

# Iterative Source–Channel Coding Approach to Witsenhausen’s Counterexample

Johannes Karlsson, Ather Gattami, Tobias J. Oechtering, Mikael Skoglund

**Abstract**—In 1968, Witsenhausen introduced his famous counterexample where he showed that even in the simple linear quadratic static team decision problem, complex nonlinear decisions could outperform any given linear decision. This problem has served as a benchmark problem for decades where researchers try to achieve the optimal solution. This paper introduces a systematic iterative source–channel coding approach to solve problems of the Witsenhausen Counterexample-character. The advantage of the presented approach is its simplicity. Also, no assumptions are made about the shape of the space of policies. The minimal cost obtained using the introduced method is 0.16692462, which is the lowest known thus far.

## I. INTRODUCTION

The most fundamental problem in control theory, namely the static output feedback problem has been open since the birth of control theory. The question is whether there is an efficient algorithm that can decide existence and find stabilizing controllers, linear or nonlinear, based on imperfect measurements and given memory. The static output feedback problem is just an instance of the problem of control with information structures imposed on the controllers, which has been very challenging for decision theory researchers. In 1968, Witsenhausen [21] introduced his famous counterexample:

$$\inf_{\gamma_1(\cdot), \gamma_2(\cdot)} \mathbf{E} [k^2 \gamma_1^2(X_0) + X_2^2] \quad (1)$$

where

$$X_1 = \gamma_1(X_0) + X_0, \quad (2)$$

$$X_2 = X_1 - \gamma_2(Y_2), \quad (3)$$

$$Y_1 = X_0, \quad (4)$$

$$Y_2 = X_1 + W, \quad (5)$$

$X_0 \sim N(0, \sigma^2)$ , and  $W \sim N(0, 1)$ . Here we have two decision makers, one corresponding to  $\gamma_1$  and the other to  $\gamma_2$ . The problem is a two-stage linear quadratic Gaussian control problem, where the cost at the first time-step is  $\mathbf{E}[k^2 \gamma_1^2(X_0)]$  and  $\mathbf{E}X_2^2$  at the second one. At the first time-step, the controller has full state measurement,  $Y_1 = X_0$ . At the second time-step, it has imperfect state measurement,  $Y_2 = X_1 + W$ . What is different to the classical output feedback problem, is that the controller at the second stage does not have information from the past since it has no information about the output  $Y_1$ . Thus, the controller is restricted to be

The authors are with the School of Electrical Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden. E-mail: {johk, gattami, tobias.oechtering, mikael.skoglund}@ee.kth.se.

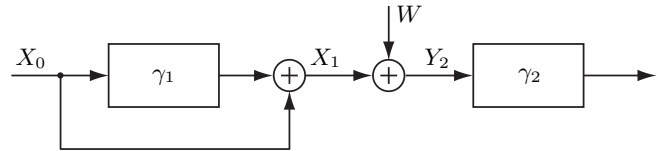


Fig. 1. Schematic view of the system.

a static output feedback controller. Witsenhausen showed that even in the simple linear quadratic Gaussian control problem above, complex nonlinear decisions could outperform any given linear decision. This problem has served as a benchmark problem for decades where researchers try to achieve the optimal solution. It has been pointed out that the problem is complicated due to a so called “signaling-incentive”, where decisions are not only chosen to minimize a given cost, but also to encode information in the decisions in order to signal information to other decision makers in the team. In the example above, decision maker 2 measures  $Y_2 = X_0 + \gamma_1(X_0) + W$ , so its measurement is affected by decision maker 1 through  $\gamma_1$ . Hence, decision maker 1 not only tries to optimize the quadratic cost in (1), but also *signal* information about  $X_0$  to decision maker 2 through its decision,  $\gamma_1(X_0)$ .

The problem also has a nice communication theoretic analogue. First, write Witsenhausen’s counterexample as minimizing the cost

$$\mathbf{E} [k^2 \gamma_1^2(X_0) + (X_0 + \gamma_1(X_0) - \gamma_2(X_0 + \gamma_1(X_0) + W))^2]$$

with respect to  $\gamma_1(\cdot)$  and  $\gamma_2(\cdot)$ . Now consider the slightly modified problem

$$\begin{aligned} &\text{minimize } \mathbf{E} (X_0 - \gamma_2(\gamma_1(X_0) + W))^2 \\ &\text{subject to } \mathbf{E} \gamma_1^2 \leq p \end{aligned}$$

The modification made is that we removed  $X_0$  term from the measurement of  $\gamma_2$ , and removed  $\gamma_1$  from the objective function, and instead added a constraint  $\mathbf{E} \gamma_1^2 \leq p$  to make sure that it has a limited variance (of course we could set an arbitrary power limitation on the variance). The modified problem is exactly the Gaussian channel coding/decoding problem!

Previous work has been pursued on understanding the Witsenhausen Counterexample. Suboptimal solutions were found in [13] studied variations of the problem when the signaling incentive was eliminated. In [14], [12], connections to information theory were studied. An extensive study of the information theoretic connection was made in [3], where it was shown that coupling between decision makers

in the cost function introduced the nonlinear behavior of the optimal strategies. An ordinal optimization approach was introduced in [5] and a hierarchical search approach was introduced in [16], where both rely on a given structure of the decisions. The first method that showed that optimal strategies may have “slopes” to the quantizations was given in [2]. Solutions with bounds are studied in [11]. A potential games approach in the paper by [17] found the best known value to the date of its publication, namely 0.1670790.

In this paper, we will introduce a generic method of iterative optimization based on ideas from source–channel coding [9], [8], [20], [15], that could be used to solve problems of the Witsenhausen Counterexample character. The numerical solution we obtain for the benchmark problem is of high accuracy and renders the lowest value known thus far, **0.16692462**.

## II. NOTATION

$p(\cdot)$  and  $p(\cdot|\cdot)$  denote probability density functions (pdfs) and conditional pdfs, respectively.

## III. ITERATIVE OPTIMIZATION

We will now present an iterative design algorithm, based on person-by-person optimality, for solving the minimization in equation (1). The method we propose is related to the Lloyd–Max algorithm [18], [19], [10] that is successfully used when designing quantizers. A quantizer can be described by its partition cells and their corresponding reproduction value. The partition cells define to which codeword analog values are encoded and the reproduction values define how the analog value is reproduced from the codeword. In general, there is no explicit, closed-form solution to the problem of finding the optimal quantizer [10]. The idea of the Lloyd–Max algorithm is to assume that either the partition cells or the reproduction values are fixed; with one part fixed, it is straightforward to derive an optimal expression for its counterpart. By optimizing one part at a time in an iterative fashion, the quantizer will converge to a local optimum. The Lloyd–Max algorithm has been generalized and used in various joint source–channel coding applications, for example, [6], [22], [7], [8], [20], [15]. The generalization of the Lloyd–Max algorithm that will be used in this paper involves four key elements:

- 1) Formulation of necessary conditions on  $\gamma_1$  and  $\gamma_2$  such that they are individually optimal given that  $\gamma_2$  and  $\gamma_1$ , respectively, are fixed.
- 2) Discretization of the “channel” space between  $\gamma_1$  and  $\gamma_2$  such that  $X_1$  and the input to  $\gamma_2$  are restricted to belong to a finite set  $\mathcal{S}_L$ .
- 3) Iterative optimization of  $\gamma_1$  and  $\gamma_2$  to make sure that they, one at a time, fulfill their corresponding necessary conditions.
- 4) Use of a technique called noisy “channel” relaxation that makes the solution less sensitive to the initialization.

### A. Necessary Conditions on $\gamma_1$

Let us first define the function  $\tilde{\gamma}_1$  as

$$\tilde{\gamma}_1(x_0) \triangleq \gamma_1(x_0) + x_0 = x_1. \quad (6)$$

Without loss of generality, we will optimize with respect to  $\tilde{\gamma}_1$ . The cost we want to minimize is given by

$$J \triangleq \mathbf{E}[k^2\gamma_1^2(X_0) + (X_1 - \gamma_2(Y_2))^2]. \quad (7)$$

Using Bayes’ rule, the expected cost function can now be expressed as

$$\begin{aligned} J &= \int p(x_0, y_2|\tilde{\gamma}_1) F(x_0, \tilde{\gamma}_1(x_0), \gamma_2(y_2)) dx_0 dy_2 \\ &= \int p(x_0)p(y_2|x_0, \tilde{\gamma}_1) F(x_0, \tilde{\gamma}_1(x_0), \gamma_2(y_2)) dx_0 dy_2 \\ &= \int p(x_0)p(y_2|\tilde{\gamma}_1(x_0)) F(x_0, \tilde{\gamma}_1(x_0), \gamma_2(y_2)) dx_0 dy_2, \end{aligned} \quad (8)$$

where

$$F(x_0, x_1, \gamma_2(y_2)) = \left( k^2(x_1 - x_0)^2 + (x_1 - \gamma_2(y_2))^2 \right). \quad (9)$$

Since the integrand in (8) is positive for all values of  $x_0$ , it is clear that the optimization of  $\tilde{\gamma}_1$  (assuming  $\gamma_2$  is fixed) can be done individually for each  $x_0$ . A necessary condition for  $\tilde{\gamma}_1$  to be optimal is given by

$$\tilde{\gamma}_1(x_0) = \arg \min_{x_1 \in \mathbb{R}} \left( \int p(y_2|x_1) F(x_0, x_1, \gamma_2(y_2)) dy_2 \right) \quad (10)$$

for all  $x_0 \in \mathbb{R}$ .

### B. Necessary Conditions on $\gamma_2$

If we now assume that  $\gamma_1$  is fixed, we see that the the first term in (7) is a constant. The minimization of  $J$  with respect to  $\gamma_2$  is therefore equivalent to

$$\min_{\gamma_2(\cdot)} \mathbf{E}[(X_1 - \gamma_2(Y_2))^2], \quad (11)$$

which is the mean-squared error (MSE). It is well known that the MSE is minimized by the conditional expected value; hence,

$$\gamma_2(y_2) = \mathbf{E}[X_1|y_2] \quad (12)$$

for all  $y_2 \in \mathbb{R}$ , is a necessary condition for  $\gamma_2(y_2)$  to be optimal.

### C. Discretization

Although (10) and (12) would be possible to numerically evaluate for a particular  $x_0$  and  $y_2$ , respectively, they are impractical since the full representation of the functions is infinite-dimensional. To get around this problem we introduce the set

$$\mathcal{S}_L = \left\{ -\Delta \frac{L-1}{2}, -\Delta \frac{L-3}{2}, \dots, \Delta \frac{L-3}{2}, \Delta \frac{L-1}{2} \right\}, \quad (13)$$

where  $L \in \mathbb{N}$  and  $\Delta \in \mathbb{R}_+$  are two parameters that determine the number of points and the spacing between the points, respectively. Next, we impose the constraint  $x_1 \in \mathcal{S}_L$ , that is, the output of  $\tilde{\gamma}_1$  can only take one out of a finite number of values. In a similar way, the input to  $\gamma_2$  is discretized such that,

$$\gamma_2(y_2) = \tilde{\gamma}_2(\tilde{y}_2), \quad \tilde{y}_2 = Q_{\mathcal{S}_L}(y_2) \in \mathcal{S}_L, \quad (14)$$

where  $Q_{\mathcal{S}_L}(y_2)$  maps  $y_2$  to the closest point in the set  $\mathcal{S}_L$ .  $\gamma_2$  can now be stored in the form of a lookup table where each point in  $\mathcal{S}_L$  is associated with an output value. The approximation of the real space with  $\mathcal{S}_L$  can be made more and more accurate by decreasing  $\Delta$  and increasing  $L$ .<sup>1</sup> Finally, since  $X_0$  is still infinite-dimensional, we use Monte-Carlo samples of  $X_0$  to represent the input to  $\tilde{\gamma}_1$ .  $\tilde{\gamma}_1$  is now specified by evaluating

$$\tilde{\gamma}_1(x_0) = \arg \min_{x_1 \in \mathcal{S}_L} \sum_{\tilde{y}_2 \in \mathcal{S}_L} p(\tilde{y}_2|x_1) F(x_0, x_1, \tilde{\gamma}_2(\tilde{y}_2)) \quad (15)$$

for each of the Monte-Carlo samples that represent  $X_0$ . In a similar way,  $\tilde{\gamma}_2$  can be expressed as

$$\tilde{\gamma}_2(\tilde{y}_2) = \mathbf{E}[X_1|\tilde{y}_2], \quad (16)$$

for all  $\tilde{y}_2 \in \mathcal{S}_L$ , where the expectation with respect to  $X_0$  is evaluated by using the Monte-Carlo samples.

#### D. Design Algorithm

Given the above expressions for  $\tilde{\gamma}_1$  and  $\tilde{\gamma}_2$  it will be possible to optimize the system iteratively. We do this by keeping one part of the system fixed while we optimize the other part. One common problem with iterative techniques like the one suggested here is that the final solution will depend on the initialization of the algorithm. If the initialization is bad we are likely to end up in a poor local minimum.

In joint source-channel coding, one method that has proven to be helpful in counteracting this is noisy channel relaxation (NCR) [9], [8], [20], [15]. The idea of NCR is to change some parameter and first design a system for a completely different scenario with a simpler solution. The solution that is obtained is then used as initialization when designing for a scenario that is a bit closer to the true scenario. In joint source-channel coding, this is done by first designing a system for a channel with a lower signal-to-noise ratio (SNR) than the target SNR, which explains the name of the method. In the Witsenhausen setup, we have found that the ideas from NCR can be used as follows. Design a system for a high value of  $k$  first and then gradually decrease  $k$  until the desired value of  $k$  is reached. The reason to start with a high value of  $k$  is that the design algorithm will find a solution where  $\tilde{\gamma}_1(x_0) \approx x_0$  in this case (i.e.,  $\gamma_1(x_0) \approx 0$ ) independently of  $\gamma_2$ . The design procedure including the NCR part is given in Algorithm 1.

Each update on line 7 and 8 in Algorithm 1 will decrease the cost. Since the cost is lower bounded, it is clear that the

<sup>1</sup>While decreasing  $\Delta$ , one has to increase  $L$  to make sure that  $\max(x \in \mathcal{S}_L) = \Delta(L-1)/2$  does not decrease.

---

#### Algorithm 1 Design Algorithm

---

**Require:** Initial mapping of  $\tilde{\gamma}_2$ , the value  $k$  for which the system should be optimized and the threshold  $\delta$  that determines when to stop the iterations.

**Ensure:** Locally optimized  $\tilde{\gamma}_1$  and  $\tilde{\gamma}_2$ .

- 1: Let  $k' > k$ .
  - 2: **while**  $k' > k$  **do**
  - 3:   Decrease  $k'$  according to some scheme (e.g., linearly).
  - 4:   Set the iteration index  $i = 0$  and  $J^{(0)} = \infty$ .
  - 5:   **repeat**
  - 6:     Set  $i = i + 1$
  - 7:     Find the optimal  $\tilde{\gamma}_1$  by using (15).
  - 8:     Find the optimal  $\tilde{\gamma}_2$  by using (16).
  - 9:     Evaluate the cost function  $J^{(i)}$  according to (7).
  - 10:    **until**  $(J^{(i-1)} - J^{(i)})/J^{(i-1)} < \delta$
  - 11: **end while**
- 

algorithm will converge. It may happen that the algorithm converges to a local optimum, however, as will be seen in the following section the *local* optima we obtain are still better than any previously reported results.

## IV. RESULTS

### A. Implementation Aspects

For the evaluation of the design algorithm we have initially used  $L = 201$  levels and chosen  $\Delta(L) = 10\sigma/(L-1)$ . We have used 400000 Monte-Carlo samples in the final optimizations to represent  $X_0$ . Since it is known that the optimal  $\gamma_1$  is symmetric about origin [21], we have restricted  $\tilde{\gamma}_1$  to have this symmetry by generating only positive Monte-Carlo samples and thereafter reflecting the resulting  $\tilde{\gamma}_1$ -function for negative values of  $x_0$ .

To be able to compare our results to previously reported results, we have set  $\sigma = 5$  and  $k = 0.2$ . However, since we are using the NCR idea, we have initially used the value  $k' = 3$  and decreased it according to the series  $\{3, 2, 1.5, 1, 0.6, 0.4, 0.3, 0.2\}$ . Before running the design algorithm, we require  $\tilde{\gamma}_2$  to be initialized. However, due to the NCR this has little impact on the final solution and we have used the initialization  $\tilde{\gamma}_2 \equiv 0$ .

Once we have obtained the solution for  $k' = 0.2$ , we have increased the precision by expanding the number of points in the discrete set from  $L$  to  $L'$  and updated  $\tilde{\gamma}_2$  according to

$$\tilde{\gamma}_2^{(L')}(\tilde{y}_2) = \tilde{\gamma}_2^{(L)}(Q_{\mathcal{S}_L}(\tilde{y}_2)) \quad (17)$$

for all  $\tilde{y}_2 \in \mathcal{S}_{L'}$ . Thereafter the inner part of the design algorithm, that is, lines 4–10, have been run again to obtain a system optimized for the increased number of points  $L'$ . By repeating this refinement, the precision increases and the cost decreases as will be shown later. This method of refining the precision is similar to the one-way multigrid algorithm that is analyzed in [4]. The evaluations of (15) and (16) have been done using an exhaustive search, therefore, the run time is exponential in the number of levels  $L$ . An example

TABLE I  
FINAL COST FOR DIFFERENT SOLUTIONS ( $L = 12801$ )

Steps	Stage-1 Cost	Stage-2 Cost	Total Cost
Witsenhausen [21] <sup>a</sup>	0.40423088	0.00002232	0.40425320
Bansal and Bansar [3] <sup>a</sup>	0.36338023	0.00163460	0.36501483
Deng and Ho [5] <sup>a</sup>	0.13948840	0.05307290	0.19256130
Baglietto et al. [2]			0.1701
Lee et al. [16]	0.13188408	0.03542912	0.16731321
Li et al. [17]			0.1670790
This paper, 3-step	0.13493778	0.03201113	0.16694891
This paper, 3.5-step	0.13462186	0.03230369	0.16692555
This paper, 4-step	0.13484828	0.03207634	<b>0.16692462</b>

<sup>a</sup> Costs obtained from [16].

TABLE II  
COSTS FOR DIFFERENT PRECISIONS FOR THE 4-STEP SOLUTION

$L$	$M$	Stage-1 Cost	Stage-2 Cost	Total Cost
201	16	0.12104248	0.05764052	0.17868301
401	22	0.13015033	0.03883388	0.16898421
801	30	0.13530785	0.03200857	0.16731642
1601	56	0.13496610	0.03206243	0.16702853
3201	110	0.13486814	0.03208140	0.16694954
6401	210	0.13485895	0.03207070	0.16692966
12801	396	0.13484828	0.03207634	0.16692462

of the number of iterations that are required and the total computation time can be found in Appendix II.

### B. Numerical Results

During the first steps of the NCR  $k'$  is high. This means that the output of  $\tilde{\gamma}_1$  should follow the input closely to avoid large costs in the first stage. If continuous outputs were allowed, the output would be identical to the input. However, since we are working with a discretized system, only outputs from the set  $\mathcal{S}_L$  are possible. As  $k'$  reaches 0.4–0.6 the step behavior of the output appears. Depending on the realization of the Monte-Carlo samples we get either a 3.5-step mapping as shown in Figure 2 or a 4-step mapping as shown in Figure 3 (occasionally, a 3-step solution has occurred). The total costs for these solutions are stated in Table I. For ease of comparison, we have also included the costs of previously reported results. As can be seen, all our mappings have similar performance and all of them give lower costs than the previously reported lowest cost — 0.1670790 [17].

In Table II we show how the cost decreases as the number of points  $L$  is increased. The method we use to calculate the total cost as well as some notes on the accuracy can be found in Appendix I. The lowest cost we have achieved with our algorithm is 0.16692462. The mapping that achieves this cost is the 4-step mapping shown in Figure 3 with  $L = 12801$  points. Although the mapping contains four clear output levels it should be emphasized that each level is slightly sloped; this can be seen in Figure 4, where the first step has been zoomed in. It is reasonable to assume that as the precision (i.e.,  $L$ ) increases further, each step of the mapping will converge to a straight line that is slightly sloped.

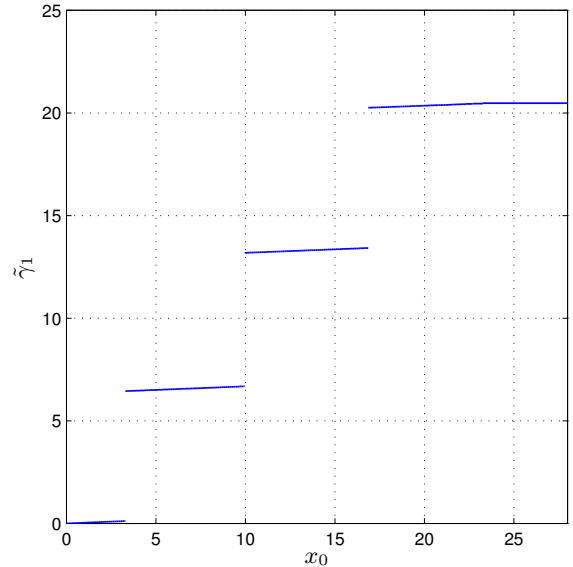


Fig. 2. 3.5-step solution ( $L = 12801$ )

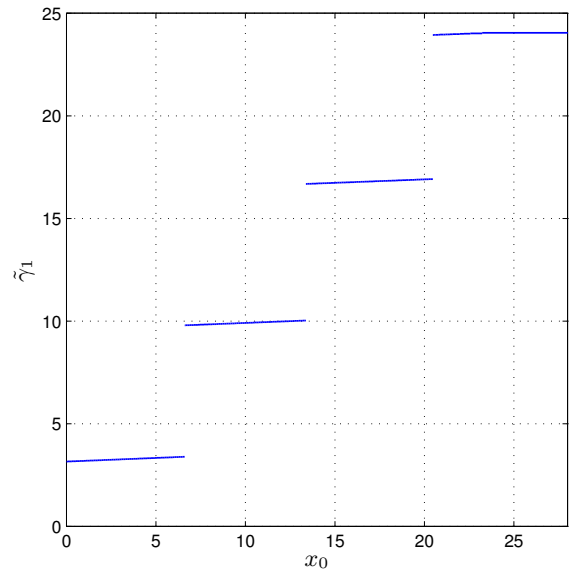


Fig. 3. 4-step solution ( $L = 12801$ )

## V. COMPARISON TO PREVIOUS RESULTS

In this section, we will compare the presented method with the previous methods. Besides the fact that the new method improves the optimized cost, there are further advantages compared to previous work:

- No structure is assumed for the decision functions. In [5] and [16], monotonicity of the decisions was assumed. The space of decisions is assumed to be a normed linear space in [2].
- A significant analytic/modeling work was performed before posing the optimization problem to be solved

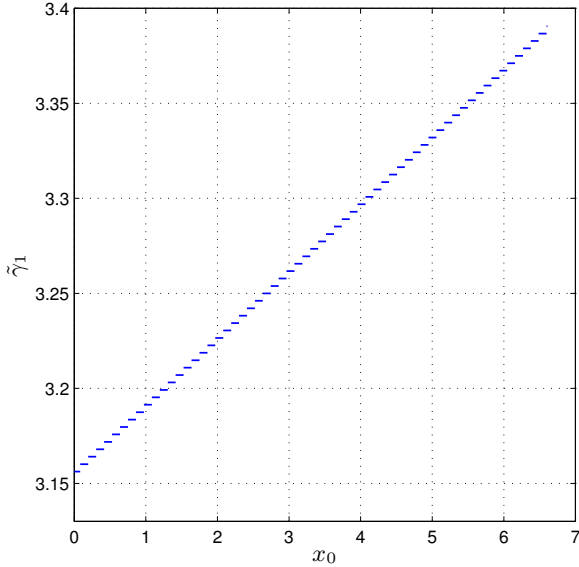


Fig. 4. Detailed view of the first step in the 4-step solution.

in [5], [16], and [2]. The first two require manual adjustments for the proper choice of interval values and signal levels, and the third requires some prior analysis to determine a constant “ $c$ ”. In [17], modeling work is needed in converting the problem into a potential game.

The method presented in this paper is fully automated and its advantages rely on the following:

- Discretization of the decision space.
- Iterative optimization between the different decision functions.
- A noisy channel relaxation technique that makes the solution insensitive to the initialization.

## VI. CONCLUSIONS

In this paper, we introduced a generic method of iterative optimization based on ideas from Source–Channel coding, that could be used to solve problems of the Witsenhausen Counterexample character. The numerical solution we obtain for the benchmark problem is of high accuracy and renders the lowest value known thus far, **0.16692462**.

Also, the design algorithm does not make any assumption on the structure of the policies — the solutions are allowed to have arbitrary shapes (within the restrictions imposed by the discretization). The results can therefore be seen as a confirmation that the step-shaped behavior is beneficial.

## VII. ACKNOWLEDGEMENT

The authors would like to thank the reviewers for their valuable and constructive feedback.

This work was supported in part by the Swedish Research Council and the European Commission through the FP7 project FeedNetBack.

## APPENDIX I CALCULATION OF THE TOTAL COST

In the design algorithm,  $\tilde{\gamma}_1$  is specified implicitly by, for each Monte-Carlo sample, storing the output symbol to which it is mapped. This representation is used when evaluating the cost during the iterations in the design algorithm. However, to evaluate the final total cost we need higher numerical accuracy. Therefore, the first step in calculating the total cost is to use the sample-based representation to find an explicit specification of  $\tilde{\gamma}_1$  given by

$$\tilde{\gamma}_1(x_0) = \alpha_i \in \mathcal{S}_L \quad \text{if } A_i \leq x_0 < A_{i+1}, \quad (18)$$

for  $i \in \{0, \dots, M-1\}$ , with  $A_0 = -\infty$  and  $A_M = \infty$ . That is, we transform the sample-based representation of  $\tilde{\gamma}_1$ , which is explicitly defined only for the Monte-Carlo samples, to a function which is defined for all real numbers. This representation makes it possible to numerically evaluate the integrals that are needed to find the total cost

$$\begin{aligned} J &= \mathbf{E}[k^2\gamma_1^2(X_0) + (X_1 - \gamma_2(Y_2))^2] \\ &= \underbrace{\mathbf{E}[k^2(\tilde{\gamma}_1(X_0) - X_0)^2]}_{=J_1} + \underbrace{\mathbf{E}[(\tilde{\gamma}_1(X_0) - \tilde{\gamma}_2(\tilde{Y}_2))^2]}_{=J_2}, \end{aligned} \quad (19)$$

where

$$\begin{aligned} J_1 &= \int_{x_0} p(x_0)k^2(\tilde{\gamma}_1(x_0) - x_0)^2 dx_0 \\ &= k^2 \sum_{i=0}^{M-1} \int_{A_i}^{A_{i+1}} p(x_0)(\alpha_i - x_0)^2 dx_0, \quad (20) \\ J_2 &= \int_{x_0} \sum_{\tilde{y}_2 \in \mathcal{S}_L} p(x_0, \tilde{y}_2)(\tilde{\gamma}_1(x_0) - \tilde{\gamma}_2(\tilde{y}_2))^2 dx_0 \\ &= \int_{x_0} \sum_{\tilde{y}_2 \in \mathcal{S}_L} p(x_0)P(\tilde{y}_2|x_0)(\tilde{\gamma}_1(x_0) - \tilde{\gamma}_2(\tilde{y}_2))^2 dx_0 \\ &= \sum_{i=0}^{M-1} \int_{A_i}^{A_{i+1}} p(x_0) \sum_{\tilde{y}_2 \in \mathcal{S}_L} P(\tilde{y}_2|\alpha_i)(\alpha_i - \tilde{\gamma}_2(\tilde{y}_2))^2 dx_0 \\ &= \sum_{i=0}^{M-1} \left\{ \sum_{\tilde{y}_2 \in \mathcal{S}_L} P(\tilde{y}_2|\alpha_i)(\alpha_i - \tilde{\gamma}_2(\tilde{y}_2))^2 \right\} \int_{A_i}^{A_{i+1}} p(x_0) dx_0, \end{aligned} \quad (21)$$

and

$$P(\tilde{y}_2|\alpha_i) = \begin{cases} \int_{-\infty}^{\tilde{y}_2+\Delta/2} p(w = y_2 - \alpha_i) dy_2 & \text{if } \tilde{y}_2 = -\Delta \frac{L-1}{2} \\ \int_{\tilde{y}_2-\Delta/2}^{\tilde{y}_2+\Delta/2} p(w = y_2 - \alpha_i) dy_2 & \text{if } \tilde{y}_2 = \Delta \frac{L-1}{2} \\ \int_{\tilde{y}_2-\Delta/2}^{\tilde{y}_2+\Delta/2} p(w = y_2 - \alpha_i) dy_2 & \text{otherwise} \end{cases} \quad (22)$$

All integrals have been calculated numerically using the Matlab function *quadl* with the tolerance specified to be  $t = 10^{-18}$ , which means that the absolute error of the result from *quadl* is not greater than  $t$ . All integrands are continuous and have a smooth behavior that should cause no problem for

quadr. To upper bound the total cost, we have upper bounded each integral by adding  $t$  to each individual result from quadr and reevaluated the total cost. In this way we have estimated the absolute error to be in the order of (or less than)  $10^{-11}$ . Matlab code for our calculations of the total cost, including our decision functions can be found in [1].

APPENDIX II  
COMPUTATION TIME

Table III and IV show the computation times for the design algorithm and the refinement process, respectively. The algorithms were implemented in C++ on a computer with an Intel Core2 Quad CPU running at 2.66 GHz.

TABLE III  
COMPUTATION TIME — ALGORITHM 1 ( $L = 201$ )

$k'$	# Iterations	Time
3	4	32 s
2	2	13 s
1.5	2	14 s
1	2	13 s
0.6	2	13 s
0.4	100	11 min
0.3	36	4 min
0.2	30	3 min
Total	178	20 min

TABLE IV  
COMPUTATION TIME — REFINEMENT

$L$	# Iterations	Time
401	44	15 min
801	83	98 min
1601	12	52 min
3201	3	49 min
6401	2	126 min
12801	2	501 min
Total		14 h

REFERENCES

[1] <http://www.ee.kth.se/~johk/witsenhausen/>.  
 [2] M. Baglietto, T. Parisini, and R. Zoppoli. Numerical solutions to the Witsenhausen counterexample by approximating networks. *IEEE Trans. Automatic Control*, 46(9):1471–1477, September 2001.

[3] R. Bansal and T. Basar. Stochastic teams with nonclassical information revisited: When is an affine law optimal? *IEEE Trans. Automatic Control*, 32(6):554–559, June 1987.  
 [4] C.S. Chow and J. N. Tsitsiklis. An optimal one-way multigrid algorithm for discrete-time stochastic control. *Automatic Control, IEEE Transactions on*, 36(8):898–914, 1991.  
 [5] M. Deng and Y.C. Ho. An ordinal optimization approach to optimal control problems. *Automatica*, 35:331–338, 1999.  
 [6] N. Farvardin and V. Vaishampayan. Optimal quantizer design for noisy channels: An approach to combined source–channel coding. *IEEE Trans. on Information Theory*, 33(6):827–838, November 1987.  
 [7] N. Farvardin and V. Vaishampayan. On the performance and complexity of channel-optimized vectorquantizers. *IEEE Trans. on Information Theory*, 37(1):155–160, January 1991.  
 [8] A. Fuldseth and T. A. Ramstad. Bandwidth compression for continuous amplitude channels based on vector approximation to a continuous subset of the source signal space. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3093–3096, Munich, Germany, April 1997.  
 [9] S. Gadhari and K. Rose. Noisy channel relaxation for VQ design. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2048–2051, May 1996.  
 [10] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer academic publishers, Dordrecht, The Netherlands, 1992.  
 [11] P. Grover, S. Y. Park, and A. Sahai. The finite-dimensional Witsenhausen counterexample. In *ConCom*, Seoul, Korea, March 2009.  
 [12] Y.C. Ho and T. S. Chang. Another look at the nonclassical information structure problem. *IEEE Trans. on Automatic Control*, 25(3), 1980.  
 [13] Y.C. Ho and K.C. Chu. Team decision theory and information structures in optimal control problems—part I. *IEEE Trans. on Automatic Control*, 17(1), 1972.  
 [14] Y.C. Ho, M. P. Kastner, and E. Wong. Teams, signaling, and information theory. *IEEE Trans. on Automatic Control*, 23(2), 1978.  
 [15] J. Karlsson and M. Skoglund. Optimized low-delay source–channel–relay mappings. *IEEE Trans. on Communications*, 58(5):1397–1404, May 2010.  
 [16] J. T. Lee, E. Lau, and Y.C. Ho. The Witsenhausen counterexample: A hierarchical search approach for nonconvex optimization problems. *IEEE Trans. Automatic Control*, 46(3):382–297, March 2001.  
 [17] N. Li, R. Marden, and J. S. Shamma. Learning approaches to the Witsenhausen counterexample from a view of potential games. In *IEEE Conference on Decision and Control*, pages 157–162, December 2009.  
 [18] S. P. Lloyd. Least Squares Quantization in PCM. *IEEE Trans. on Information Theory*, 28(2):129–137, March 1982.  
 [19] J. Max. Quantizing for minimum distortion. *IRE Trans. on Information Theory*, 6:7–12, March 1960.  
 [20] N. Wernersson, J. Karlsson, and M. Skoglund. Distributed quantization over noisy channels. *IEEE Trans. on Communications*, 57(6):1693–1700, June 2009.  
 [21] H. S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal on Control*, 6(1):138–147, 1968.  
 [22] K. A. Zeger and A. Gersho. Vector quantizer design for memoryless noisy channels. In *IEEE Internationell Conference on Communications*, pages 1593–1597, Philadelphia, USA, June 1988.