MIC sets MIC^V_{-1} and MIC^W_{-1} such that x_{-1} satisfies MIC^V_{-1} and d_{-1} satisfies MIC^W_{-1}.*
2. *(Synthesis) For any two signals z and y which satisfy MIC^V_{-1} and MIC^W_{-1}, respectively, $z + y$ satisfies MIC_ϕ.*

The proof of Thm. 2 in the appendix gives an explicit construction of MIC^V_{-1} and MIC^W_{-1}. It can be illustrated using another MRA diagram, this time over MIC sets. This should be compared to the diagram in Fig. 3.2.

$$\begin{array}{l} \ldots \longrightarrow \mathrm{MIC}^V_{-3} \longrightarrow \mathrm{MIC}^V_{-2} \longrightarrow \mathrm{MIC}^V_{-1} \longrightarrow \mathcal{L}(\phi) \\ \ldots \nearrow \ \mathrm{MIC}^W_{-3} \nearrow \ \mathrm{MIC}^W_{-2} \nearrow \ \mathrm{MIC}^W_{-1} \ \nearrow \end{array} \tag{4.2}$$

Theorems 1 and 2 are summarized in this diagram.

$$\begin{array}{cc} x \models \phi \quad \longleftrightarrow & x \quad \models MIC_\phi \\ \downarrow & \updownarrow \\ x_{-1} \models \phi_{-1} & x_{-1} \models MIC^V_{-1} \\ d_{-1} \models \delta_{-1} & d_{-1} \models MIC^W_{-1} \end{array} \tag{4.3}$$

The diagram only shows the case $j = -1$ but as usual, the same applies at every scale.

The arrow directions emphasize that working with the MICs (right column), we can decompose the formula's encoding and re-synthesize it. Working with the logical expressions (left column) we are able to decompose, but we cannot somehow combine ϕ_{-1} and δ_{-1} to yield back ϕ. This has practical implications which we explore in the next section.

Chapter 5: Application and Experiments

In this chapter, an application of the result of Section 4.1 is presented. This application is then demonstrated in three separate instances. We next perform an experiment on 20,000 randomly generated signals.

5.1 Early Violation Detection

We propose an application of our results: the ability to detect violations of a formula through coarse observations of the signal without reconstructing the signal.

The application requires an understanding of a typical wavelet-based communication system: in such as system, a signal x is transmitted in stages. First the wavelet coefficients of x_{-J} are sent, namely $\langle x, \varphi_{-J,k}\rangle, 0 \leq k \leq N/2^J - 1$. (Here J is some pre-specified maximum decomposition level). From these the receiver reconstructs x_{-J} using Eq. (3.3). Then the coefficients $\langle x, \psi_{-J,k}\rangle$ of d_{-J} are sent so the receiver reconstructs x_{-J+1}. Then the coefficients $\langle x, \psi_{-J+1,k}\rangle$ of d_{-J+1} are sent so the receiver reconstructs x_{-J+2}, and so on for a total of J stages. Thus the receiver builds progressively higher-resolution approximations of x: $x_{-J}, x_{-J+1}, \ldots, x_{-1}$ and finally x.

The incremental transmission scheme described above makes sense only if the partial approximations x_{-j} are still useful to the receiver. E.g., in a phone call, if the connection quality suddenly drops, the receiver can still make out the broad outlines of what her interlocutor was saying based on what she has already received. So far in the literature, runtime monitoring was all-or-nothing: you needed the whole signal (or the ability to estimate missing values) to perform monitoring.

We now leverage Thm. 1 to monitor x using only its approximations and details, and so detect violations early. Figure 5.1 depicts this process. The monitor receives the approximation and then each subsequent level of details, monitoring each until it either finds a violation or until x is fully reconstructed. At this point the monitor simply checks whether $x \models \phi$, thus finding the answer as is usual in runtime verification.

This multiresolution monitoring of the logical expressions is only 'half-sound': a

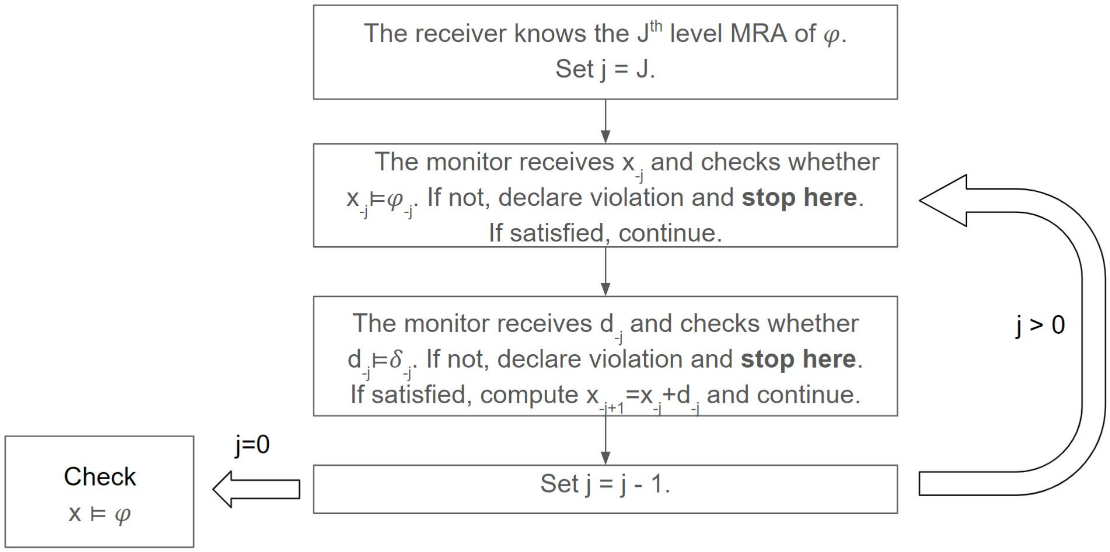

Figure 5.1: Early Violation Detection

violation at scale $-j$ implies violation of ϕ, but it is possible that $x_{-1} \models \phi_{-1}$ and $d_{-1} \models \delta_{-1}$ but $x \not\models \phi$. On the other hand, monitoring the logical expression can be done fast (linear time in signal length) using established toolboxes like S-TaLiRo [4] and Breach [8].

To avoid this half-soundness, an alternative is to use the MIC representation of ϕ and its projected languages (right column of (4.3)). To determine whether $x_{-j} \models \text{MIC}^V_{-j}$, say, requires solving a SAT instance (or solving a mixed-integer program). MIC-based multiresolution monitoring is fully sound: if $x_{-1} \models \text{MIC}^V_{-1}$ and $d_{-1} \models \text{MIC}^W_{-1}$ then we know that $x \models \phi$, and similarly in case of violation. However, SAT solving (or mixed integer programming) can take too much time depending on problem size. Interesting future work would be to try and 'reverse-engineer' the MIC to get a DT-STL formula out of it and run a classical monitor on it.

5.2 Examples in Monitoring

In this section, three examples of monitoring are provided which demonstrate early violation detection as well as the benefits and limitations of monitoring in the wavelet domain. 5.2.1 shows an example where a violation of a formula is found. In 5.2.2 satisfaction is correctly determined when monitoring x_{-1}. Lastly, 5.2.3 shows an example where a violation is missed, and would thus require the full reconstruction of the original signal in order find the violation.

5.2.1 Finding A Violation

We consider the formula $\phi = \Box_{[0,2]}\, x > 0.5$. The corresponding low-resolution formula is $\phi_{-1}(0) = \Box_{\{0,2\}}\, p_{-1}(0) \wedge \Box_{1}\, p_{-1}(1)$. Plot (a) of Fig. 5.2 shows the signal and the amplitude of the proposition. We can see immediately that $x \not\models \phi$ as positions 0 and 1 do not satisfy the proposition. The hope is that we catch this when we monitor x_{-1} against ϕ_{-1}.

Looking at plot (b) of Fig. 5.2, we see that we catch a violation immediately during the first check. $x_{-1}, 0 \not\models p_{-1}(0)$. So a violation is found, and we can stop monitoring at plot (b). Not only are we done monitoring x_{-1}, but we no longer need to monitor d_{-1} or fully reconstruct x. We know at this point $x \not\models \phi$

However, plots (c) and (d) demonstrate a couple of additional points. In plot (c), $x_{-1}, 1 \not\models p_{-1}(1)$ which is good. This is another violation caught. Crucially, we do *not* see a violation in plot (d). Here $x_{-1}, 2 \models p_{-1}(0)$ which it has to, because $x, 2 \models p$.

5.2.2 Finding Satisfaction

We now consider the formula $\phi = \Diamond_{[0,2]}\, x > 0.5$ and the same signal as in Section 5.2.1. We now find $\phi_{-1}(0) = \Diamond_{\{0,2\}}\, p_{-1}(0) \wedge \Diamond_{1}\, p_{-1}(1)$. Now, mechanically we check the same positions of x_{-1} as before so we can again look to Fig. 5.2.

Plot (a) tells us that $x, 0 \models \phi$, as $x, 2 \models p$, so we do not expect to find a violation here. And we don't. $x_{-1}, 2 \models p_{-1}(0)$ so $\phi_{-1}(0)$ is satisfied.

However, just because we found satisfaction while monitoring x_{-1} does not mean we are done.

We next monitor the details, shown in Fig. 5.3. Here we see that d_{-1} satisfies the appropriate extended propositions at each time-step, shown in plots (b), (c), and (d). So $d_{-1} \models \delta_{-1}(0)$.

We are *still* not done at this point. To determine satisfaction with certainty, we need

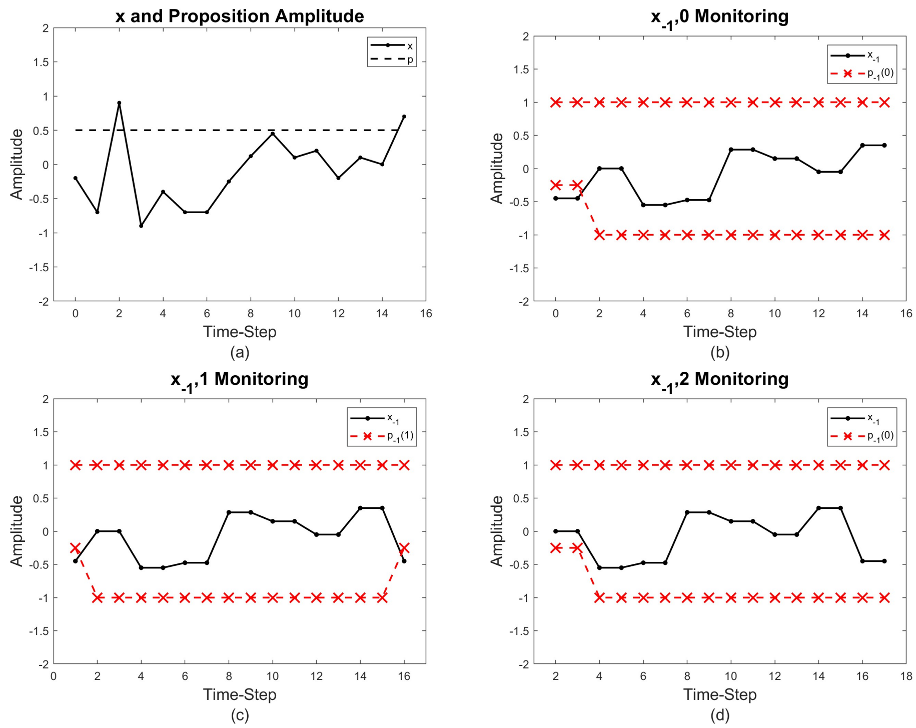

Figure 5.2: Monitoring x_{-1} in which a violation is found in plots (b) and (c).

to then fully reconstruct the signal to verify satisfaction.

5.2.3 Missing a Violation

Here we monitor against the same formula as in 5.2.2, but we tweak x such that the signal no longer satisfies the formula. This is shown in plot (a) of Fig. 5.4. $x \not\models p$ at time-steps 0, 1, and 2.

Now monitoring similarly to before, we see violations of the extended propositions in plots (b) and (c), but we miss the violation in plot (d). Because $x_{-1}, 2 \models p_{-1}(0)$, $x_{-1}, 0 \models \phi_{-1}(0)$. Thus we have missed the violation and will now need to check the details.

In Figure 5.5, we see that we miss violations in two positions, plots (b) and (d) when monitoring the details, so the details formula, δ_{-1} is also satisfied.

This means that we need to, in this case, fully reconstruct x in order to detect this

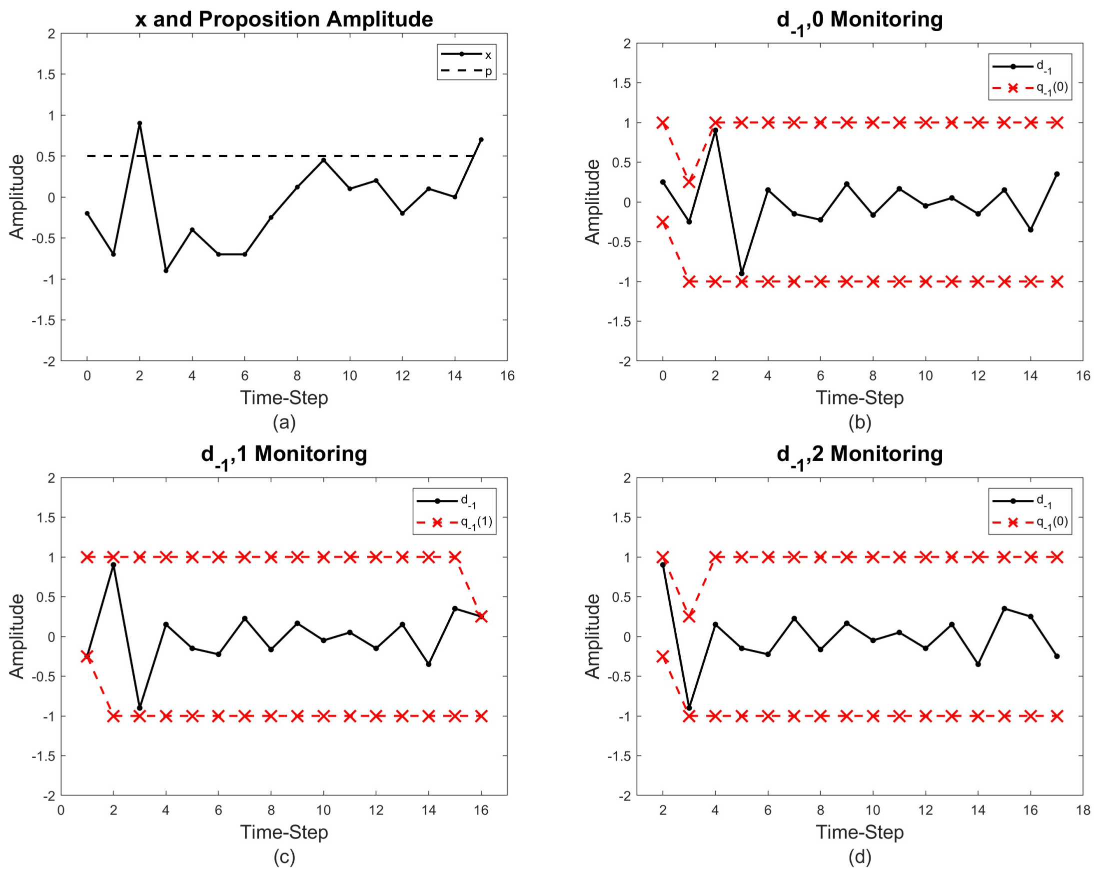

Figure 5.3: Satisfaction found while monitoring d_{-1}.

particular violation.

5.3 Experiments in Monitoring

To test the usefulness of our monitoring application we randomly generate signals and monitor against two formulas. $\phi_1 = \Diamond_{[0,5]} x >= 0.5$ and $\phi_2 = \Box_{[0,5]} x >= 0.5$.

We consider two types of signals. The first is pure noise. At each n we randomly generate a real number in $[-1, 1]$, which are our state bounds. The second is a sum of sines such that

$$x[n] = \sum_{k=1}^{4} 0.25sin(\omega_k n + t_k)$$

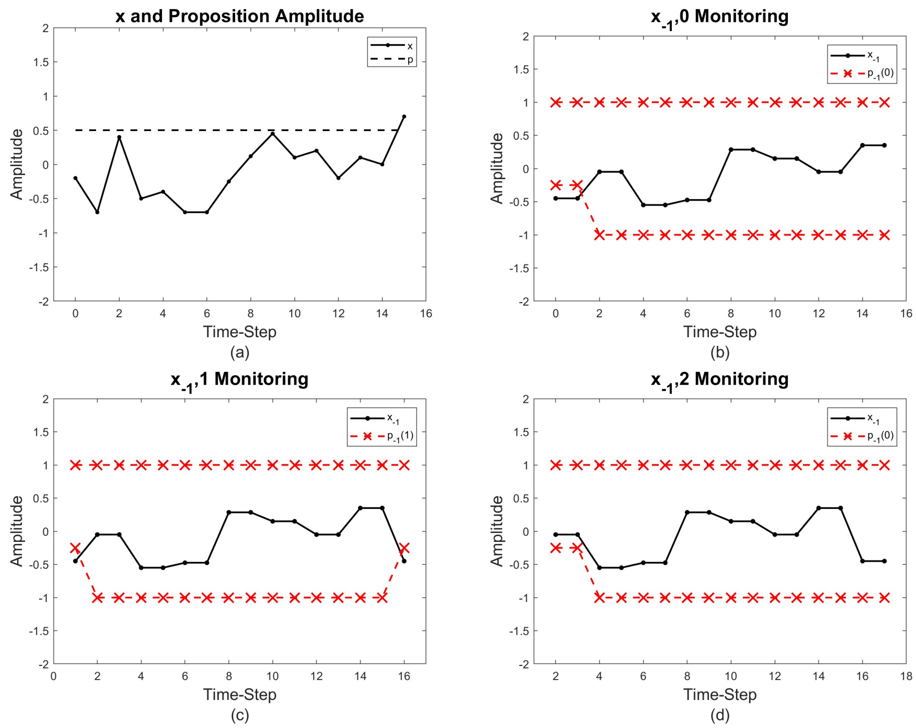

Figure 5.4: Monitoring x_{-1} in which a violation is missed in plot (d).

This keeps the amplitude of the total wave in $[-1, 1]$, thus not violating the state bounds.

The next section details the procedure and the following section presents some results.

5.3.1 Experimental Procedure

We define the above formulas and set the signal length to 512. We set the maximum decomposition level to $J = 5$. We choose the Haar and Debauches' order 6 wavelets.

From here we construct our low resolution formulas, as in Section 4.1.2, for each level of decomposition and for both the details and approximations. We find a list of low-resolution formulas

$$\phi_{-5}, \delta_{-5}, \phi_{-4}, \delta_{-4}, \phi_{-3}, \delta_{-3}, \phi_{-2}, \delta_{-2}, \phi_{-1}, \delta_{-1}$$

We then repeat the following steps 10,000 times for each class of signal, monitoring

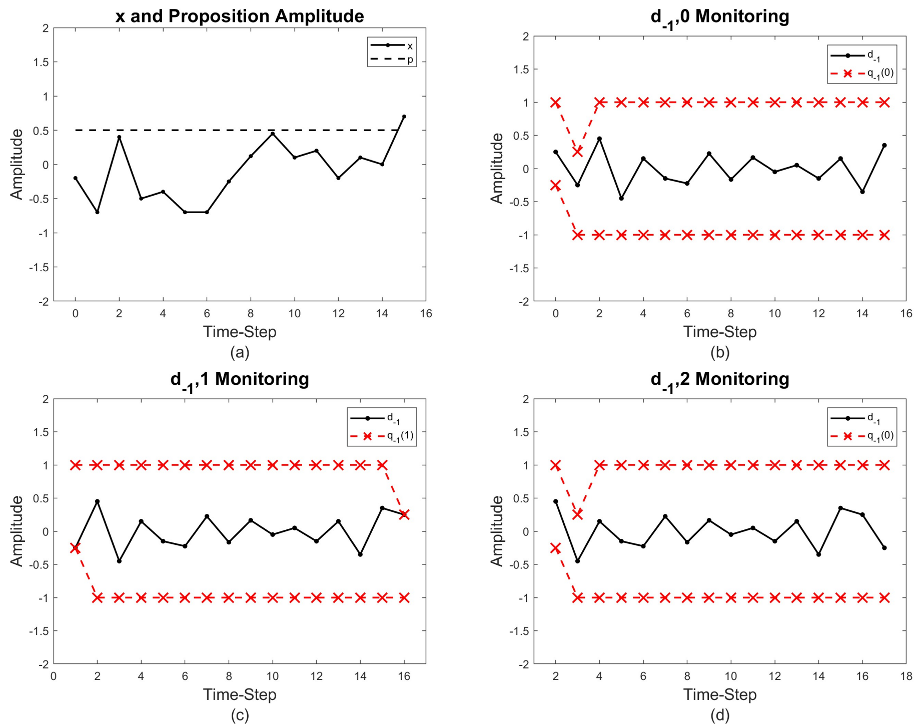

Figure 5.5: Monitoring d_{-1} in which a violation is missed.

each signal against both ϕ_1 and ϕ_2.

1. Generate a randomized signal x. This generation sets $x[n]$ to a random number between $[-1, 1]$ if we are generating the signals that are simply noise. If we are generating sinusoidal signals, we randomly generate $w_k \in [0, 0.1]$ and $t_k \in [-5, 5]$.

2. Check $x \models \phi$ and record the result.

3. Decompose $x = x_{-5} + d_{-5} + \ldots + d_{-1}$ using the chosen wavelet basis.

4. Perform monitoring as presented in Figure 5.1. If a violation is caught, record at what level of decomposition.

5. Repeat 1-4 until all 10,000 signals have been generated and monitored against both $\phi_1 = \Diamond_{[0,5]}\, x >= 0.5$ and $\phi_2 = \Box_{[0,5]}\, x >= 0.5$.

5.3.2 Results

Table 5.1 displays how many signals of each type violate each formula. It's perhaps surprising that ϕ_1 was violated far less frequently by the noisy signal. ϕ_1 is easier to satisfy in a sense. It only requires that a single time-step in the interval results in satisfaction. The purely random signal has good odds of producing at least a single satisfying point over the course of the 6 time-steps the interval is concerned with.

	Sinusoidal	Noise
# signals generated	10000	10000
# violating ϕ_1	9064	1785
# violating ϕ_2	9431	9998

Table 5.1: Violating signal counts for each signal type and formula.

The number of violations caught for each formula by each wavelet are shown in Tables 5.2 and 5.3. They illustrate the point in Section 4.1.1 well: the choice of wavelet *matters*. Clearly, for both of these signal types and formulas, low-resolution monitoring using the Haar wavelet was more effective than when using the Daubechies' wavelet. In fact, the Daubechies' wavelet did not catch a single violation of ϕ_1 while 2661 of 10849 total violations were caught by using the Haar wavelet, which is still only about 25% of the violations in this experiment.

These two tables also seem to suggest that the effectiveness of monitoring using these low-resolution formulas is not only dependent on the features of the signal and wavelet used, but also on ϕ *itself*. Monitoring the same signals against different formulas produces very different ability to catch violations. For example, the Haar wavelet catches the majority of violations, due to noisy signals, of ϕ_2, while missing the majority of ϕ_1. All that was changed was the *temporal operator*. In fact, switching the globally operator improved the ability of both wavelets to catch a violation.

Tables 5.4 through 5.7 breakdown these violations by level of decomposition they were caught at. They suggest that, at least for these two types of signals, violations are most likely to be caught at a lower level of decomposition. Meaning that scale $-j$ is more likely to catch a violation than scale $-j-1$. In each table, scale -1 detects more violations than scale -2. The same can be said for scale -2 when compared to scale -3 and so on, whenever violations are still being caught. Note that we do not find a

	Sinusoidal	Noise
# violations of ϕ_1	9064	1785
# caught using Haar Wavelet	2449	212
# missed using Haar Wavelet	6615	1573
# caught using Daubechies' Wavelet	0	0
# missed using Daubechies' Wavelet	9064	1785

Table 5.2: Violations of ϕ_1 caught and missed by each wavelet.

	Sinusoidal	Noise
# violations of ϕ_2	9431	9998
# caught using Haar Wavelet	3101	9784
# missed using Haar Wavelet	6330	214
# caught using Daubechies' Wavelet	162	1535
# missed using Daubechies' Wavelet	9269	8463

Table 5.3: Violations of ϕ_2 caught and missed by each wavelet.

violation at scale -5 here. This does not mean that we won't, but it could be the case that it happens too rarely to be seen after creating only 10000 signals.

Scale	Haar Violations	DB6 Violations
1	2063	0
2	271	0
3	96	0
4	19	0
5	0	0
Total	2449	0

Table 5.4: Violations, by sinusoidal signals, of ϕ_1 found at each scale during low resolution monitoring out of 9064 total violations.

Scale	Haar Violations	DB6 Violations
1	212	0
2	0	0
3	0	0
4	0	0
5	0	0
Total	212	0

Table 5.5: Violations, by noisy signals, of ϕ_1 found at each scale during low resolution monitoring out of 1785 total violations.

Scale	Haar Violations	DB6 Violations
1	2515	162
2	471	0
3	96	0
4	19	0
5	0	0
Total	3101	162

Table 5.6: Violations, by sinusoidal signals, of ϕ_2 found at each scale during low resolution monitoring out of 9431 total violations.

Scale	Haar Violations	DB6 Violations
1	9115	1533
2	669	2
3	0	0
4	0	0
5	0	0
Total	9784	1535

Table 5.7: Violations, by noisy signals, of ϕ_2 found at each scale during low resolution monitoring out of 9998 total violations.

Chapter 6: Conclusion

We demonstrated that DT-STL formulas have a multi-scale structure that parallels that of signals, and leveraged it to define the application Early Violation Detection, in which we monitor low-resolution signals. We demonstrated that this application does have the potential to catch violations without fully reconstructing the signal, but that it can also miss violations requiring us to fully reconstruct the signal to verify satisfaction. We show that the effective ability to catch violations while monitoring low-resolution signals is seemingly dependent on wavelet, signal, and formula. Regardless of the choice between these three, violations are most likely to be caught at lower levels of decomposition. We constructed a mixed integer encoding that is capable of telling with certainty, given x_{-1} and d_{-1}, whether $x \models \phi$ and also giving the capability to detect violations.

Future work is to also extend this MRA of DT-STL formulas into continuous-time and non-periodic discrete time.

Our methodology could also be applied to robust semantics, in order to study the *rate* of convergence of the multi-scale monitor to the correct answer. It may also be interesting to study how the robustness of the signal is related to the satisfaction value of the decomposed signal and vice-versa. In other words, is a violation of a specification more likely to be found in the low resolution domain by a signal which results in a low robustness value, than by one with a higher robustness value, despite both violating the specification? Or if we see satisfaction of the low-resolution formula with only a small, or relatively near zero, robustness value, can we say anything about how likely the reconstructed signal is to then violate the specification?

Also interesting would be to investigate how these low-resolution formulas interact with TFL as presented by [10], or how they can describe frequency characteristics of a formal specification on their own.

Bibliography

[1] Houssam Abbas, Yash Vardhan Pant, and Rahul Mangharam. Temporal logic robustness for general signal classes. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 45–56, 2019.

[2] Houssam Abbas, Alena Rodionova, Ezio Bartocci, Scott A Smolka, and Radu Grosu. Regular expressions for irregular rhythms. *arXiv preprint arXiv:1612.07770*, 2016.

[3] Houssam Abbas, Alena Rodionova, Konstantinos Mamouras, Ezio Bartocci, Scott A Smolka, and Radu Grosu. Quantitative regular expressions for arrhythmia detection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(5):1586–1597, 2018.

[4] Yashwanth S. R. Annapureddy and Georgios E. Fainekos. Ant colonies for temporal logic falsification of hybrid systems. In *Proc. of the 36th Annual Conference of IEEE Industrial Electronics*, pages 91–96, 2010.

[5] Eugene Asarin and Aldric Degorre. Volume and entropy of regular timed languages: Analytic approach. In Joël Ouaknine and Frits W. Vaandrager, editors, *Formal Modeling and Analysis of Timed Systems*, pages 13–27, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[6] N. Basnet and H. Abbas. Logical signal processing: A fourier analysis of temporal logic. In *Proceedings of the 20th International Conference on Runtime Verification (RV)*, pages 359–382, 2020.

[7] Davide G. Cavezza, Dalal Alrajeh, and András Gýorgy. A weakness measure for gr(1) formulae. *Formal Aspects of Computing*, 33, 2021.

[8] Alexandre Donzé, Thomas Ferrère, and Oded Maler. Efficient robust monitoring for stl. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification*, pages 264–279, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[9] Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In *Proceedings of the International Conference on Formal Modeling and Analysis of Timed Systems*, 2010.

[10] Alexandre Donzé, Oded Maler, Ezio Bartocci, Dejan Nickovic, Radu Grosu, and Scott Smolka. On temporal logic and signal processing. In Supratik Chakraborty and Madhavan Mukund, editors, *Automated Technology for Verification and Analysis*, pages 92–106, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[11] Michael W. Frazier. *A Linear Algebra Introduction to Wavelets*. Springer, 1999.

[12] Noella Grady. Functions of bounded variation. *Dostopno prek: https://www. whitman. edu/Documents/Academics/Mathematics/gra dy. pdf (Dostopano: 7.2. 2017)*, 2009.

[13] Paul Halmos. *Measure Theory*. Van Nostrand and Co., 1950.

[14] S. Jaksić, E. Bartocci, R. Grosu, and D. Nicković. An algebraic framework for runtime verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11):2233–2243, 2018.

[15] Yogi Joshi, Guy Martin Tchamgoue, and Sebastian Fischmeister. Runtime verification of ltl on lossy traces. In *Proceedings of the Symposium on Applied Computing*, SAC '17, page 1379–1386, New York, NY, USA, 2017. Association for Computing Machinery.

[16] Oded Maler and Dejan Nickovic. *Monitoring Temporal Properties of Continuous Signals*. Springer Berlin Heidelberg, 2004.

[17] Stéphane Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A*, 374, April 2016.

[18] The MathWorks. dwt2.

[19] Hector Mesa. Adapted wavelets for pattern detection. In *Iberoamerican Congress on Pattern Recognition*, pages 933–944. Springer, 2005.

[20] Luan Viet Nguyen, James Kapinski, Xiaoqing Jin, Jyotirmoy V Deshmukh, Ken Butts, and Taylor T Johnson. Abnormal data classification using time-frequency temporal logic. In *Proceedings of the 20th international conference on hybrid systems: Computation and control*, pages 237–242, 2017.

[21] V. Raman, A. Donze, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia. Model predictive control with signal temporal logic specifications. In *53rd IEEE Conference on Decision and Control*, pages 81–87, Dec 2014.

[22] Alena Rodionova, Ezio Bartocci, Dejan Nickovic, and Radu Grosu. Temporal logic as filtering. In *Proceedings of the 19th International Conference on Hybrid Systems:*

Computation and Control, HSCC '16, page 11–20, New York, NY, USA, 2016. Association for Computing Machinery.

[23] A. Prasad Sistla, Miloý Žefran, Yao Feng, and Yue Ben. Timely monitoring of partially observable stochastic systems. In *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*, HSCC '14, page 61–70, New York, NY, USA, 2014. Association for Computing Machinery.

[24] J.-L. Starck, F. Murtagh, and A. Bijaoui. *Image Processing and Data Analysis: The Multiscale Approach.* Cambridge University Press, 1998.

[25] Scott D. Stoller, Ezio Bartocci, Justin Seyster, Radu Grosu, Klaus Havelund, Scott A. Smolka, and Erez Zadok. Runtime verification with state estimation. In Sarfraz Khurshid and Koushik Sen, editors, *Runtime Verification*, pages 193–207, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

APPENDICES

Appendix A: Proof of Theorem 1

Recall Thm. 1.

Fix an MRA of $\ell^2(\mathbb{Z}_N)$*, and let* $x \in \ell^2(\mathbb{Z}_N \to X)$ *be a signal with decomposition* $x = x_{-J} + d_{-J} + \ldots + d_{-1}$ *in the given MRA. Let* ϕ *be a DT-STL formula. Then there exist DT-STL formulas* ϕ_{-j} *and* δ_{-j}*, given by Algorithm 1, s.t.* $x \models \phi$ *only if* $x_{-j} \models \phi_{-j}$ *and* $d_{-j} \models \delta_{-j}$*.*

The proof proceeds by induction on the structure of the formula. The proof for ϕ_{-j} and x_{-j} is identical to that for δ_{-j} and d_{-j} (except using the $\psi_{-j,k}$ basis) so we only give the proof for ϕ_{-j}.

We will actually need to prove something slightly more general: namely, that given formula ϕ, *for all* n, we can find a formula $\phi_{-j}(n)$ s.t. if $x, n \models \phi$, then $x_{-j}, n \models \phi_{-j}(n)$. (The theorem statement only concerns the case $n = 0$).

<u>Case $\phi = \top$</u> Since $\top$ imposes no constraints on x, then x_{-j} is similarly unconstrained.

<u>Case $\phi = p = \mu(x) \geq 0$</u> By continuity of μ, $\mathcal{L}(p)$ is a union of disjoint intervals S^i, and by bounded variation of μ, there is only a finite number of S^i's. Because the state space X is bounded, so is every S^i. Therefore, by continuity of μ again, every S^i is closed.

Suppose for now that $\mathcal{L}(p)$ consists of only one interval S, thus p specifies that $x(0) \in S$. Recall that

$$\begin{aligned}
x_{-j}(m) &= \sum_k \langle x, \varphi_{-j,k} \rangle \varphi_{-j,k}(m) = \sum_k \sum_\ell x(\ell) \varphi_{-j,k}(\ell) \varphi_{-j,k}(m) \\
&= \sum_\ell x(\ell) \sum_k \varphi_{-j,k}(\ell) \varphi_{-j,k}(m) = \sum_\ell x(\ell) v_{m,\ell} \\
&= v_{m,n} x(n) + \sum_{\ell \neq n} v_{m,\ell} x(\ell)
\end{aligned}$$

$x, n \models p$ iff $x(n) \in S$, we have that

$$x_{-j}(m) \in v_{m,n} S + \sum_{\ell \neq n} v_{m,\ell}[-a, a] := S_{-j}[m, n] \tag{A.1}$$

Thus every element of x_{-j} is constrained, even though the formula p constrains only $x(n)$.

For the purposes of the induction argument, we are not done. We haven't yet established the induction hypothesis for this base case, since Eq. (A.1) states that $x_{-j}, m \models (x_{-j} \in S_{-j}[m, n])$, which depends on both m and n, while the induction hypothesis requires ϕ_{-j} to be independent of m.

We start by defining new atomic propositions $s_{m,n} := (x_{-j} \in S_{-j}[m, n])$ for a constraint applied to a given n. We then find that

$$\phi_{-j}(n) = \underbrace{\wedge_m \Diamond_m s_{m,n}}_{p_{-j}(n)} \tag{A.2}$$

applying each constraint to the appropriate position of x_{-j}. The case of interest for monitoring is $n = 0$, so considering

$$\phi_{-j}(0) = p_{-j}(0)$$

is sufficient for the base case. Once we begin to consider temporal operators, in the following cases, we will see that considering only $n = 0$ is no longer sufficient.

We call $p_{-j}(n)$ an "extended proposition" as it constrains every time-step of x_{-j}.

If $\mathcal{L}(p)$ consists of $K > 1$ intervals, then p is equivalent to $(x(0) \in S^1) \vee (x(0) \in S^2) \vee ... \vee (x(0) \in S^K)$. Using the above expression for each S^i yields the formulas on Line 25.

Case $\phi = \neg p$ $\mathcal{L}(\neg p)$ is a finite union of disjoint open bounded intervals. Therefore the above reasoning for p applies mutatis mutandis to this case.

Case $\phi = \eta \vee \xi$ If $x, n \models \eta \vee \xi$ then by the induction hypothesis $x_{-j}, n \models \eta_{-j}(n)$ or $x_{-j}, n \models \xi_{-j}(n)$, therefore $x_{-j}, n \models \eta_{-1}(n) \vee \xi_{-1}(n)$.

Case $\phi = \eta \wedge \xi$ Similar to previous case.

<u>Case $\phi = \Diamond_I \eta$</u> We now need to be careful when considering $\Diamond_I \eta$. Temporal operators, such as the eventually operator, apply formulas, and thus propositions, to different time-steps. In the base case of $\phi = p$ above, we only consider when p is applied to $n = 0$. Because temporal operators shift this constraint, we need to consider the constraints induced on x_{-j} when the proposition is applied to $x(n \neq 0)$.

Unfortunately $s_{m,0} \neq s_{m+n,n}$ for at least some n. This is the requirement needed in order to use the same extended proposition on all positions of x_{-j} and still result in sound monitoring. So we cannot make a statement such as $x, n \models \Diamond_I p$ implies that $x_{-j}, n \models \Diamond_I p_{-j}(0)$. This would suggest that we could potentially need a unique

$$p_{-j}(n') \; \forall \; n' \in I \tag{A.3}$$

from Eqn. A.2, which is undesirable. We now have two options.

The first is that we introduce an over-approximation $\bar{S}$ of all sets $S_{-j}[m, n]$,

$$x_{-j}(m) \in S_{-j}[m, n] \implies x_{-j}(m) \in \bar{S}$$

thereby allowing us to only consider the case of $\bar{s} := (x_{-j} \in \bar{S})$. Then we could set $\phi_{-j} = \wedge_m \Diamond_m \bar{s}$. One example is introducing $\bar{v}$ $=\max_m \sum_{\ell \neq n} |v_{m,\ell}|$. Then we have $S_{-j}[m, n] \subseteq \cup_m v_{m,n} S + [-a\bar{v}, a\bar{v}] := \bar{S}$. However, introducing this over-approximation would result in less accurate monitoring, meaning that violations could be caught less often.

The second option is that we attempt to reduce the number of $n' \in I$ we must consider in Eqn. A.3. The rotational nature of the wavelet basis does provide some consolation in this regard.

It is the case that $s_{m,n} = s_{m+2^j,n+2^j}$, which suggests we can limit the number of extended propositions, $p_{-j}(n)$, needed to 2^j. We prove this by first considering $\varphi_{-j,k}$.

$\varphi_{-j,k}$ is simply a rotation of $\varphi_{-j,0}$ by $2^j k$ and thus, $\varphi_{-j,k}(m) = \varphi_{-j,0}(m + 2^j k)$. Now, from Eqn. A.1, we consider the summations.

$$\begin{aligned} v_{m,n} &= \sum_{k=0}^{N/2^j-1} \varphi_{-j,k}(m)\varphi_{-j,k}(n) \\ v_{m+2^j,n+2^j} &= \sum_{k=0}^{N/2^j-1} \varphi_{-j,k}(m+2^j)\varphi_{-j,k}(n+2^j) \end{aligned}$$

Considering the upper sum first.

$$v_{m,n} = \sum_{k=0}^{N/2^j-1} \varphi_{-j,k}(m)\varphi_{-j,k}(n) = \sum_{k=0}^{N/2^j-1} \varphi_{-j,0}(m+2^j k)\varphi_{-j,0}(n+2^j k) \tag{A.4}$$

And now the lower sum.

$$\begin{aligned} v_{m+2^j,n+2^j} &= \sum_{k=0}^{N/2^j-1} \varphi_{-j,k}(m+2^j)\varphi_{-j,k}(n+2^j) \\ &= \sum_{k=0}^{N/2^j-1} \varphi_{-j,0}(m+2^j(k+1))\varphi_{-j,0}(n+2^j(k+1)) \\ &= \sum_{k=1}^{N/2^j} \varphi_{-j,0}(m+2^j k)\varphi_{-j,0}(n+2^j k) \end{aligned} \tag{A.5}$$

By periodicity

$$\varphi_{-j,0}(m+0) = \varphi_{-j,0}(m+N) = \varphi_{-j,0}(m+2^j\frac{N}{2^j})$$

Which means we can rewrite Eqn. A.5 as

$$\sum_{k=0}^{N/2^j-1} \varphi_{-j,0}(m+2^j k)\varphi_{-j,0}(n+2^j k)$$

thereby establishing that $v_{m,n} = v_{m+2^j,n+2^j}$ for any wavelet basis.

Then the equality

$$v_{m,n}S + \sum_{l \neq n} v_{m,l}[-a,a] = v_{m+2^j,n+2^j}S + \sum_{l \neq n+2^j} v_{m+2^j,l+2^j}[-a,a]$$

$$S_{-j}[m,n] = S_{-j}[m+2^j, n+2^j] \tag{A.6}$$

is immediate. This means that the constraint on $x_{-j}(m)$ due to a constraint on $x(n)$ is the same as the constraint on $x_{-j}(m+2^j)$ due to a constraint on $x(n+2^j)$.

Once we define

$$h_i := (n+i) \textbf{ mod } 2^j$$

which we use throughout the rest of this proof, this allows us to write

$$x_{-j}, n \models \underbrace{\wedge_m \Diamond_m s_{m+h_0,h_0}}_{p_{-j}(h_0)} \tag{A.7}$$

which gives us 2^j extended propositions. Each $p_{-j}(n)$ can then be used to monitor $x_{-j}, (n + c2^j)$, where c is a positive integer.

For the purposes of Algorithm MRA-DT-STL (Algorithm 1), Eq. (A.7) is now sufficient. Indeed, we now only need to compute $p_{-j}(n)$ for the cases $n = 0, 1, \ldots, 2^j - 1$. We then just need to construct our low-resolution formulas such that they consider the correct extended propositions at the correct time-steps.

For example, in the case of a first level decomposition, we would have a $p_{-1}(0)$ and a $p_{-1}(1)$. Then, we simply choose the appropriate extended proposition, $p_{-1}(0/1)$ for the constraint applied to each $x(n)$. Where n is odd we would use $p_{-1}(1)$. Similarly when n is even we monitor against $p_{-1}(0)$.

For the remainder of this proof, we assume this $p_{-j}(h_0)$ formulation, rather than the over-approximation.

Now we go back to proving the Eventually case. By definition, $x, n \models \Diamond_I \eta$ iff $\exists n' \in I$ s.t. $x, n{+}n' \models \eta$. Then, by the induction hypothesis $\exists n' \in I$ s.t. $x_{-j}, n{+}n' \models \eta_{-j}(n{+}n')$.

Define

$$I(m) := \left\{ i : i \in I, i = m + 2^j y, y \in \mathbb{N} \right\}$$

$I(m)$ is the set of indices such that $\eta_{-j}(n+m) = \eta_{-j}(n+i)$ for all $i \in I(m)$ as a result of Eqn. A.6.

$\exists n' \in I(m)$ s.t. $x, n+n' \models \eta_{-j}(h_{n'})$ is then the definition of $\Diamond_{I(m)} \eta_{-j}(h_m)$. The induction hypothesis only requires that a single $n' \in I$ results in satisfaction. So we only require that a single $\Diamond_{I(m)} \eta_{-j}(h_m)$ returns satisfaction for $m = 0, 1, \ldots, 2^j - 1$.

This is then the definition of the low resolution formula

$$\begin{aligned} \phi_{-j}(n) = \Diamond_{I(0)} \eta_{-j}(h_0) \vee \Diamond_{I(1)} \eta_{-j}(h_1) \vee \ldots \\ \vee \Diamond_{I(2^j-1)} \eta_{-j}(h_{2^j-1}) \end{aligned} \tag{A.8}$$

where $I(m)$ and h_i are defined as above. This simply groups intervals of 2^j. For the example of $j = 1$, one grouping would contain all the evens in I, while another would contain all the odds. Which contains the odds and which contains the evens is dependent on the current observation moment. This construction ensures that the correct $p_{-1}(n)$ is used during monitoring.

Case $\phi = \Box_I \eta$ By definition, $x, n \models \Box_I \eta$ iff $\forall n' \in I$ $x, n+n' \models \eta$. Then, by the induction hypothesis $\forall n' \in I$ $x_{-j}, n+n' \models \eta_{-j}(n+n')$.

The corresponding low resolution formula for the always case is then similar to that of the eventually case above. We now require all $n' \in I$ result in satisfaction, so we trade out each $\Diamond$ for $\Box$ and each $\vee$ for $\wedge$. We then have

$$\begin{aligned} \phi_{-j}(n) = \Box_{I(0)} \eta_{-j}(h_0) \wedge \Box_{I(1)} \eta_{-j}(h_1) \wedge \ldots \\ \wedge \Box_{I(2^j-1)} \eta_{-j}(h_{2^j-1}) \end{aligned} \tag{A.9}$$

where $I(m)$ and h_i are defined as in the eventually case.

Case $\phi = \eta \mathcal{U}_I \xi$. By periodicity we can always consider $I \subset [0 : N-1]$. The construction of $\phi_{-j}(n)$ then follows from the constructions of the always and eventually cases above. If $I = [a, b]$, $\mathcal{U}_I$ can be rewritten in terms of always and eventually.

$$\xi \mathcal{U}_{[a,b]} \eta = (\Diamond_a \eta) \vee (\Box_{[a,a]} \xi \wedge \Diamond_{a+1} \eta) \vee \ldots \vee (\Box_{[a,b-1]} \xi \wedge \Diamond_b \eta)$$

This then allows us to use the previous cases to construct a low resolution formula for until. It is perhaps inefficient, but suffices for the proof.

Case $\phi = \eta \mathcal{R}_I \xi$. This is handled similarly to the Until case.

Appendix B: Proof of Lemma 1

Recall Lemma 1.

Let ϕ be a DT-STL formula. Then there exists a set MIC_ϕ of mixed integer constraints s.t.

1. (Equivalence) *a signal satisfies ϕ iff it satisfies MIC_ϕ - i.e., $\mathcal{L}(\phi) = \mathcal{L}(MIC_\phi)$.*
2. (State-Independence) *Every constraint that is not purely boolean is of the form $c_n \leq x(n) \leq d_n$ where c_n and d_n are expressions that do not involve $x(m)$ for any value of m. (They might, however, involve the boolean variables).*
3. (One-Per-n) *For any n, $x(n)$ appears in at most one mixed constraint.*

The set of constraints is obtained by an application of the semantics.

Recall the definitions and notation in Def. 3. Define the shift operator R_t by $R_t x = (x(N-t), x(N-t+1), \ldots, x(N-1), x(0), x(1), \ldots, x(N-t-1))$. For integers n, q, (n) mod q is the remainder of dividing n by q.

Case $\phi = \top$ $\text{MIC}_\top = 0 \leq 1$.

Case $\phi = p \in AP$ Write $\mathcal{L}(p) = S_1 \cup \ldots S_K$ where $S_i = [c_i, d_i]$ is a compact interval (this is always the case - see Proof of Thm. 1). Then p is equivalent to $x(0) \in [c_1, d_1] \vee x(0) \in [c_2, d_2] \ldots \vee x(0) \in [c_K, d_K]$. Since the S_i's are disjoint at most one of these disjuncts is true. Thus p is in turn equivalent to the MIC

$$\sum_{i=1}^{K} b_i c_i \leq x(0) \leq \sum_{i=1}^{K} b_i d_i \tag{B.1}$$

$$b_i \in \{0, 1\}, \ \sum_i b_i = 1 \tag{B.2}$$

This MIC satisfies the three properties listed in the Lemma (Equivalence, One-Per-n and State-Independence).

Case $\phi = \neg p$ Write $\mathcal{L}(\neg p) = S_1 \cup \ldots S_K$ where $S_i = (c_i, d_i)$ is a bounded interval (this is always the case - see Proof of Thm. 1). Thus this can be treated in the same way as the p case, but using strict inequalities.

Case $\phi = \eta \wedge \xi$ We build MIC_ϕ as follows: it contains all the purely boolean constraints from MIC_η and MIC_ξ. It also contains all the mixed constraints; if both MIC_η and MIC_ξ have a constraint on $x(n)$, then in MIC_ϕ these are replaced by one constraint which uses the max of the lower bounds, and the min of the upper bounds. This combining ensures MIC_ϕ has property One-Per-n. Moreover MIC_ϕ inherits the properties of Equivalence and State-Independence from MIC_η and MIC_ξ.

Case $\phi = \eta \vee \xi$ Let $c_n \leq x(n) \leq d_n$ be the mixed constraint on $x(n)$ from MIC_η, and let $e_n \leq x(n) \leq f_n$ be the mixed constraint from MIC_ξ. By the One-Per-n property, these are the only constraints (if any) involving $x(n)$. Create a fresh boolean variable b_0 (not used in either MIC_η or MIC_ξ), and define MIC_ϕ as follows. MIC_ϕ has all the purely boolean constraints of MIC_η and MIC_ξ, and for every n it contains

$$b_0 c_n + (1 - b_0)e_n \leq x(n) \leq b_0 d_n + (a - b_0)f_n \tag{B.3}$$

MIC_ϕ has the Equivalence property. Indeed, if $x \models \eta \vee \xi$, it satisfies either formula, or both. If it satisfies η then by the induction hypothesis there exists an assignment A to the booleans of MIC_η s.t. x satisfies the resulting constraints. Then the assignment $b_0 = 1$, combined with A, shows that x satisfies MIC_ϕ. Similarly if it satisfies ξ. If it satisfies both, then a fortiori it satisfies η, and we use the preceding reasoning.

Now suppose x satisfies MIC_ϕ, and let A be the boolean assignment that witnesses that. If $A(b_0) = 0$ then x satisfies MIC_ξ, therefore (by the induction hypothesis) satisfies ξ and so ϕ. If $b_0 = 1$ in A then x satisfies MIC_η, therefore x satisfies η and so ϕ.

Moreover MIC_ϕ has the State-Independence property: by the induction hypothesis, c_n, d_n, e_n, f_n do not depend on the state x, and so neither do the bounds in Eq. (B.3). Finally, it's obvious from Eq. (B.3) that it has One-Per-n.

Even though the Always and Eventually operators are derived from the Until, we give these two cases explicitly here, because we will use them to do the proof for the Until case.

Case $\phi = \Box_I \eta$ By periodicity we may consider, without loss of generality, that $I \subset [0 : N-1]$, say $I = [\ell : k]$. Let

$$c_n(B) \leq x(n) \leq d_n(B), 0 \leq n \leq N-1 \tag{B.4}$$

be the mixed constraints in MIC_η, where $B = \text{MIC}_\eta.B$. The B variables are constrained by purely boolean constraints which we denote BC. By State Independence x does not appear in the bounds.

To derive the MIC set for ϕ we simply replicate the η bounds $k - \ell + 1$ times, from $n = \ell$ to $n = k$. For example: if an η constraint is $2b + 3b' \leq x(2) \leq 2(1-b),\ b + b' \geq 1$, and $I = [2,3]$ then we construct

$$2b_2 + 3b_2' \leq x(2+2) \leq 2(1-b_2)$$

$$2b_3 + 3b_3' \leq x(2+3) \leq 2(1-b_3)$$

$$b_2 + b_2' \geq 1, b_3 + b_3' \geq 1$$

In general, let $B_m^{\ell+j}$ be a fresh copy of the variables in B, for $0 \leq j \leq k - \ell$, $\ell + j \leq m \leq (\ell + j + N - 1) \bmod N$.

Then MIC_ϕ has the following constraints: every $B_m^{\ell+j}$ is constrained by $\text{MIC}_\eta.BC$; and

$$c_{m-(\ell+j-1)}(B_m^{\ell+j}) \leq x(m) \leq d_{m-(\ell+j-1)}(B_m^{\ell+j})$$

As with the AND case, if $x(n)$ appears in more than one inequality, we substitute them all with an inequality that uses the minimum of the right-hand sides and the max of the left-hand sides, thus meeting One-Per-n. State Independence is inherited from MIC_η. Equivalence can be seen to hold by inspection.

Even though Eventually and Always are derived operators we will give them explicitly since it is easier to prove the Until case from them.

Case $\phi = \diamondsuit_I\, \eta$ We re-use the definitions from the Always case. Recall the state-space bounds $|x| \leq a$. To derive the MIC set for ϕ we create a choice over the moment where the η bounds apply, from $n = \ell$ to $n = k$. For example: if an η constraint is $2b + 3b' \leq x(2) \leq 2(1-b),\ b + b' \geq 1$, and $I = [2,3]$ then we introduce the new booleans g_2, g_3 (choice of time moment), b_2, b_2', b_3, b_3' (copies of the original booleans) and construct the inequalities:

$$g_2(2b_2 + 3b_2') - (1 - g_2)a \leq x(2+2) \leq g_2(2(1-b_2)) + (1 - g_2)a$$

$$g_3(2b_3 + 3b_3') - (1 - c3)a \leq x(2+3) \leq g_3(2(1-b_3)) + (1 - g_3)a$$

$$b_2 + b_2' \geq 1, b_3 + b_3' \geq 1 \text{ (copies of constraints)}$$

$$g_2 + g_3 \geq 1 \text{ (at least one time step witnesses satisfaction of } \eta)$$

Thus if $g_t = 1$, it follows that $R_t x \models \eta$, and so on.

In general, let $B_m^{\ell+j}$ be a fresh copy of the variables in B, for $0 \leq j \leq k - \ell$, $\ell + j \leq m \leq (\ell + j + N - 1) \bmod N$. Let $g_\ell, \ldots, g_k$ be fresh booleans. Then MIC_ϕ has the following constraints:

a) every $B_m^{\ell+j}$ is constrained by $\mathrm{MIC}_\eta.B$, and by $\sum_i g_i \geq 1$;

b) for $0 \leq j \leq k - \ell$ (every step in I) and for $\ell + j \leq m \leq (\ell + j + N - 1) \bmod N$ (over the entire length of the shifted trajectory $R_{\ell+j}x$)

$$g_\ell \cdot c_{m-(\ell+j-1)}(B_m^{\ell+j}) - (1 - g_\ell) \cdot a \leq x(m) \tag{B.5}$$

$$x(m) \leq g_\ell \cdot d_{m-(\ell+j-1)}(B_m^{\ell+j}) + (1 - g_\ell) \cdot a \tag{B.6}$$

(We broke the inequality into two for better presentation). This MIC set can be seen to have the Equivalence property: e.g., if $x \models \Diamond_I \eta$ then $R_i x \models \eta$ for some $i \in I$, and by the induction hypothesis there exists an assignment $A : B \to \{0, 1\}$ s.t. $R_i x \models \mathrm{MIC}_\eta$, i.e. s.t. Eq. (B.4) is satisfied. This can be completed arbitrarily to an assignment to the full set of variables $\{B_m^{\ell+j}\}$, and with $g_i = 1$, to yield a witness assignment for MIC_ϕ. The reverse direction is analogous. State Independence and One-Per-n are easily seen to hold.

<u>Case $\phi = \eta \mathcal{U}_I \xi$</u> By periodicity of x we may consider, without loss of generality, $I \subset [0 : N - 1]$, say $I = [\ell : m]$. Then we can re-write ϕ as

$$\begin{aligned}\phi = (\Box_{[0,\ell-1]}\, \eta \wedge \Diamond_\ell\, \xi) \vee (\Box_{[0,\ell]}\, \eta \wedge \Diamond_{\ell+1}\, \xi) \vee \ldots \\ \vee (\Box_{[0,m-1]}\, \eta \wedge \Diamond_m\, \xi)\end{aligned}$$

(This is an inefficient encoding, but it suffices for the proof). We can now re-use the above results to get the encoding for the Until.

<u>Case $\phi = \eta \mathcal{R}_I \xi$</u> Similar to the Until case.

Appendix C: Proof of Theorem 2

Given a DT-STL formula ϕ and a signal x that satisfies it (equivalently, satisfies MIC_ϕ), then

1. *(Analysis) there exist two MIC sets MIC^V_{-1} and MIC^W_{-1} such that x_{-1} satisfies MIC^V_{-1} and d_{-1} satisfies MIC^W_{-1}.*

2. *(Synthesis) For any two signals z and y which satisfy MIC^V_{-1} and MIC^W_{-1}, respectively, $z + y$ satisfies MIC_ϕ.*

In what follows we give the proof for MIC^V_{-1}. The proof for MIC^W_{-1} is identical so we skip it. Recall the notation introduced in Def. 3. Let $v_{n,m} := \sum_k \varphi_k(n)\varphi_k(m)$ and $w_{n,m} := \sum_k \psi_k(n)\psi_k(m)$ for $0 \le n, m \le N - 1$. So we can write $x_{-1}(m) = \sum_n v_{m,n}x(n)$ and $d_{-1}(m) = \sum_n w_{m,n}x(n)$.

Proof for the Analysis step

Case $\phi = \top$ We define $\mathrm{MIC}^V_{-1} = \mathrm{MIC}^W_{-1} = 0 \le 1$.

We now tackle all the remaining cases in one go. Let ϕ be a DT-STL formula with MIC set M. Lemma 1 established that M has at most N mixed constraints of the form

$$c_n \le x(n) \le d_n \tag{C.1}$$

where c_n, d_n are expressions over constants and booleans *but not states*, and the booleans are constrained by $M.BC$. Thus from the perspective of projection onto V_{-1}, the bounding quantities c_n, d_n are constant. Since $x_{-1}(m) = \sum_n v_{m,n}x(n)$ it follows that $x_{-1}(m) \in \sum_n v_{m,n}[c_n, d_n]$ (in interval arithmetic). Thus

$$\begin{aligned} M_{-1}.C &:= \left\{ x_{-1}(m) \in \sum_n v_{m,n}[c_n, d_n] \mid \ 0 \le m \le N-1 \right\} \\ M_{-1}.BC &:= M.BC \end{aligned}$$

Proof for the Synthesis step

We will first need the following simple facts about orthonormal basis matrices. Define matrices $\Phi = [\varphi_1, \dots \varphi_{N/2-1}]$ and $\Psi = [\psi_1, \dots, \psi_{N/2-1}]$. Then it is easily seen that $x_{-1} = \Phi\Phi^T x$ and $d_{-1} = \Psi\Psi^T x$. Since $x = x_{-1} + d_{-1} = (\Phi\Phi^T + \Psi\Psi^T)x$ for any x, it holds that $\Phi\Phi^T + \Psi\Psi^T = I$, the identity matrix. Therefore, denoting by $\Phi_{i:}$ (resp. $\Psi_{i:}$) the i^{th} row of Φ (resp. Ψ), we have that

$$\Phi_{m:}^T \Phi_{n:} + \Psi_{m:}^T \Psi_{n:} = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases} \tag{C.2}$$

Finally note that $v_{m,n} = \Phi_{m:}^T \Phi_{n:}$ and $w_{m,n} = \Psi_{m:}^T \Psi_{n:}$ so

$$v_{m,n} + w_{m,n} = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases} \tag{C.3}$$

Eq. (C.3) allows us to conclude that when $m \neq n$, if $v_{m,n} \geq 0$ then $w_{m,n} \leq 0$.

Now let ϕ be a DT-STL formula with MIC encoding $M = \{c_n \leq x(n) \leq d_n \mid 0 \leq n \leq N-1\}$, with projections MIC_{-1}^V and MIC_{-1}^W. Define the index sets $P_n := \{m \neq n \mid v_{m,n} \geq 0\}$ and $N_n := \{m \neq n \mid v_{m,n} < 0\}$. Let y be an arbitrary element of $\mathcal{L}(\mathrm{MIC}_{-1}^V)$ and z an arbitrary element of $\mathcal{L}(\mathrm{MIC}_{-1}^W)$. From the Analysis step and Eq. C.3 we know that

$$y(n) \in v_{n,n}[c_n, d_n] + \sum_{m \in P_n} [v_{m,n}c_n, v_{m,n}d_n] + \sum_{m \in N_n} [v_{m,n}d_n, v_{m,n}c_n]$$

$$z(n) \in w_{n,n}[c_n, d_n] + \sum_{m \in N_n} [w_{m,n}c_n, w_{m,n}d_n] + \sum_{m \in P_n} [w_{m,n}d_n, w_{m,n}c_n]$$

Adding the two inequalities and leveraging Eq. C.3, it comes that $(y + z)(n) \in (v_{n,n} + w_{n,n})[c_n, d_n] = [c_n, d_n]$: thus we recover exactly the MIC encoding of ϕ.