Multilevel Lattice Network Codes

Alister Burr and Yi Wang, University of York

alister.burr@york.ac.uk
Introduction: Lattices in wireless communications

Multilevel coded modulation

Multilevel lattices: the elementary divisor construction

Multilevel lattice decoding
  - Layered Integer Forcing
  - Iterative decoding

Conclusions
- i.e. **forward error-correcting** (FEC) codes
- A **code** is a finite set of **codewords** of length \( n \)
  - Code contains \( M \) codewords – encodes \( \log_2(M) \) bits
- where a codeword is a sequence of \( n \) **symbols**, usually drawn from a finite **alphabet** of size \( q \)
  - we will often assume the alphabet is a Galois field \( \mathbb{F}_q \) or \( \text{GF}(q) \) or a ring \( \mathbb{R}(q) \)
- In a communication system the codewords must be translated into **signals** of length \( nT \)
  - representing the variation in time of some quantity, such as electromagnetic field strength
- Each code symbol is typically **modulated** to some specific real or complex value of this variable
Message:

Encode

Codeword:

Modulate

Signal:
- Each coded signal can then be represented as a point in \( N \)-D **signal space**
  - where modulated values of symbols provide the \( n \) coordinate values
- Code is represented by ensemble of points in signal space
- Noise on channel equivalent to vector \( z \) in signal space
- Decoder chooses closest point
- Error probability determined by **minimum Euclidean distance** between signal space points
A **lattice code** is then defined by the (finite) set of lattice points within a certain region

- the **shaping region**
- ideally a hypersphere centred on the origin
- this limits the maximum signal energy of the codewords

Lattice may be offset by adding some vector
- Define fine lattice $\Lambda$ for the code
  - plus a **coarse lattice** $\Lambda'$ which is a sub-lattice of $\Lambda$
- Then use a Voronoi region $V'$ of the coarse lattice as the shaping region
- Modulo-$\Lambda'$ operation
  - for any point $P \notin V'$ find $P - (\lambda \in \Lambda') \in V'$
Wireless signals consist of a sine wave *carrier* at the transmission frequency (MHz – GHz)

Sine waves can be modulated in both amplitude and phase

- hence the signal corresponding to each modulated symbol is 2-D
- also conveniently represented as a complex value
- typically represented on a *phasor* diagram

Hence wireless signals can be represented in $2n$ dimensions

- or $n$ complex dimensions
Constructions based on FEC codes

- For practical purposes in communications, we require lattices in very large numbers of dimensions
  - typically 1000, 10 000, 100 000…
- Lattices of this sort of dimension most easily constructed using FEC codes such as LDPC and turbocodes
- Most common constructions encountered are called Constructions A and D (Conway and Sloane)
  - Construction A based on a single code
  - Construction D is multilevel, based on a nested sequence of codes
Construction A

- Start with a $q$-ary linear code $C$ with generator matrix $G_C$
- The set of vectors $\lambda$ such that $\lambda \mod q$ is a codeword of $C$ form a Construction A lattice from $C$:
  \[
  \Lambda = \{ \lambda : \lambda \mod q \in C \} \]
- Alternatively we can write:
  \[
  \Lambda = q\mathbb{Z}^n + C
  \]
- The generator matrix of the lattice:
  \[
  G = \begin{bmatrix}
  G_C & 0 \\
  qI_{n-k} & 
  \end{bmatrix}
  \]
- Note that minimum distance is limited by $q$
  - this also limits the coding gain of the lattice code to at
Let $\mathcal{C}_0 \subseteq \mathcal{C}_1 \subseteq \mathcal{C}_2 \ldots \subseteq \mathcal{C}_a$ be a family of linear binary codes

- where $\mathcal{C}_0$ is the $(n, n)$ code and $\mathcal{C}_i$ is an $(n, k_i)$ code

Then the lattice is defined by:

$$
\Lambda = \left\{ \lambda : \lambda = z + \sum_{l=1}^{a} \sum_{j=1}^{k_l} d_j^l \frac{c_{j,l}}{2^{l-1}} \right\}
$$

where $z \in 2^n$, $c_{j,l}$ is the $j^{th}$ basis codeword of $\mathcal{C}_i$, and $d_j^l \in \{0,1\}$ denotes the $j^{th}$ data bit for the $i^{th}$ code

$$
\begin{align*}
\sum & \quad 1/2 \\
1/4 & \quad 1/4 \\
1/2a-1 & \quad 1/2a-1 \\
\vdots & \quad \vdots \\
\end{align*}
$$
Construction of complex lattices

- Construction A/D as described result in real lattices, based on integers
- Gaussian and Eisenstein integers form the algebraic equivalent in complex domain of the ring of integers
  - all are Principal Ideal Domains (PID)
- Hence can similarly construct complex constellations from them to form complex lattices

\[ Z[i] = a + bi, \ a, b \in \mathbb{Z} \]

\[ Z[\omega] = a + b \omega, \ a, b \in \mathbb{Z}, \ \omega = e^{2\pi i/3} \]
Consider *fine* and *coarse* lattices, $\Lambda$ and $\Lambda_c$, both based on Gaussian integers

$$\Lambda_c \subset \Lambda$$

Here we assume that each point in the coarse lattice is a point in the fine multiplied by some Gaussian integer $q$

- i.e. the coarse is a scaled and rotated version of the fine
- and the fine is just the Gaussian integers

$$\Lambda_c = q\mathbb{Z}[i] \subset \Lambda = \mathbb{Z}[i]$$

We then define our signalling *constellation* as consisting of those Gaussian integers which fall in the Voronoi region of the coarse lattice
Example

- e.g. $q = 2 + i$
- Blue points are fine lattice
- Red points are coarse lattice
- Fundamental region $V'(0)$ is region closer to origin than any other coarse lattice point
- Hence constellation is green points, plus origin
- $|q|^2 = 5$
This construction provides an isomorphic mapping between constellation points and a finite field or ring:

- a finite field if \( q \) is a Gaussian prime
- a ring otherwise
- in either case the size is \( |q|^2 \)

We can treat this as a quotient lattice:

\[
\Lambda/\Lambda_c = \mathbb{Z}[i]/q\mathbb{Z}[i] \cong \begin{cases} 
\mathbb{F}_{|q|^2} & q \text{ prime} \\
\mathbb{R} \left( |q|^2 \right) & \text{otherwise}
\end{cases}
\]

If \( q \) is prime, can define a code over the field.

Again, minimum Euclidean distance limited by \( q \)
- We need relatively large $q$ to construct useful lattice codes
  - then code radix is large
  - which typically makes decoding much more complex
- “Conventional” codes used in wireless communications are binary
  - often admit of simpler decoders
- Size of constellation given by $|q|^2$, either a prime or the square of a prime
  - “conventional” constellation size is a power of 2
  - difficult to map binary data to a prime constellation size
  - Only some constellation sizes are available
Outline

- Introduction: Lattices in wireless communications
- Multilevel coded modulation
  - Multilevel lattices: the *elementary divisor construction*
- Multilevel lattice decoding
  - Layered Integer Forcing
  - Iterative decoding
- Conclusions
More bandwidth-efficient wireless communications requires high order constellations – $M \geq 2$

One approach would be to design an $M$-ary code to use with such constellations

- however these tend to be either very complex
- or to give poor performance
- rarely used in practice

Alternative is multilevel coded modulation (MLCM)
In MLCM we devise a multi-symbol mapping scheme for the constellation

- typically an $M$-point constellation with $M = 2^m$ is labelled with $m$-bit labels
- we choose $m$ length $n$ binary codes, typically with different rates and minimum distance
This structure allows codes to be decoded in succession

Soft output demodulator (SODem) generates likelihoods of code symbols

Decoder $1$ then attempts to decode level 1 code

- feeds result back to SODem to help demodulate next level
- and so on until level $m$ is decoded

Mapping may play important role
Iterative Decoding further develops this by allowing several iterations of whole decoding process

- similar to turbo decoding
- gives performance close to Shannon bound, with appropriate choice of codes
- BER within 0.5 dB of Shannon (constellation-constrained) capacity
- with 12 iterations
Introduction: Lattices in wireless communications

Multilevel coded modulation

Multilevel lattices: the *elementary divisor construction*

Multilevel lattice decoding
  - Layered Integer Forcing
  - Iterative decoding

Conclusions
We partition the quotient lattice into the direct sum of a series of **primary sub-lattices**:

\[ \Lambda / \Lambda' = \Lambda_{p_1} / \Lambda'_{p_1} \oplus \Lambda_{p_2} / \Lambda'_{p_2} \oplus \cdots \oplus \Lambda_{p_m} / \Lambda'_{p_m} \]

where \( \Lambda_{p_i} / \Lambda'_{p_i} \triangleq \{ \lambda \in \Lambda / \Lambda' : p_i^\gamma \lambda = 0 \} \)

and \( p_i \) is some prime in the PID on which the lattice is based (here we assume Gaussian or Eisenstein integers) and \( \gamma \) is some integer.

In each level of the partition we select elements of the quotient lattice which are **annihilated** by multiplication by some power of \( p_i \)

that is, they are reflected back onto the coarse lattice

This operates in a similar way to mapping by set partitioning in MLCM
■ We can then define the isomorphism:

\[
\Lambda_{p_i}/\Lambda'_{p_i} \cong S/\langle p_{i_1}^{\theta_1} \rangle \oplus S/\langle p_{i_2}^{\theta_2} \rangle \oplus \cdots \oplus S/\langle p_{i_t}^{\theta_t} \rangle
\]

■ (This is simplified in cases of practical interest to us in that only the first term is present)

■ \(S/\langle p_{i_1}^{\theta_1} \rangle\) is a quotient ring which is isomorphic either to a finite field (if \(\theta = 1\)) or a finite chain ring (otherwise)

■ Thus the space of message symbols \(w^i\) at the \(i^{th}\) level is over this ring

■ This provides a mapping \(\phi_i\) between the quotient primary sub-lattice and a vector of data symbols:

\[
\hat{w}^i = \bigoplus_j S/\langle p_{i_j}^{\theta_j} \rangle
\]
Elementary Divisor Construction

- So far this does not provide a practical means of constructing the lattices
- The *Elementary Divisor Construction* provides a means equivalent to Construction A/D to construct a lattice from a set of codes and a complex constellation
  - in fact it subsumes these constructions
- The fine lattice is defined as:

  \[ \Lambda \triangleq \{ \lambda \in S^m : \tilde{\sigma}(\lambda) \in C^1 \oplus C^2 \oplus \cdots \oplus C^m \} \]

  - where \( \square^1, \square^2 \ldots \square^m \) denote the codes for each level, over corresponding ring/field
  - and \( \sigma : S \mapsto S/\langle p_1^{n_1} \rangle \times S/\langle p_2^{n_2} \rangle \times \cdots \times S/\langle p_m^{n_m} \rangle \)
    is a map (labelling scheme) defined by the partition of the PID by the primes \( p_1, p_2 \ldots p_m \)
- Lattice based on Eisenstein integers $\mathbb{Z}[[\omega]]$, with $q = 2(1+2\omega)$, $p_1 = 2$, $p_2 = (1+2\omega)$

- Coarse lattice:

- Constellation cardinality $|q|^2 = 12$

- Because this is a ring rather than a field we cannot readily form a Construction A lattice

- Instead we partition into 2 quotient lattices based on $p_1$ and $p_2$
Cardinality of level 1 quotient lattice $\Lambda_{p_1}/\Lambda'$ is

$$\frac{|q|^2}{|p_1|^2} = 3$$

Provides first digit of constellation label
Cardinality of level 2 quotient lattice $\Lambda_{p_2}/\Lambda'$ is

$$\frac{|q|^2}{|p_2|^2} = 4$$

Provides second digit of constellation label
- Hence the partition provides a mapping scheme for the constellation.
- In the example:
  \[ w^1 \in \mathbb{3}, \ w^2 \in \mathbb{4} \]
- Hence use two codes, on \( \mathbb{3} \) and \( \mathbb{4} \).
Outline

- Introduction: Lattices in wireless communications
- Multilevel coded modulation
- Multilevel lattices: the *elementary divisor construction*
- Multilevel lattice decoding
  - Layered Integer Forcing
  - Iterative decoding
- Conclusions
Multilevel lattice network code: decoding

- Multilevel structure also assists with decoding
  - allows each level to be decoded separately
  - with simpler decoder because of reduced cardinality

- Two approaches:
  - Layered Integer Forcing (LIF)
  - Multilevel decoding, one-shot or with iteration
Layered Integer Forcing

- Transmitter: mapping, add dither, shaping
- Receiver: MMSE scaling (different for different levels), remove dither, quantise, demap
- On each level, can treat as a separate nested lattice code
For example, we decode Level 2 independently of Level 1 by performing modulo-$\Lambda_{p2}$ operation on received signal rather than modulo-$\Lambda'$.
We also introduce an alternative decoder based on soft output detection at each level:
- c.f. multistage decoding of MLCM

Again we can use a one-shot multistage decoder:
- or apply multiple iterations of the decoding process.
In the example – level 2

- **Received signal**
- **After modulo-lattice**

- Level 2 detector has information from level 1 decoder to assist
- e.g. if level 1 decoder indicates that ‘0’ more likely than ‘1’ at level 1, then level 2 detector will select ‘2’ rather than ‘3’
- Comparison of separate level decoding (Non-MSD), multistage decoding (MSD) and iterative decoding (IMSD)
- Shows how level 1 assists decoding of level 2
- Iteration results in “turbo-cliff” ~0.8 dB from capacity
- \( h = [1, 1] \)
Layered Integer Forcing

- Layered Integer Forcing with and without iterations
- Without iteration result a little worse than separate soft detection of levels
- With iteration, about 1.8 dB from capacity
- \( h = [1.17 + 2.15i, 1.25 - 1.63i] \)
Conclusions

- Multilevel lattice network coding has the potential to significantly reduce decoding complexity
  - compared to lattices based on Construction A
- Also provides much more flexibility in choice of constellation size
- We introduce *layered integer forcing* as a general approach for decoding general multilevel lattice codes
- Also the *elementary divisor construction* (EDC) for multilevel network codes based on codes
  - this subsumes Constructions A and D
  - also an iterative multistage decoding approach, which gives simulated performance close to capacity
Further work

- Note that we have presented here only one “flavour” of EDC
  - that in which each level is isomorphic to a finite field
- Other “flavours” have levels which are isomorphic to a finite chain ring
  - hybrid forms are also possible
- In these the level label is itself composite
- Note that in the ideal case the label would be binary, with constellation size a power of 2
  - allowing binary codes to be used
  - this may be possible for the finite chain ring form, with lattices based on Gaussian integers
The presentation is based on two of our papers:

  - also submitted to IEEE Trans. Information Theory
- The work was supported in part by EC FP7 contract no. 31877 “DIWINE”