XX Congreso de Ecuaciones Diferenciales y Aplicaciones
X Congreso de Matemática Aplicada
Sevilla, 24-28 septiembre 2007
(pp. 1–??)

Convolutional decoding through a tracking problem

José Ignacio Iglesias Curto[1,2], Uwe Helmke[1]

[1] *Mathematisches Institut, Universität Wrzburg.*
[2] *Dpto. de Matemáticas, Universidad de Salamanca*
E-mails: joseig@usal.es,helmke@mathematik.uni-wuerzburg.de.

Resumen

Convolutional codes can be regarded as discrete time linear systems. This relationship has been studied along decades, and concepts from both theories have found their counterparts into the other one.

In this context, decoding of a received word can be interpreted as a tracking problem. This should allow to give practical decoding algorithms for convolutional codes.

However, coding theory is usually studied over finite fields while optimal control problems have been considered over the real or complex fields. The solutions to these problems are not applicable as they make use of an Euclidean metric in which finite fields lack.

We state a tracking problem over finite fields using the Hamming metric instead of a bilinear quadratic form, and we propose a solution via block decoding. In particular, we focus on the tracking problem associated to a convolutional decoding problem, which leads to a method for decoding general convolutional codes. Under some conditions, a bigger number of errors than half the minimum distance can be corrected.

1. Introduction

Optimal control problems are usually considered over the fields of real or complex numbers. However there are problems that can be formulated in terms of linear systems defined over a finite field. Examples of that arise in coding theory, in particular in convolutional coding. It is a well known fact that a convolutional code can be interpreted in several ways as a linear dynamical system

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + Du_t \end{aligned}$$

defined over a finite field. Accordingly several results from linear systems theory have found their counterparts in convolutional coding theory, leading to a few constructions of codes and a better understanding of their properties.

In particular, we emphasize that the decoding process can be interpreted in terms of systems theory in at least two ways [4]. On the one hand, it can be viewed as a tracking problem, where the decoder should track the received message by the most probable codeword sent. On the other hand, it can be considered as a filtering problem, where the decoder is requested to filter the noise introduced by the channel. Here, we will focus on the first one.

We consider well known optimal control problems over finite fields where we, as an initial step, transfer known results over infinite fields to problems posed over finite fields, for example those related with the construction and decoding of convolutional codes. In particular we will focus on a tracking problem and the decoding of general convolutional codes.

In section 2 we consider a tracking problem. We first present the classical case and we discuss why it cannot be directly applied to the finite fields context. Then, we set the problem over a finite field for a finite input sequence and we present our solution.

In section 3 we focus on the convolutional decoding process as a tracking problem (which will be addressed as an infinite process). A Receding Horizon solution will be applied to this problem. As a result, we obtain a decoding method for general convolutional codes.

2. The tracking problem over a finite field

Consider a discrete-time linear system

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t \\ x_{t_0} &= x_0 \end{aligned} \tag{2.1}$$

A classical tracking problem is stated in the following way: Given a sequence $\{\tilde{y}_t\}_{t_0}^T$, find an input sequence $\{u_t\}_{t_0}^T$ that minimizes the cost function

$$J(x_0, u(\cdot), T) = \sum_{t=t_0}^{T} [u_t^\top R u_t + (y_t - \tilde{y}_t)^\top Q (y_t - \tilde{y}_t)]$$

with R and Q positive and nonnegative definite matrices respectively of the appropriate dimensions.

This is a well known problem, studied over the real and complex numbers [1, 2]. However, its solution is not applicable for convolutional decoding. Convolutional codes are usually defined over a finite field $\mathbb{F}$, while the solutions of tracking problems over infinite fields make use of an Euclidean inner product which is not defined over finite fields. Over finite fields the metric considered is the one induced by the Hamming distance.

We restate the problem so that it makes sense over a finite field. For that, let's consider the linear system (2.1), where A, B, C are constant matrices, $x_t \in \mathbb{F}^n$, $u_t \in \mathbb{F}^k$, $y_t \in \mathbb{F}^p$ and $\mathbb{F}$ is any finite field.

Problem 2.1. Given a sequence $\{\tilde{y}_t\}_{t_0}^T$, find an input sequence $\{u_t\}_{t_0}^T$ for the system (2.1) that minimizes the cost function

$$\begin{aligned} J(x_0, u(\cdot), T) &= \sum_{t=t_0}^{T} [w(Ru_t) + w(Q(y_t - \tilde{y}_t))] \\ &= w(QCx_0 - Q\tilde{y}_{t_0}) + \sum_{t=t_0}^{T-1} [w(Ru_t) + w(QCx_{t+1} - Q\tilde{y}_{t+1})] + w(Ru_T). \end{aligned}$$

Here $w(x)$ denotes the Hamming weight of the vector x and Q and R are constant matrices of appropriate dimensions, R with maximum rank.

Remark 2.2. In the classical tracking problem, the matrices Q and R in the cost function are nonnegative and positive definite, respectively. This means that for a non-zero vector u_t (resp. $y_t - \tilde{y}_t$), the corresponding term in cost function will always increase (resp. will not decrease) the sum. As the Hamming weight takes values in the non-negative integers (and in particular zero only for the zero vector), to have the same notion in our case, we should impose that Ru is non-zero for every non-zero vector u (i.e. R has maximum rank) while no condition is required on $Q(y_t - \tilde{y}_t)$.

A brute force solution given by checking all the possible sequences is infeasible unless T is very small. To solve it efficiently, we note that the last term of the sum is minimized just with $u_T = 0$, and we can write the cost function as

$$\begin{aligned} &J(x_0, u(\cdot), T) = w(QCx_0 - Q\tilde{y}_{t_0}) + \sum_{t=t_0}^{T-1} [w(Ru_t) + w(QCx_{t+1} - Q\tilde{y}_{t+1})] \\ &= J(x_0, u(\cdot), T-1) + w(Ru_{T-1}) + w(QCx_T - Q\tilde{y}_T) \end{aligned}$$

As x_0 is known, the first term of the sum is fixed, $J(x_0, u(\cdot), t_0) = w(QCx_0 - Q\tilde{y}_{t_0})$. So we are in a position to make use of a well-known optimality principle:

Bellman's optimality principle [3]: *An optimal trajectory, has the property that at an intermediate point, no matter how it was reached, the rest of the trajectory must coincide with an optimal trajectory as computed from this intermediate point as the initial point.*

This allows to reduce the overall minimization problem to a sequence of single-stage minimizations of the expressions

$$w(Ru_t) + w(QCx_{t+1} - Q\tilde{y}_{t+1}).$$

We have

$$QCx_{t+1} = QCAx_t + QCBu_t$$

and we can group the known terms at time t as

$$z_t = QCAx_t - Q\tilde{y}_{t+1}.$$

So, we can pose generically the single-stage minimization problem as: find a vector u that minimizes the expression

$$w(Ru) + w(QCBu + z).$$

Let us now consider the vector and the matrix

$$z' = (z_1, \dots, z_n, 0, \overset{k}{\dots}, 0)^\top \quad B_1 = \begin{pmatrix} QCB \\ R \end{pmatrix}$$

so that $z' + B_1 u = (z + QCBu, Ru)$, and $w(z' + B_1 u) = w(Ru) + w(QCBu + z)$. Then, the problem is equivalent to finding the vector of minimum weight in the coset $\{z' + B_1 u\}_u$.

Let us consider the block code generated by B_1, $\mathcal{C}_{B_1}$, and $z' = e + B_1 v$ as a received word, with $B_1 v$ a codeword and e the error.

These cosets are equal

$$\{z' + B_1 u\}_u = \{e + B_1(u + v)\}_{w=u+v}$$

and the vector with minimum weight is e.

So, finding the vector $z' + B_1 u$ of minimum weight is equivalent to decode z' as a codeword from the code $\mathcal{C}_{B_1}$. This can be done with an appropriate decoding algorithm. Note that in all iterations the code to consider (and as a result also the decoding algorithm) will be the same.

We can summarize the solution process in the next algorithm.

Algorithm 2.3.

- `Calculate` $J(x_0, u(\cdot), t_0) = w(QCx_0 - Q\tilde{y}_{t_0})$.
- `For every` t, $t_0 \le t < T$
  - `Use a block decoding algorithm to decode` $z'_t = (z_t, 0)$ `as a codeword from the code` $\mathcal{C}_{B_1}$, `obtaining` v_t `as the information word.`
  - `Update` $u_t = -v_t$, $x_{t+1} = Ax_t + Bu_t$, $J(x_0, u(\cdot), t+1) = J(x_0, u(\cdot), t) + w(Ru_t) + w(QCx_{t+1} - Q\tilde{y}_{t+1})$.
- `The solution is` $\{u_{t_0}, \dots, u_{T-1}, 0\}$ `and the optimal value` $J(x_0, u(\cdot), T)$.

We can just give an upper bound for the optimal value of the cost function, which will depend on the *covering radius* ρ_{B_1} of the code $\mathcal{C}_{B_1}$.

Definition 2.4. The *covering radius* $\rho_{\mathcal{C}}$ of a code $\mathcal{C}$ is the maximum distance from any vector of $\mathbb{F}^n$ to its nearest codeword.

Theorem 2.5. *Let ρ_{B_1} be the covering radius of the code C_{B_1} and u^* an optimal solution to problem 2.1. Then the optimal cost is upper bounded as*

$$J(x_0, u^*, T) \le (T-1)\rho_{B_1} + w(QCx_0 - Q\tilde{y}_{t_0}).$$

Proof:

The proof is immediate. For any vector, including the vector z' that we decode at every step, there is a codeword in a distance less or equal than ρ_{B_1}, and we can bound the cost at every time step by

$$J(x_0, u^*, t+1) - J(x_0, u^*, t) = \min_u \{w(z' + B_1 u)\} = w(e) \le \rho_{B_1}.$$

So after $T-1$ time instants we can give the bound

$$J(x_0, u^*, T) \le (T-1)\rho_{B_1} + w(QCx_0 - Q\tilde{y}_{t_0}).$$

$\square$

3. Convolutional decoding as a tracking problem

Decoding is together with the construction of efficient codes the main task in coding theory, and in general a non-trivial objective. As explained in [4], the decoding of convolutional codes can be interpreted as a tracking problem. Applying the solution for the tracking problem over finite fields (over which codes are usually defined) we can give a decoding algorithm for general convolutional codes.

Let us consider the generator matrix $G(z)$ of a convolutional code. As shown in [4] we can divide it into

$$G(z) = \begin{pmatrix} P(z)_{n-k\times k} \\ Q(z)_{k\times k} \end{pmatrix}$$

so that $degdetQ(z) = \delta$, the degree of the code. This leads to a controllable state space representation of the code as a linear system

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + Du_t \\ x_0 &= 0 \end{aligned} \tag{3.1}$$

with transfer function $C(zId - A)^{-1}B + D = P(z)Q(z)^{-1}$.

Given a codeword $c(z) = \sum_{t\geq 0} c_t z^t$ each vector coefficient can be divided into

$$c_t = \begin{pmatrix} y_t \\ u_t \end{pmatrix}.$$

Decoding a received sequence $\{v_t = (\tilde{y}_t, \tilde{u}_t)\}$ consists of finding

$$\min_{c\in\mathcal{C}}\{d(c,v)\} = min\left\{\sum_{t=0}^{T} d(y_t, y'_t) + d(u_t, u'_t)\right\} = min\left\{\sum_{t=0}^{T} w(y_t - \tilde{y}_t) + w(u_t - \tilde{u}_t)\right\}.$$

This can be thought as a tracking problem with cost function

$$\begin{aligned} J(x_0, u(\cdot), T) &= \sum_{t=0}^{T} [w(u_t - \tilde{u}_t) + w(y_t - \tilde{y}_t)] = \\ &= J(x_0, u(\cdot), T-1) + w(u_T - \tilde{u}_T) + w(y_T - \tilde{y}_T). \end{aligned} \tag{3.2}$$

In a practical convolutional decoding process, in order to save time, decoding can start before the whole information stream is received, which in practice results in considering it as an infinite process. To solve the corresponding problem we apply a Receding Horizon technique which consists on following iteratively these steps:

- we consider the initial (known) state x_t.
- we solve an N-step finite horizon tracking problem, i.e., we find $\{u_{t+i}\}_{i=0}^{N}$ which minimizes

$$J(x_t, u, N) = \sum_{i=0}^{N} [w(u_{t+i} - \tilde{u}_{t+i}) + w(y_{t+i} - \tilde{y}_{t+i})]$$

- we update our "decoded" input with $\{u_t, \ldots, u_{t+L}\}$ and our "decoded" output with $\{y_t, \ldots, y_{t+L}\}$, and we get x_{t+L+1}. L will depend on how many steps we can assure are decoded without errors.

Step 2 represents the main problem to be solved. To consider a direct solution (instead of an iterative one) will give some information on how big should L be.

Given x_t we want to find $\{u_t, \ldots, u_{t+N-1}\}$ that minimize

$$\sum_{i=0}^{N}[w(u_{t+i} - \tilde{u}_{t+i}) + w(y_{t+i} - \tilde{y}_{t+i})] = w(z_N)$$

with

$$z_N = (y_{t+N} - \tilde{y}_{t+N}, u_{t+N} - \tilde{u}_{t+N}, \ldots, y_t - \tilde{y}_t, u_t - \tilde{u}_t).$$

We know that

$$x_{t+i} = A^i x_t + \sum_{r=0}^{i-1} A^{i-r-1} B u_{t+r}$$
$$y_{t+i} = C x_{t+i} + D u_{t+i}.$$

So the vector z_N can be written as

$$z_N = \widehat{w_{t,N}} + \widehat{B_N} u_{t,N}$$

where

$$\widehat{w_{t,N}} = \begin{pmatrix} CA^N x_t - \tilde{y}_{t+N} \\ -\tilde{u}_{t+N} \\ CA^{N-1} x_t - \tilde{y}_{t+N-1} \\ -\tilde{u}_{t+N-1} \\ \vdots \\ CAx_t - \tilde{y}_{t+1} \\ -\tilde{u}_{t+1} \\ Ax_t - \tilde{y}_t \\ -\tilde{u}_t \end{pmatrix} \quad \widehat{B_N} = \begin{pmatrix} D & CB & CAB & \ldots & CA^{N-1}B \\ Id & 0 & 0 & 0 & 0 \\ 0 & D & CB & \ldots & CA^{N-2}B \\ 0 & Id & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & & 0 & & D \\ & & & & Id \end{pmatrix}$$

$$u_{t,N} = (u_{t+N}, u_{t+N-1}, u_{t+N-2}, \ldots, u_t).$$

Sorting properly we have a vector $w_{t,N} + B_N u_{t,N}$ with the same weight of z_N, being

$$w_{t,N} = \begin{pmatrix} CA^N \\ \vdots \\ C \\ 0 \\ \vdots \\ 0 \end{pmatrix} x_t - \begin{pmatrix} \tilde{y}_{t+N} \\ \vdots \\ \tilde{y}_t \\ \tilde{u}_{t+N} \\ \vdots \\ \tilde{u}_t \end{pmatrix}$$

$$B_N = \begin{pmatrix} D & H_0 & H_1 & \dots & H_{N-1} \\ 0 & D & H_0 & \dots & H_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & H_0 \\ 0 & \dots & \dots & 0 & D \\ & & & & \\ & & Id & & \\ & & & & \end{pmatrix}$$

with $H_i = CA^iB$.

The problem will be solved by decoding, with an appropriate block decoding algorithm, the vector $w_{t,N}$ as a codeword from the code generated by B_N.

The generator matrix, B_N, is systematic so the check matrix of the code is

$$H = \left(\begin{array}{c|ccccc} & D & H_0 & H_1 & \dots & H_{N-1} \\ & 0 & D & H_0 & \dots & H_{N-2} \\ -Id & \vdots & \ddots & \ddots & \ddots & \vdots \\ & \vdots & & \ddots & \ddots & H_0 \\ & 0 & \dots & \dots & 0 & D \end{array} \right).$$

It is well-known that the minimum number of linear dependent columns of H is the minimum distance of the code C_{B_1}, and this bounds the number of correctable errors that can occur in N time steps.

The length of the receding horizon, N, and the number of steps to update, L, will depend on how many errors we are able to correct in the code generated by B_N.

Theorem 3.1. *For a fixed N, being d the minimum distance of the code $\mathcal{C}_{B_N}$ generated by B_N, we can correct $\lfloor \frac{d'}{2} \rfloor$ errors ($d' \geq d$) if every codeword of $\mathcal{C}_{B_N}$ with weight $\leq d'$ has zeros in the components from $(N-L)(n-k)+1$ till $N(n-k)$ and from $Nn - Lk + 1$ till Nn.*

Proof:

Recall that a decoding error happens if instead of the codeword sent, c, the decoding result is another codeword $c' = c + c_e$, where the decoding error c_e is also a codeword.

In our case, as the generator matrix B_N is systematic, the decoded codeword can be divided in an information part and a parity check part. The information part is actually a sequence of N inputs of the solution for our tracking problem. As we will only update L of these inputs, there is an admissible decoding error, as long as it happens in the last $N-L$ inputs (corresponding to the first coordinates of the information word) which will not affect to the solution of our problem.

An error only in the last $N-L$ inputs corresponds to a decoding error codeword c_e with non-zeros only in the components from 1 till $(N-L)(n-k)$ and from $N(n-k)+1$ till $Nn - Lk$.

So, all the codewords with support only in these components are admissible errors, and in particular, if every word with weight $\leq d'$ is and admissible error then we can correct $\lfloor \frac{d'}{2} \rfloor$ error positions. $\square$

We obtain a decoding algorithm for a general convolutional code with a state space representation (3.1).

Algorithm 3.2.

- Fix the length of the receding horizon, N, and the update length L, according to Theorem 3.1.
- $x_0 = 0$, $J(x_0, u(\cdot), 0) = 0$.
- For $i > 0$, $(i-1)L < T$,
  - Use a block decoding algorithm to decode $w_{iL,N}$ as a codeword from the code $\mathcal{C}_{B_N}$, obtaining $u_i = \{u_{iL+1}, \ldots, u_{iL+N}\}$ as the information word.
  - Update the solution input u with $\{u_{iL+1}, \ldots, u_{(i+1)L}\}$.
  - Calculate $x_{t+1} = Ax_t + Bu_t,\ y_t = Cx_t + Du_t$ for all $iL+1 \leq t \leq (i+1)L$.
  - Update $J(x_0, u(\cdot), (i+1)L) = J(x_0, u(\cdot), iL) + \sum_{i=0}^{L} [w(u_{t+i} - \tilde{u}_{t+i}) + w(y_{t+i} - \tilde{y}_{t+i})]$.
- The decoding output is the sequence $\{c_t = (y_t, u_t)\}$, and the number of errors $J(x_0, u(\cdot), T)$.

4. Conclusion

We have posed a tracking problem over a finite field, modifying the cost function to be minimized, so that it considers the Hamming weight instead of an Euclidean metric. We gave a solution to this problem by using coding theory tools.

This makes it possible to consider convolutional decoding as a tracking problem. We set the corresponding problem, whose solution gives a decoding method for general convolutional codes.

Acknowledgments

The first author has been partially supported for this work by fellowship A/06/12428 from DAAD.

We want to thank Prof. Harald Wimmer for his valuable advice and helpful comments.

Referencias

[1] B.D.O. Anderson and J.B. Moore, *Linear optimal control*, Prentice-Hall, January 1971.

[2] B.D.O. Anderson and J.B. Moore, *Optimal control: Linear quadratic methods*, Prentice-Hall, 1989.

[3] R. Bellman, *Dynamic programming*, Princeton University Press, Princeton, NJ, 1957.

[4] J. Rosenthal, *Some interesting problems in systems theory which are of fundamental importance in coding theory*, Proceedings of the 36th IEEE Conference on Decision and Control, 1997.