

# Optimization Approaches for Controller and Schedule Codesign in Networked Control

Stefano Longo \* Guido Herrmann \* Phil Barber \*\*

\* *Department of Mechanical Engineering, Faculty of Engineering, The University of Bristol, Queen's Building, University Walk, Bristol, BS8 1TR, UK, (Fax: +441179294423; tel. S. Longo: +441173317371; tel. G. Herrmann: +441179288100; e-mails: {s.longo, g.herrmann}@bristol.ac.uk).*

\*\* *Jaguar and Land Rover Research, W/2/021 Engineering Centre, Abbey Road, Whitley, Coventry, CV 4LF, UK.*

---

**Abstract:** We consider the offline optimization of a sequence for communication scheduling in networked control systems. Given a continuous-time Linear Quadratic Regulator (LQR) problem we design a sampled-data periodic controller based on the continuous time LQR controller that takes into account the limited communication medium and inter-sampling behavior. To allow for a Riccati equation approach, singularities in the weighting matrices and time-variance are accounted for using a lifting approach. Optimal scheduling can be obtained by solving a complex combinatorial optimization problem. Two stochastic algorithms will be proposed to find a (sub)optimal sequence and the associated optimal controller which is the result of a discrete algebraic Riccati equation for the given optimal sequence.

---

## 1. INTRODUCTION

In a Networked Control System (NCS), sensors, actuators and controllers are spatially distributed and interconnected via a shared communication medium. NCS increase modularity, flexibility and allow quick and easy maintenance at low cost. They are essential to automotive industry (see Leen and Heffernan [2002]), avionic systems (see Gwaltney and Briscoe [2006]), robots (see Oda et al. [2001], Göktaş et al. [1997]) and automated manufacturing systems (see Lian et al. [2000]) to reduce hardwiring and costs of installation and implementation.

The focus in this paper are contention-free protocols where the sequence of the control signal is solely dependent on the progression in time and therefore allows pre-planning of a particular periodic communication sequence at the design stage. The main problem we are solving is the following: *given a continuous-time infinite horizon Linear Quadratic Regulator (LQR) problem for a distributed system, find a scheduler with a fixed, periodic communication sequence and the corresponding sampled-data controller based on the continuous time LQR cost that takes into account the limited communication medium and inter-sampling behavior.* This problem was introduced in Rehfinder and Sanfridson [2004] and reconsidered in Ben Gaid et al. [2006]. Zhang and Hristu-Varsakelis [2006], Ionete and Çela [2006], Hristu-Varsakelis [2007] proved that communication sequences that preserve reachability and observability exist and have an upper bound on their period but the optimality of such sequences is not addressed.

The solution to the complex combinatorial optimization problem proposed by Rehfinder and Sanfridson [2004] is a heuristic algorithm based on a partition of the problem into three sub-problems. Only the single-channel case is considered i.e. only one control signal can be updated at any time tick. In Lu et al. [2003], the (sub)optimal sequence for a  $H_\infty$  control problem is solved by using another heuristic. In Ben Gaid et al. [2006], the LQ problem is translated into a mixed integer quadratic programming formulation and solved using a branch and bound based method. Lincoln and Bernhardsson [2002] uses a tree

pruning technique and optimality, for finite horizon problems, is proven if the number of pruned branches is kept small.

In this paper, our approach is: from a standard continuous time LQR problem, we first obtain the equivalent sampled-data representation. Then we model the limited communication channel as in Rehfinder and Sanfridson [2004], Lu et al. [2003], and merge the two models to obtain an augmented model which includes the dynamics of the scheduler. Since the pre-planned schedule is periodic, the resulting LQ problem will also be periodic. By using the lifting technique (see Chen and Francis [1995]), for a given sequence, this periodicity can be eliminated by creating a higher dimensional system and the solution to the problem will be given by solving a Discrete Algebraic Riccati Equation (DARE).

We exploit the features of two stochastic algorithms to solve the (combinatorial) optimization problem. The first one is a modification of the simple Genetic Algorithm (GA) (see Vosa [1999]) to a integer-coded GA (see Deb and Goyal [1997]). The second one is the Particle Swarm Optimization (PSO) algorithm proposed by Kennedy and Eberhart [1995] and readapted for a discrete-variable search space, (see Parsopoulos and Vrahatis [2002]). A third PSO based algorithm is also proposed. This exploits the benefits of partitioning the problem into smaller sub-problems resulting in being faster and more effective.

## 2. PROBLEM FORMULATION

In this section, we introduce the theoretical framework of limited communication for control. The framework follows the one of Rehfinder and Sanfridson [2004] and extends it to the multi-channel case of Ben Gaid et al. [2006].

### 2.1 Augmented system model

Consider a plant where the actuators are spatially distributed and the limited communication medium used for the actuator signals is represented by a shared bus. Only a limited number of actuators can be controlled and we assume that the actuator

inputs latch so that when the switch contact is opened the actuator holds its signal value (zero-order-hold). We assume that the spatially distributed plant is a linear time-invariant system described by

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad (1)$$

where  $x(t) \in \mathbb{R}^n$ ,  $u(t) \in \mathbb{R}^m$  and  $(A, B)$  is a controllable pair. The control input  $u(t)$  is a discrete input signal created by a zero-order-hold element.

The model in (1) has to be sampled with a periodic sampling interval  $h$  giving the sampled-data system:

$$x(jh + h) = \Phi x(jh) + \Gamma u(jh), \quad x(0) = x_0, \quad (2)$$

where  $\Phi$  and  $\Gamma$  are given in [Åström and Wittenmark, 1997, p.408] and  $j$  is the sampling instant. For simplicity, we will omit the sampling period  $h$  from now on.

The next step is to obtain a model for the scheduler that emulates the limited communication channel (see Rehinder and Sanfridson [2004], Ben Gaid et al. [2006]). From a practical view point, it is reasonable to assume that sequences are periodic.

*Definition 1.* If  $m$  is the number of actuators,  $m_{max}$  the maximum number of control signals that can be transmitted at any time tick  $t = ih$  and  $m_r(k)$ ,  $1 \leq m_r(k) \leq m_{max} \leq m$ , for all  $k$  is the reduced number of actuators controlled at time  $k$ , a  $p$ -periodic communication sequence  $\sigma = \{\sigma(0), \sigma(1), \dots, \sigma(p-1)\}$  is an ordered list of  $p$  vectors  $\sigma(k) = [\sigma_1(k) \ \sigma_2(k) \ \dots \ \sigma_{m_r(k)}(k)]^T$  where  $\sigma_i(k) \in \mathbb{N}$ ,  $1 \leq \sigma_i(k) \leq m$ , is the index of the actuator for which control communication is enabled at time tick  $k$ .  $\square$

*Definition 2.* The sequence of binary scheduling matrices is  $S = \{S(0), S(1), \dots, S(p-1)\}$  where  $S(k) := (s_{r,c}(k))_{m \times m}$  and  $s_{r,c}(k)$  are the elements of  $S(k)$  in the  $r^{th}$  row and  $c^{th}$  column. The elements  $s_{r,c}(k)$  satisfy:

$$s_{r,c}(k) = \begin{cases} 1 & \text{if } r = c = \sigma_i(k) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$\forall r, c = 1, 2, \dots, m; \quad i = 1, 2, \dots, m_r(k). \quad (4)$$

We also define  $\bar{S} = \{\bar{S}(0), \bar{S}(1), \dots, \bar{S}(p-1)\}$ , where  $\bar{S}(k) := (\bar{s}_{r,c}(k))_{m \times m}$ , as  $\bar{S}(k) = I - S(k)$  where  $I$  is the identity matrix.  $\square$

*Definition 3.* A  $p$ -periodic sequence  $\sigma$  that generates a sequence of matrices  $S$  such that

$$\text{rank} \left( \sum_{k=0}^{p-1} S(k) \right) = m \quad (5)$$

will be called feasible.  $\square$

Condition (5) ensures that each actuator control signal is updated at least once during the control sequence. This condition is required to guarantee controllability of the NCS as shown in Longo et al. [2009].

Having defined the  $p$ -periodic communication sequence and the respective scheduling matrices, the scheduler can be represented by

$$v(j) = \bar{S}(k)v(j-1) + S(k)\hat{u}(j), \quad v(0) = u_0, \\ k = \text{mod}(j, p) \quad (6)$$

where  $v(\cdot)$  is the limited control vector,  $\hat{u}(\cdot)$  is the full control vector and  $\text{mod}(\cdot, \cdot)$  is the modulo operator. The state matrix  $S(k)$  is  $p$ -periodic according to the scheduling policy. The plant model including the limited communication can now be modified as

$$x(j+1) = \Phi x(j) + \Gamma (\bar{S}(k)v(j-1) + S(k)\hat{u}(j)). \quad (7)$$

We let  $j = pl + k$  and merge the two system equations (7) and (6) to form the augmented system

$$\hat{x}(pl + k + 1) = \hat{\Phi}(k)\hat{x}(pl + k) + \hat{\Gamma}(k)\hat{u}(pl + k). \quad (8)$$

where the augmented states and matrices can be found in Ionete and Çela [2006]. The dynamics of the periodic scheduler will inevitably introduce a periodic augmented model.

## 2.2 Eliminate periodicity via lifting

To eliminate the time dependence in  $k$ , the lifting technique can be used since  $S(k)$  is periodic in  $p$ . This is described in Chen and Francis [1995] and used in Rehinder and Sanfridson [2004], Ben Gaid et al. [2006]. The aim here is to create a higher dimensional system in order to eliminate the dependence on  $k$ . The lifted system from (8) is <sup>1</sup>

$$\hat{x}(pl + p) = \bar{\Phi}_p \hat{x}(pl) + \bar{\Gamma}_p \bar{u}(pl) \quad (9)$$

where

$$\bar{u}(pl) = [\hat{u}(pl)^T \ \hat{u}(pl+1)^T \ \dots \ \hat{u}(pl+p-1)^T]^T, \\ \bar{\Phi}_p = \prod_{j=1}^p \hat{\Phi}(p-j), \quad \bar{\Gamma}_p = [G(0) \ G(1) \ \dots \ G(p-1)], \\ \text{and } G(k) = \prod_{j=1}^{p-k-1} \hat{\Phi}(p-j)\hat{\Gamma}(k). \quad (10)$$

## 2.3 Cost function

Consider the given continuous time, infinite horizon LQ problem:

$$J = \min_u \int_0^\infty x(t)^T Q_{c1} x(t) + u(t)^T Q_{c2} u(t) dt. \quad (11) \\ \text{s.t. (1)}$$

We assume that  $Q_{c1} \geq 0$  and  $Q_{c2} > 0$  are given as weights for a desirable ideal closed-loop response and that  $(A, Q_{c1})$  is an observable pair. The solution to the continuous-time control problem is given by the Continuous-time Algebraic Riccati Equation (CARE)

$$A^T P_c + P_c A - P_c B Q_{c2}^{-1} B^T P_c + Q_{c1} = 0. \quad (12)$$

The equivalent discrete cost function of (11) for the sampled system (2) is:

$$J = \min_u \sum_{j=0}^{\infty} \begin{bmatrix} x(jh) \\ u(jh) \end{bmatrix}^T \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix} \begin{bmatrix} x(jh) \\ u(jh) \end{bmatrix}, \quad (13) \\ \text{s.t. (2)}$$

where the expressions  $Q_1$ ,  $Q_2$  and  $Q_{12}$  are given in [Åström and Wittenmark, 1997, pp.411-412]. The cost associated with the augmented plant of (8) is

$$J = \min_{\hat{u}} \sum_{l=0}^{\infty} \sum_{k=0}^{p-1} \underbrace{\begin{bmatrix} \hat{x}(pl+k) \\ \hat{u}(pl+k) \end{bmatrix}^T \begin{bmatrix} \hat{Q}_1(k) & \hat{Q}_{12}(k) \\ \hat{Q}_{12}(k)^T & \hat{Q}_2(k) \end{bmatrix} \begin{bmatrix} \hat{x}(pl+k) \\ \hat{u}(pl+k) \end{bmatrix}}_{=\hat{Q}(k)}, \quad (14) \\ \text{s.t. (8)}$$

where the expression for  $\hat{Q}(k)$  is given in Ben Gaid et al. [2006]. The cost associated with the augmented lifted plant of (9) is

<sup>1</sup> When  $k = p - 1$ , the product is undefined. For this reason, to allow for this short notation, we need to define in the context of our paper  $\prod_{j=1}^i f(j) = 1$  for  $i < j$ .

$$J = \min_{\substack{\bar{u} \\ \text{s.t. (9)}}} \sum_{l=0}^{\infty} \left[ \begin{array}{c} \hat{x}(pl) \\ \bar{u}(pl) \end{array} \right]^T \underbrace{\left( \sum_{k=0}^{p-1} \left[ \begin{array}{cc} \bar{Q}_1(k) & \bar{Q}_{12}(k) \\ \bar{Q}_{12}(k)^T & \bar{Q}_2(k) \end{array} \right] \right)}_{\left[ \begin{array}{cc} \tilde{Q}_1 & \tilde{Q}_{12} \\ \tilde{Q}_{12}^T & \tilde{Q}_2 \end{array} \right]} \left[ \begin{array}{c} \hat{x}(pl) \\ \bar{u}(pl) \end{array} \right]. \quad (15)$$

where

$$\begin{aligned} \bar{Q}_1(k) &= \bar{\Phi}_k^T \hat{Q}_1(k) \bar{\Phi}_k, \\ \bar{Q}_2(k) &= \bar{\Gamma}_k^T \hat{Q}_1(k) \bar{\Gamma}_k + E^T \hat{Q}_{12}(k) \bar{\Gamma}_k \\ &\quad + \bar{\Gamma}_k^T \hat{Q}_{12}(k) E + E^T \hat{Q}_2 E, \\ \bar{Q}_{12}(k) &= \bar{\Phi}_k^T \hat{Q}_1(k) \bar{\Gamma}_k + \bar{\Phi}_k^T \hat{Q}_{12}(k) E, \\ \bar{\Phi}_0 &= I, \quad \bar{\Gamma}_0 = 0, \end{aligned}$$

$$E = [E(0) \ E(1) \ \dots \ E(p-1)], \quad E(i) = \begin{cases} I & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases}$$

(the matrix  $E$  extracts the particular control signal  $\hat{u}(pl+k)$ ).

It should be noticed that the control input dimension of the matrix  $\bar{\Gamma}_p$  has increased by a factor of  $p$  to  $mp$ . It is easily shown that there exists a binary matrix  $F \in \mathbb{R}^{mp \times \sum_k m_r(k)}$  with elements 0 and 1 only such that  $\text{rank}(\bar{\Gamma}_p F) = m$ ,  $\bar{\Gamma}_p F \in \mathbb{R}^{(m+n) \times \sum_k m_r(k)}$  and  $\text{rank}(F^T \bar{Q}_2 F) = \sum_{k=0}^{p-1} m_r(k)$ ,  $F^T \bar{Q}_2 F \in \mathbb{R}^{\sum_k m_r(k) \times \sum_k m_r(k)}$ , i.e.  $F^T \bar{Q}_2 F$  is non-singular. The final LQR time-invariant problem is

$$J = \min_{\substack{\bar{u} \\ \text{s.t. (9)}}} \sum_{l=0}^{\infty} \left[ \begin{array}{c} \hat{x}(pl) \\ \bar{u}(pl) \end{array} \right]^T \left[ \begin{array}{cc} \tilde{Q}_1 & \tilde{Q}_{12} F \\ F^T \tilde{Q}_{12}^T & F^T \tilde{Q}_2 F \end{array} \right] \left[ \begin{array}{c} \hat{x}(pl) \\ \bar{u}(pl) \end{array} \right], \quad (16)$$

where  $\bar{u}(pl) = F \bar{u}(pl)$ . With a given feasible sequence of length  $p$  the solution to (16) is given by the feedback controller

$$\begin{aligned} \bar{u}(pl) &= -F \bar{K} \hat{x}(pl) \\ &= -F (F^T \bar{\Gamma}_p^T \bar{P} \bar{\Gamma}_p F + F^T \tilde{Q}_2 F)^{-1} \\ &\quad (F^T \bar{\Gamma}_p^T \bar{P} \bar{\Phi}_p + F^T \tilde{Q}_{12}^T) \hat{x}(pl) \end{aligned} \quad (17)$$

where  $\bar{P}$  is the solution of the DARE

$$\begin{aligned} \bar{P} &= \bar{\Phi}_p^T \bar{P} \bar{\Phi}_p - (\bar{\Phi}_p^T \bar{P} \bar{\Gamma}_p F + \tilde{Q}_{12} F) (F^T \bar{\Gamma}_p^T \bar{P} \bar{\Gamma}_p F + \\ &\quad F^T \tilde{Q}_2 F)^{-1} (\bar{\Phi}_p^T \bar{P} \bar{\Gamma}_p F + \tilde{Q}_{12} F)^T + \tilde{Q}_1. \end{aligned} \quad (18)$$

Since we are considering the lifted system, only one Riccati equation needs to be solved.

Assuming the initial states  $\hat{x}(0)$  are random following a Gaussian process with unity variance, the expectation of the cost  $J$  subject to a random  $\hat{x}(0)$  is

$$J_E = E[J] = \text{trace}[\bar{P}]. \quad (19)$$

Moreover, the cost equivalent to the optimality function used in Rehinder and Sanfridson [2004] is:

$$J_O = \max \left\{ \lambda(P_c^{-1} \bar{P}) \right\} \quad (20)$$

where  $P_c$  is the solution of the CARE (12) and  $\bar{P}$  is the top left submatrix of  $\bar{P}$ , i.e.  $\bar{P} = \begin{bmatrix} \bar{P} & * \\ * & * \end{bmatrix}$ .

Finally, the optimization problem is: *find an optimal feasible sequence that will minimize the cost in (20)*. The associated feedback controller will be given by (17).

*Remark 1.* The solvability of the LQR problem is implied from controllability<sup>2</sup> of the lifted augmented system that also de-

<sup>2</sup> The more relaxed assumption of stabilizability could be used instead.

pends on the communication sequence and the observability of  $(A, Q_{c1})$ . A sufficient condition involving ‘feasible’ scheduling sequences (Definition 3) and a more general form of non-pathological sampling frequencies is given in Longo et al. [2009].  $\circ$

*Remark 2.* Although the cost function here is only defined for the LQR problem, the approach is readily extended to LQG (e.g. Zhang and Hristu-Varsakelis [2005]),  $H_2$  and  $H_\infty$  (e.g. Lu et al. [2003]) design.  $\circ$

### 3. COMBINATORIAL OPTIMIZATION

This problem falls into the area of combinatorial optimization and can be described by the tuple  $(\mathcal{S}, \mathcal{P}, \mathcal{F}, J_{E/O}, \min)$  where  $\mathcal{S}$  is the solution space of size  $N_S$  on which  $\mathcal{P}$  and  $J_{E/O}$  are defined,  $\mathcal{P}$  is the feasibility predicate (5) and  $\mathcal{F}$  is the set of feasible solutions of size  $N_F$ . The set of unfeasible solutions is  $\mathcal{U}$  and its size is  $N_U = N_S - N_F$  since  $\mathcal{F} \cup \mathcal{U} = \mathcal{S}$ .

#### 3.1 Analysis of properties

We will analyze some of the properties specific to the optimization problem and give some definitions.

The behavior of an optimization algorithm can be predicted by analyzing the fitness landscape that the cost function  $J_{E/O}$  gives rise to (see Reeves and Rowe [2003]). The number of local optima in the fitness landscape will be one important factor in the performance of the search algorithms. Even more important are the relative basins of attraction of the optima that will depend on the search strategy.

*Remark 3.* It is intuitive to assume that given a sequence  $\sigma = \{\sigma(0), \sigma(1), \dots, \sigma(p-1)\}$  any  $k$  circular shifting of  $\sigma$  will result in another sequence  $\sigma_k$  with the same cost of  $\sigma$ . This is argued in Rehinder and Sanfridson [2004] in what is called ‘the set of equivalence classes’. Although this is true when the plant consists of several decoupled and identical systems, it may not be the case for other situations (e.g. a decoupled plant with some stable and some unstable subsystems).  $\circ$

A visual landscape analysis can be performed by inspecting the solution space. Fig. 1 shows typical costs  $J_O$  for all the

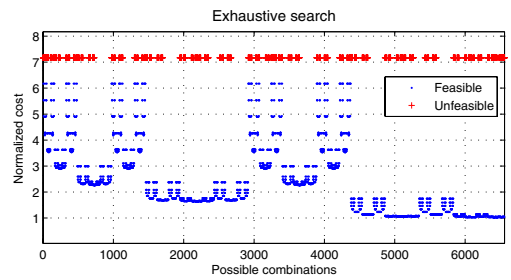


Fig. 1. Typical LG cost of every  $(3, 8)$ -Gray encoded solution.

$m(\sum_k m_r(k))$  permutations for a system with  $m = 3$ ,  $m_r(k) = 1$ , for all  $k$  and  $p = 8$ . All the possible combinations have been encoded using the  $(m, p)$ -Gray code algorithm of Guan [1998]. This allows to see the changes in cost for neighbor solutions as a Hamiltonian path in a generalized hypercube network in  $(\mathbb{Z}_m)^p$ . All the unfeasible solutions have been plotted with a symbolic high cost. The example shows the discontinuity of the search space in terms of neighbor solutions.

Let us now consider the feasibility issues. We only consider the actuator scheduling here but the analysis is similar and can be also applied to sensor scheduling. For the purpose of analysis,

we define a new type of sequence  $\bar{\sigma}$  that we call the augmented sequence. This is a sequence, the elements of which are the entries of the vectors  $\sigma(k)$  in  $\sigma$  distributed as follows

$$\begin{aligned} \bar{\sigma} &= \{\sigma_1(0), \dots, \sigma_{m_r(0)}(0), \sigma_1(1), \dots, \sigma_{m_r(1)}(1), \dots \\ &\quad \dots, \sigma_1(p-1), \dots, \sigma_{m_r(p-1)}(p-1)\} \\ &= \{\bar{\sigma}(0), \bar{\sigma}(1), \dots, \bar{\sigma}(\bar{p}-1)\} \end{aligned} \quad (21)$$

where  $\bar{p} = \sum_{k=0}^{p-1} m_r(k)$  is the augmented period. For feasibility analysis,  $\sigma$  and  $\bar{\sigma}$  are equivalent. Notice that if  $m_r(k) > 1$  for some  $k$  there will be equivalent sequences for  $\bar{\sigma}$ . These are those sequences where the order of the elements of  $\sigma(k)$  is interchanged.

*Observation 1.* If  $N_S = m^{\bar{p}}$  is the size of the solution space, the number of feasible and unfeasible permutations ( $N_{\mathcal{F}}$  and  $N_{\mathcal{U}}$  respectively) depends on  $m$  and  $\bar{p}$  and can be found analytically as follows

$$N_{\mathcal{F}} = N_S - N_{\mathcal{U}},$$

$$N_{\mathcal{U}} = \sum_{i=1}^{m-1} c(i) \binom{m}{i},$$

$$\text{where } c(i) = i^{\bar{p}} - \sum_{j=1}^i c(i-j) \binom{i}{i-j}, \quad c(0) = 0 \quad (22)$$

$$\text{and } \binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Fig. 2 shows the percentage of unfeasible augmented sequences

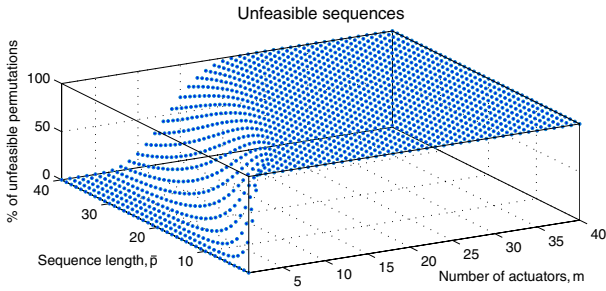


Fig. 2. Percentage of unfeasible sequences in  $\mathcal{S}$  as a function of  $m$  and  $\bar{p}$ .

as a function of  $m$  and  $\bar{p}$  for  $1 \leq \bar{p}, m \leq 40$ . As the length of the augmented sequence  $\bar{p}$  gets closer to the number of actuators  $m$ , the number of unfeasible solutions  $N_{\mathcal{U}}$  increases until, when  $m = \bar{p}$ , it becomes  $N_{\mathcal{U}} = m^{\bar{p}} - m!$  (and therefore  $N_{\mathcal{F}} = m!$ ). This is important when considering the sequence length  $p$  and channel capacity  $m_r(k)$  for a given number of actuators  $m$  at the design stage.

### 3.2 Optimization algorithms

The analysis in the previous section will be used to select suitable parameters for the optimization algorithms and enhance their performance. The algorithm run time is expressed here as the number of evaluations of the cost function as this is the predominantly expensive computation. An exhaustive search will run in  $O(m^{\bar{p}})$  time.

The highly discontinuous and unstructured search space (see Fig. 1) prevents gradient descent or local search algorithms to find satisfactory solutions. Complexity increases for  $m_r(k) > 1$ . The heuristics proposed by Rehbindler and Sanfridson [2004] greatly simplify the problem by braking it into three disjoint parts but it is only valid for  $m_r(k) = 1$  for all  $k$ .

*Definition 4.* For a sequence  $\bar{\sigma}$  (see Equation (21)), the elements of the vector  $\alpha = [\alpha_1 \alpha_2 \dots \alpha_m]$  are the number of times actuator  $i$  appears in  $\bar{\sigma}$ , therefore

$$\alpha_i = \sum_{k=1}^{\bar{p}} \delta(k), \quad \delta(k) = \begin{cases} 1 & \text{if } \bar{\sigma}(k) = i \\ 0 & \text{if } \bar{\sigma}(k) \neq i \end{cases}, \quad \sum_{i=1}^m \alpha_i = \bar{p}. \quad (23)$$

□

The algorithm of Rehbindler and Sanfridson [2004] allows a local search approach to be effective for small, single-channel problems but strong assumptions are made. In contrast, the optimization approach suggested in this paper exploits the characteristics of two stochastic algorithms in the attempt to find a fast, near to optimal solution and avoid combinatorial explosion as the problem size grows.

*Genetic Algorithm optimization* The simple Genetic Algorithm (GA) is modified by using an integer-coded chromosome (see Deb and Goyal [1997]) instead of the traditional binary-string chromosome of Vosa [1999], Reeves and Rowe [2003]. This type of coding is the most natural way to represent the augmented sequence  $\bar{\sigma}$  as each ‘gene’ will hold a value  $i$ ,  $1 \leq i \leq m$ ,  $i \in \mathbb{N}$  that directly corresponds to the actuator to be controlled. This eliminates the problems of generating binary unfeasible solutions due to the encoding when  $m \neq 2^i$  and will also eliminate the need for decoding. With a low mutation probability, this algorithm converges rapidly to a single (sub)optimal solution. A large population size will give more chances to find the global optimum but it will slow the search. A large mutation probability will encourage a wider exploration of the solution space but will also deteriorate the convergence rate.

*Particle Swarm Optimization algorithm* In the Particle Swarm Optimization (PSO) algorithm an optimal solution is found by a swarm of particles ‘flying’ through the solution space. Although the algorithm was originally proposed to find a solution in a continuous space (see Kennedy and Eberhart [1995]), it can be readapted for discrete optimization problems by truncating the real values to integers which does not affect significantly the search performance (see He et al. [2004]). The PSO algorithm is implemented in two substantially different ways and they will be referred as PSO1 and PSO2 algorithm.

In the PSO1 algorithm, the set of coordinates  $\rho^{(i)}$  of the position of particle  $i$  corresponds to a candidate augmented scheduling sequence i.e.  $\bar{\sigma}_i \equiv \{\rho^{(i)}\}$ . The advantages of this algorithm is its simplicity and that it requires only few parameters to be tuned. The core of the algorithm is in fact the velocity and position update equations for the  $i^{\text{th}}$  particle (Kennedy and Eberhart [1995]).

In the PSO2 algorithm, the search space is partitioned in two disjoint subproblems. The first subproblem is an optimization over  $\alpha$  and it is solved in the following way: the set of coordinates  $\rho^{(i)}$  of the position of particle  $i$  corresponds to the number of instances of control action i.e.  $\alpha^{(i)}$ . In other words the algorithm is used to optimize  $\alpha^{(i)} \equiv \{\rho^{(i)}\}$  rather than  $\bar{\sigma}_i \equiv \{\rho^{(i)}\}$ . The second subproblem is an optimization over the distribution of control actions. This can be indeed solved with the neighborhood search of Rehbindler and Sanfridson [2004] for single-channel problems but numerical analysis showed that it is highly inefficient especially for large problems. Instead, once an optimal  $\alpha$  is found, a sequence  $\bar{\sigma}$  can be constructed by maximizing the distance between control actions of the same actuator for the given  $m_r(k)$ . The distribution algorithm is:

**Step 1:**  $\bar{p} = \sum_{i=1}^m \alpha_i$ ,  $\bar{\sigma} = \{0_1, 0_2, \dots, 0_{\bar{p}}\}$   
**Step 2:**  $c = \max_i \{\alpha_i\}$ ,  $k = \{i : \alpha_i = c\}$   
**Step 3:**  $q_0 = \min_{j \in \{j : \bar{\sigma}(j)=0\}} j$   
**Step 4:**  $q = \operatorname{argmin}_{j \in \{j : \bar{\sigma}(j)=0\}} |q_0 - j|$ ,  $\bar{\sigma}(q) = k$ ,  $c = c - 1$   
**Step 5:**  $q_0 = q + \lfloor (\bar{p} - q) / c \rfloor$   
**if**  $c > 0$  **go to** Step 4  
**Step 6:**  $\alpha_k = 0$   
**if**  $\|\alpha\| \neq 0$  **go to** Step 2  
**Step 7:** Construct  $\sigma$  from  $\bar{\sigma}$  knowing the structure of  $\bar{\sigma}$  from (21).

By constraining the particle position to  $\alpha^{min} \leq \alpha_i \leq \alpha^{max}$  for all  $i$ , the sequence will always be within the feasibility region if  $\alpha^{min} \geq 1$  and it is easy to keep the period length  $p$  within desired limits.

#### 4. EXAMPLE

The effectiveness of the optimization approach will be demonstrated by some numerical results. The distributed system to be controlled is formed by  $m$  inverted pendulums arranged in a circle. Their poles are mechanically connected by ideal springs. The circle is large in radius so that the interaction due to the springs can be regarded as linear. The linearized dynamics of the system with the masses chosen to be  $M_i = \mu_i = 1$  (where  $M$  and  $\mu$  are the cart and bob masses respectively) for all  $i = 1, 2, \dots, m$ , can be described by (1) where

$$A = \begin{bmatrix} 0 & 1 & \dots & 0 & 0 & \dots & 0 & 0 \\ g/l_1 & -k_{1,2} - k_{m,1} & \dots & 0 & k_{1,2} & \dots & 0 & k_{m,1} \\ 0 & 0 & \dots & 0 & \dots & \dots & 0 & 0 \\ 0 & k_{1,2} & g/l_2 & -k_{2,3} - k_{1,2} & \dots & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \dots & 0 & 1 \\ 0 & k_{m,1} & 0 & 0 & \dots & g/l_m & -k_{m,1} - k_{m-1,m} & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & \dots & 0 \\ g/l_1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & g/l_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & g/l_m \end{bmatrix}, \quad x = \begin{bmatrix} \phi_1 \\ \dot{\phi}_1 \\ \phi_2 \\ \dot{\phi}_2 \\ \vdots \\ \phi_m \\ \dot{\phi}_m \end{bmatrix}, \quad (24)$$

$l_i$ ,  $i = 1, 2, \dots, m$  is the length of the  $i^{th}$  pendulum's pole,  $\phi_i$  is the pole's angle and  $k_{i,j}$  is the spring coefficient of the spring between pendulum  $i$  and  $j$ . The sampling period has been set to  $h = 0.01$  and  $K = k_{1,2} = k_{2,3} = \dots = k_{m,1}$ . When  $K = 0$  (decoupled systems case), the time constants of the pendulum systems will depend on the length of the stick  $l_i$  according to  $\tau_i = \sqrt{l_i/g}$ . By varying  $l_i$ , we can alter the system's dynamics and this, together with different choices of weight matrices, will be used to show how different sequences are assigned to different system's requirements. Furthermore, the actuators share the communication medium used to control the cart position.

##### 4.1 Results

**Optimal sequences** In this example  $K = 0$ ,  $m = 3$ ,  $m_r(k) = 1$  for all  $k$ ,  $\tau_1 = \tau_2 = 0.32$ ,  $\tau_3 = 3.2$ , and  $Q_{c1} = Q_{c2} = \operatorname{diag}(10, \dots, 10)$ . This is the case where two systems have smaller time constants than the other one. We first run the PSO2 algorithm with  $\alpha^{min} = 1$  and  $\alpha^{max} = 4$ . The optimal sequence was found for  $p = 5$ . The reason for keeping  $p$  small is to allow a direct comparison with an exhaustive search. The GA and PSO1 algorithm have been used to find an optimal sequence with  $p = 5$ . The sequences returned by the three algorithms have been shown in Fig. 3 in comparison to their location with the entire feasible solution space. The three algorithms returned different solutions with the GA having

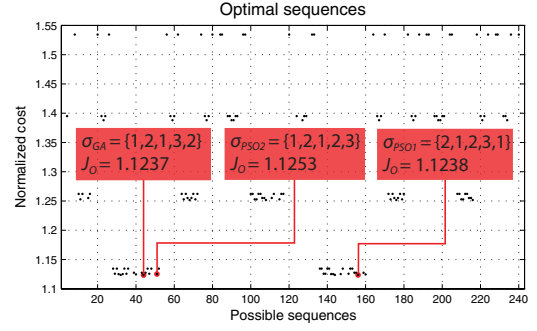


Fig. 3. Optimal sequences in comparison to the solution space.

found the optimal one. However, the sequences found by the PSO1 and PSO2 algorithms are close to the optimal. More time slots are allocated to the systems with higher requirement. Notice that  $\alpha_i^{GA} = \alpha_i^{PSO1} = \alpha_i^{PSO2}$  for all  $i$ , and therefore different costs are to be attributed to different distribution of control actions.

The same set up was used with  $m_r(k) = 2$ , for all  $k$  (multi-channel case). The optimal sequences found with the corresponding costs were

$$\sigma_{GA} = \sigma_{PSO1} = \left\{ \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix} \right\}, \quad \text{with } J_O = 1.003$$

$$\sigma_{PSO2} = \left\{ \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix} \right\}, \quad \text{with } J_O = 1.003.$$

Both sequences are in fact equivalent and produce the same sequence of scheduling matrices  $S$  (hence the cost is the same). The lower cost achieved compared to the previous (single-channel) example is due to higher channel capacity  $m_r(k)$ .

**Performance comparison** To demonstrate the algorithms performance, we first compare them with the exact set up as in Reh binder and Sanfridson [2004] and then we use a larger scale, more realistic example. For the first example the algorithms have been tested for the two identical decoupled systems with weight matrices

$$Q_{c1} = \begin{bmatrix} 1.72 & 0 \\ 0 & 0 \end{bmatrix}, \quad Q_{c2} = \begin{bmatrix} 2000000 & 0 \\ 0 & 0 \end{bmatrix},$$

for a desired time constant for pendulum 1 of 0.5s and pendulum 2 of 0.015s. The left plot of Fig. 4 shows the run time

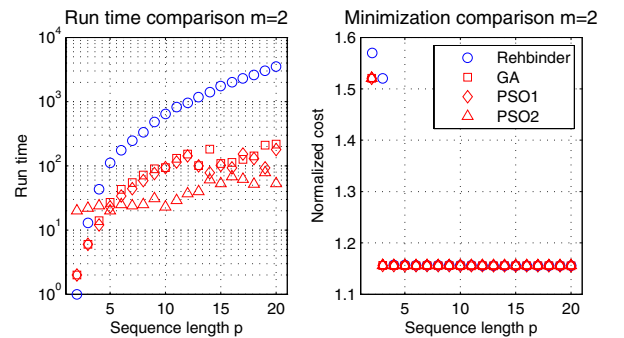


Fig. 4. Run time comparison between the minimization algorithms and respective minimum cost for  $m = 2$ .

between the algorithm in Reh binder and Sanfridson [2004] (called 'Reh binder') and the three algorithms proposed here. Apart from the trivial sequence of length 2, for any increasing length, the time required to find an optimal solution for the Reh binder algorithm is much larger compared to the one of



## 5. CONCLUSION

the other algorithms. The three algorithms proposed here on the other hand, after an initial increase in run time, are not as affected by the length of the sequence. The right plot of Fig. 4 gives the measure  $J_O$  of the quality of the solution found. Although it is not clear from the plot, the PSO1 algorithm and GA found equal or better solution than the Reh binder algorithm for any sequence length. The results do not exactly match the ones in Reh binder and Sanfridson [2004] because of Remark 3.

For the second example, the system was chosen so that  $m = 8$ ,  $K = 0.1$  and

$$Q_{c1} = \text{diag}(1.72, 0, 250, 0, 250, 0, 1.72, 0, 1.72, 0, \\ 250, 0, 250, 0, 1.72, 0),$$

$$Q_{c2} = \text{diag}(1, \dots, 1)$$

(eight coupled systems with different control specifications). The cost function used this time is  $J_E$  in (19). The results are shown in Fig. 5. The run time of the Reh binder algorithm

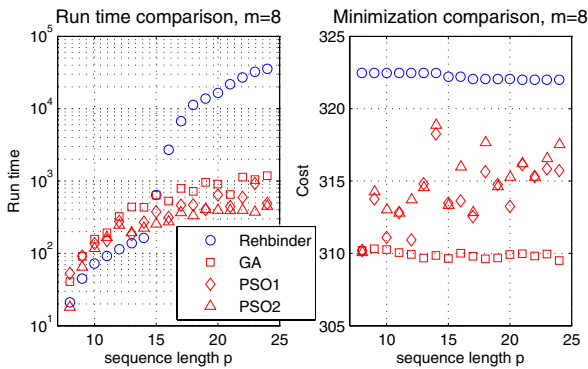


Fig. 5. Run time comparison between the minimization algorithms and respective minimum cost for  $m = 8$ .

becomes very large for  $p > 15$ . The algorithms proposed here showed to solve the problem in reasonable time, where the PSO2 algorithm is still the fastest. The GA, for all  $p$ , returns better solutions than any other algorithm.

### 4.2 Discussion

The PSO2 algorithm has the advantage of simultaneously solving two subproblems (optimization over  $\bar{p}$  and  $\alpha$ ) with an effective global search algorithm and it eliminates the need to solve the third subproblem. The search in the PSO2 algorithm is confined within the feasibility region only, ignoring the potentially large unfeasible space. Disadvantages of the PSO2 algorithm include the assumption that sequences obtained by the distribution algorithm of Section 3.2.2 are optimal.

The GA and PSO1 algorithm eliminate any assumption on the structural properties and solve the problem as a whole. The worst case run time will be  $O(N_p N_c)$  where  $N_p$  is the population size and  $N_c$  is the number of iterations until convergence. For the GA and PSO1 algorithm there is no optimization over the sequence length. Of course an exhaustive search over  $\beta_i \leq \bar{p} \leq \bar{p}_{max}$  (where  $\bar{p}_{max}$  is the maximum allowed sequence length) would be possible. If the ratio  $N_U/N_{\mathcal{F}}$  is high (see Fig. 2), sequences as offsprings (for the GA) or position (for the PSO1 algorithm) have a high probability to evolve or move into an unfeasible one and, as unfeasible sequences are rejected, fast convergence is prevented.

The GA is less sensitive to parameter changes which suggests that, for this type of problems GAs are more suited. This can be explained by the fact that GAs naturally work with discrete variables while it is possible that the forced discretization of the PSO algorithms deteriorate their search characteristics.

In this paper, we considered the joint controller and static schedule optimization. The problem was solved using the GA and the PSO algorithm. Some fitness landscape analysis showed the complexity of the search space and justified the choice of parallel search algorithms. Their performance seemed to be superior when compared to existing results especially for large scale problems. In general, the use of stochastic search methods to solve scheduling problems resulted to be effective in terms of quality of solutions found, computational time and simplicity of implementation.

### REFERENCES

- K. Åström and B. Wittenmark. *Computer-Controlled Systems*. Prentice-Hall, 1997.
- M. Ben Gaid, A. Çela, and Y. Hamam. Optimal integrated control and scheduling of networked control systems with communication constraints: application to a car suspension system. *IEEE Trans. on Control Systems Technology*, 14(4):776–787, 2006.
- T. Chen and B. Francis. *Optimal Sampled-data Control Systems*. Springer, London, 1995.
- K. Deb and M. Goyal. Optimizing engineering designs using a combined genetic search. *7th International Conf. on Genetic Algorithms*, pages 512–528, 1997.
- F. Göktas, J. Smith, and R. Bajcsy. Telerobotics over communication networks. In *Proc. of the IEEE Conf. on Decision and Control*, volume 3, pages 2399–2404, 1997.
- D. Guan. Generalised gray codes with applications. *Proc. National Science Council ROC(A)*, 22(6):841–848, 1998.
- G. Gwaltney and J. Briscoe. Comparison of communication architectures for spacecraft modular avionics systems. Technical report, NASA/TM-2006-214431, 2006.
- S. He, E. Prempain, and Q. Wu. An improved particle swarm optimiser for mechanical design optimization problems. *Engineering Optimization*, 36: 584–605, 2004.
- D. Hristu-Varsakelis. On the period of communication policies for networked control systems, and the question of zero-order holding. In *Proc. of the 46<sup>th</sup> IEEE Conf. on Decision and Control*, pages 38–43, 2007.
- C. Ionete and A. Çela. Structural properties and stabilization of NCS with medium access constraints. In *Proc. of the 45<sup>th</sup> IEEE Conf. on Decision and Control*, pages 1141–1146, 2006.
- J. Kennedy and R. Eberhart. Particle swarm optimisation. *IEEE International Conf. on Neural Networks*, 4:1942–1948, 1995.
- G. Leen and D. Heffernan. Expanding automotive electronic systems. *IEEE Computer*, 35(1):88–93, 2002.
- F. Lian, J. Moyné, and D. Tilbury. Implementation of networked machine tools in reconfigurable manufacturing systems. In *Proc. of the 2000 Japan-USA Symposium on Flexible Automation, Ann Arbor, MI*, 2000.
- B. Lincoln and B. Bernhardsson. LQR optimisation of linear system switching. *IEEE Trans. on Automatic Control*, 47(10):1701–1705, 2002.
- S. Longo, G. Herrmann, and P. Barber. Controllability, observability in networked control. To appear in the *6<sup>th</sup> IFAC symposium on Robust Control Design*, 2009.
- L. Lu, L. Xie, and M. Fu. Optimal control of networked systems with limited communication: a combined heuristic and convex optimization approach. In *Proc. of the 42<sup>nd</sup> IEEE Conf. on Decision and Control*, volume 2, pages 1194–1199, 2003.
- M. Oda, T. Doi, and K. Wakata. Tele-manipulation of a satellite mounted robot by an on-ground astronaut. In *Proc. of the IEEE Int. Conf. on Robotics & Automation*, pages 1891–1896, 2001.
- K. Parsopoulos and M. Vrahatis. Recent approaches to global optimisation problems through particle swarm optimisation. *Natural computing*, 1:235–306, 2002.
- C. Reeves and J. Rowe. *Genetic Algorithms: Principles and Perspectives*. Kluwer Academic Publishers, 2003.
- H. Reh binder and M. Sanfridson. Scheduling of a limited communication channel for optimal control. *Automatica*, 40(3):491–500, 2004.
- M. Vosa. *The Simple Genetic Algorithm. Foundations and Theory*. The MIT Press, 1999.
- L. Zhang and D. Hristu-Varsakelis. LQG control under limited communication. In *Proc. of the 44<sup>th</sup> IEEE Conf. on Decision and Control*, pages 185–190, 2005.
- L. Zhang and D. Hristu-Varsakelis. Communication and control co-design for networked control systems. *Automatica*, 42(6):953–958, 2006.