# COMPUTATIONAL BARRIERS IN LINEAR CONTROL SYSTEM DESIGN

R. Ayyagari, P. Kavitha
NIT Trichy, India 620 015
rkalyn@nitt.edu

**Abstract**: In this paper we argue that a fresh understanding of the computational issues of the problem is imperative given the numerous traditional applications of fixed-order controllers in process control as well as the emergence of newer applications such as formations of unmanned vehicles. We propose a generic framework to design fixed-order controllers that stabilize single-input single output plants. The proposed approach is based on the idea of choosing select parameters of the controller as random, and the rest as deterministic. We use a randomized algorithm to obtain the randomized parameters, which enables us to design general fixed-order controllers (including PID controllers). Subsequently, we use a deterministic technique to obtain the deterministic parameters. As an alternative to existing methods which use linear programming, we develop algorithms based on the solution of a set of linear equations by matrix inversion. A detailed complexity analysis is carried out, showing that the proposed algorithm has polynomial-time complexity. We demonstrate our methodology using two application examples.

## 1. INTRODUCTION

For the past decade there have been several results related to the computational difficulty of controller design problems such as designing a fixed order output feedback controller without any specific structural assumption. The source of these computational difficulties lie in the NP-hardness (see [1]-[2] for a discussion of complexity issues), which was, in some sense, ignored earlier. Interesting results have been reported recently in [3]-[5].

For controllers with very special structure such as PID or lead-lag a number of useful methods ([6] - [11]) are now available. On the other hand, a totally different approach based on randomization is developed in [12] for static output feedback. This method is applicable to the problem under attention. However, the classical paradigm argues that not all the parameters need to be randomized due to the special structure of the problem.

In this paper, we attempt to bring in the merits of both the perspectives and develop a, sort of, general methodology to design fixed order stabilizing controllers for SISO plants. This methodology is based on the idea of partitioning the controller parameters into two classes - deterministic and random. We use a randomized algorithm for the design of randomized parameters, which enables us to design a general fixed-order controller. Next, we use a deterministic technique to design the deterministic parameters. Since this second part considers the specific structure of the controller, we believe and hence argue that the methodology leads to computationally efficient algorithms.

## 2. NOTATION

We now introduce the notation used in this paper. We consider a SISO plant of the form:

$$P(s) = \frac{N_P(s)}{D_P(s)}$$

We study a fixed-order controller of the form

$$C(s) = \frac{N_C(s)}{D_C(s)}$$

w.l.g. we write C(s) as

$$C(s) = \frac{X(s^2) + sY(s^2)}{Z(s^2) + sV(s^2)}$$

Where X (.) etc are polynomials containing only even powers of s. These polynomials are of the form:

$$X(s^2) = \theta_0 + \theta_2 s^2 + \cdots + \theta_n X s^{nX}$$

$$Y(s^2) = \alpha_0 + \alpha_2 s^2 + \cdots + \alpha_n Y s^{nY}$$

$$Z(s^2) = \beta_0 + \beta_2 s^2 + \cdots + \beta_n s^{nz}$$

$$V(s^2) = \mu_0 + \mu_2 s^2 + \cdots + \mu_n s^{nV}$$

We assume that the coefficients of X ($s^2$) are deterministic, and the remaining coefficients are randomized. Thus, we have

$$n_\theta = \frac{nX}{2} + 1$$

$$n_\eta = \frac{nY + nZ + nV}{2} + 3$$

as the number of deterministic and random parameters respectively. Further, we define

$$\theta = [\theta_0 \cdots \theta_{nX}]^T$$

and

$$\eta = [\propto_0 \cdots \alpha_{nY} \beta_0 \cdots \beta_n \mu_0 \cdots \mu_{nY}]^T$$

with Θ and N as the sets of possible values of θ and η respectively.

The controller may now be thought of as C(s,θ), and the corresponding closed-loop polynomial of negative feedback connection is given by

$$\begin{aligned} p(s,\theta) &= N_p(s)N_c(s) + D_p(s)D_c(s) \\ &= p_0(s) + p_1(s)X(s^2) \qquad (1) \end{aligned}$$

If the deterministic parameters θ are on the boundary of the desired stabilizing set, then the system is marginally stable. We call a real number ω a *critical frequency* if for some θ the system is marginally stable.

By direct calculation, we may readily see that the number $n_f$ of critical frequencies is bounded as

$$n_f \le \frac{deg\,p_0(s)\,deg\,p_1(s) - \min\{deg\,p_0(s), deg\,p_1(s)\}}{2} + 2$$

For each critical frequency $\omega_i$, using eqn. (1) we obtain a hyper plane of the form

$$\varphi^T(\omega_i)\theta = v(\omega_i)$$

and for all the critical frequencies we get a square system

$$\varphi\theta = v \qquad (2)$$

We note that the solution θ, if it exists, gives a candidate marginal stabilizer. If, indeed, θ thus obtained is a marginal stabilizer we proceed to find a stabilizing controller. We also note that if the values of randomized parameters are fixed, the set of deterministic parameter values corresponding to stabilizing controllers enjoys some convexity property which can be exploited for efficient computation of the deterministic parameter values θ. Hence, our approach is to make use of randomized algorithms to first fix the randomized parameters, and then compute the deterministic parameters.

The principal objective of this paper is to design stabilizing controllers. We go ahead computing these controllers in the following sections.

## 3. RANDOMIZED ALGORITHMS

In this section we consider the use of randomized algorithms, which are known to be effective for many deterministically difficult problems within systems and control, see, e.g., [13]-[15].

We say that a randomized parameter value η is feasible if there exists $\theta \in \Theta$ such that C(s, θ, η) is a controller that stabilizes the system. Here, we present a randomized algorithm to find such a value η. For this purpose, we assign a probability distribution **P** to the set **N**, which is the bounding set of parameters η. Let δ and ε be any positive numbers less than unity and define

$$N_1 = \left\lceil \frac{\ln(1/\delta)}{\ln(1/(1-\varepsilon))} \right\rceil$$

Now we propose the following algorithm.

**Algorithm 1**:

```
1. for  i = 1 : N₁ do
     begin
2.       draw a sample $\eta^{(i)} \in$ N
           according to P;
3.       if $\eta^{(i)}$  is feasible then return;
     end
```

We observe that the feasibility check of Step 3 requires $\eta^{(i)}$ generated at Step 3 and a suitable which may be provided by the deterministic Algorithm 3 presented in the next section. The performance of the algorithm above is guaranteed by the following theorem, which is an immediate consequence of the results in [14].

**Theorem 1**: Suppose that the measure P(A) is greater than ε. Then, the probability that no $\eta^{(i)}$ , i = 1, 2, · · · ,$N_1$, provided by Algorithm 1 is feasible is less than δ.

Here, we let **A** denote the set of all feasible η in **N** and **P(A)** its measure according to **P**. An estimate of **P(A)** is provided in Theorem 2, based on the well-known Chernoff bound.

It is interesting to note that the maximum number of samples depends only on ε and δ. It may also be noted that line 3 of the algorithm can be carried out in polynomial time. Although the complexity to execute Line 2 depends on **N** and **P**, it is usually polynomial in the dimension $n_\eta$; for example, when **N** is box-shaped and **P** is the uniform distribution. This choice is used in many practical applications.

Next, we present another randomized algorithm to evaluate the measure **P(A)**. We choose positive numbers ε and δ to be smaller than unity and define

$$N_2 = \left\lceil \frac{1}{2\varepsilon^2} \ln\frac{2}{\delta} \right\rceil$$

**Algorithm 2:**

```
1. set $N_s = 0$
2. for i = 1 : $N_2$ do
      begin
```

```
3.        draw a sample  $\eta^{(i)} \in \mathbf{N}$
           according to P;
4.      if $\eta^{(i)}$ is feasible
          then set $N_s = N_s + 1$;
     end
```

Here, $N_s$ counts the number of feasible among the samples.

**Theorem 2**: The probability that

$$\left|\frac{N_s}{N_2} - P(A)\right| > \epsilon$$

holds is less than δ

## 4. DETERMINISTIC ALGORITHMS

In [7] it was shown that the set of all stabilizing PID controllers with fixed proportional gain is a finite union of convex polygons. We now generalize this and state the following theorem.

**Theorem 3**: Suppose that a randomized parameter vector η is selected according to Algorithm 1. Then, the set of all deterministic parameter vectors θ that stabilize the plant is either empty or is a union of a finite number of polyhedral sets.

**Proof:** We have earlier established in section 2 that the set of stabilizing $\theta_s$ has the boundary within the union of finite number of hyper planes.

**QED**

From Algorithm 1, it is now clear that the number $N_{MI}$ of square systems (2) that need to be computed is

$$N_{MI}(n_f, n_\theta) = \frac{n_f\,!}{n_\theta\,!\,(n_f - n_\theta)!}$$

It is not difficult to see that, once the controller is completely designed, the overall computational complexity is a polynomial in $n_f$.

We now state without proof a result regarding stabilization of the controller parameters.

**Theorem 5**: Let p(s, $\theta^{(i)}$) be the polynomial defined in (1) corresponding to $\theta^{(i)}$ computed in (2). There exists a marginal stabilizer if and only if there exists $\theta^{(i)}$, i = 1, 2, · · · ,$N_{MI}$, such that p (s, $\theta^{(i)}$) has its zeros within the closed left half plane.

That a candidate $\theta^{(i)}$ is a marginal stabilizer can be checked by means of Routh's test.

Remark: The marginally stabilizing controller parameter vector $\theta^{(i)}$, if it exists, is a vertex of a polyhedral set of stabilizing controllers. In this case, the $n_\theta$ rows of the corresponding matrix $\psi_i$ and the elements of $V_i$ define some of the hyperplanes generating the boundary of a polyhedral set of stabilizing controllers.

Suppose that θ is a marginally stabilizing parameter. Further, assume that all the zeros of p(s, θ) on the imaginary axis are simple. We may consider one imaginary zero $j\omega_i$ and study how it moves when we perturb θ by $\Delta\theta$. Since we need moving $\Delta\theta$ all imaginary zeros inside the left half of the plane, we consider to solve

$$\begin{bmatrix} Re\,\dfrac{\partial z_1}{\partial \theta_0} & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ Re\,\dfrac{\partial z_{n_\theta}}{\partial \theta_0} & \cdots & \dfrac{\partial z_{n_\theta}}{\partial \theta_{2(n_\theta - 1)}} \end{bmatrix} \begin{bmatrix} \Delta\theta_0 \\ \vdots \\ \Delta\theta_{2(n_\theta-1)} \end{bmatrix}$$

$$= \begin{bmatrix} -1 \\ 0 \\ -1 \end{bmatrix} \quad (3)$$

After we obtain the desired $\Delta\theta$, we consider a parameter $\theta+\Delta\theta$ for a positive α. Although a small α gives a stabilizing controller, a larger α can as well be used. Typically, we may find α using bisection method. If the imaginary zeros are not simple, we might go for another round of randomization.

In closing this section, we summarize the proposed algorithm which looks for a stabilizing controller when a randomized parameter $\eta^{(i)}$ is determined according to Algorithm 1 or 2.

**Algorithm 3:**

```
1. construct ψ and v for given $\eta^{(i)}$;
2. for j = 1 : $N_{MI}$ do
     begin
3.      compute $\theta^{(j)}$
4.      if $\theta^{(j)}$ is a marginal stabilizer
          then
     begin
5.      compute $\Delta\theta$ using (3)
6.      if a stabilizing parameter
        $\theta^{(j)} + \alpha\Delta\theta$ is found
        then stop;
     end
end
```

## 5. EXAMPLES

Let us first consider [Fujisaki et. al.]

$$P(s) = \frac{(s+1)(16s+1)(s^2-s+1)}{s(s-1)(s-90)(4s^2-s+1)}$$

and a second order controller

$$C(s) = \frac{\theta_0 + \alpha_0 s + \theta_2 s^2}{\beta_0 + \mu_0 s + \beta_2 s^2}$$

where $\theta_0$ and $\theta_2$ are the deterministic parameters $\theta_0$ and $\theta_2$, and the rest are randomized ones.

Having taken $\varepsilon = \delta = 0.01$, we have $N_1 = 459$ and $N_2 = 26492$. The controllers are

$$C_m(s) = \frac{-0.9831 - 0.7568s - 2.7546s^2}{1 - 0.3136s - 0.8570s^2}$$

and

$$C_s(s) = \frac{-0.7772 - 0.7568s - 2.4861s^2}{1 - 0.3136s - 0.8570s^2}$$

Keeping everything same as before, we now proceed to design a $H_\infty$ controller. Fig. 1 shows the sensitivity of stabilizing controller.

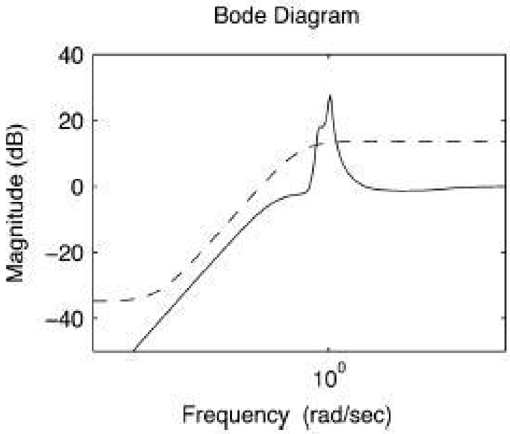

Fig -1

And, fig. 2 shows the sensitivity of $H_\infty$ controller is shown below.

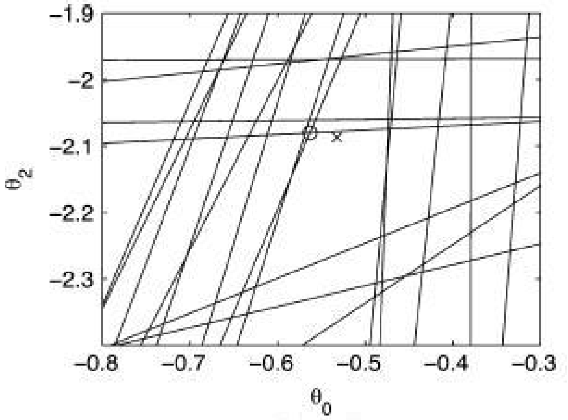

Fig - 2

## 6. CONCLUSIONS

In this paper, we studied fixed order stabilization of single-input single-output plants. We presented several randomized and deterministic algorithms for solving two different problems: stabilization and $H_1$ performance of a fixed plant. A detailed complexity analysis is also provided together with two examples. The key idea behind the proposed algorithm was to employ even-odd structure of the controller and to classify the design variables into randomized and deterministic parameters. Then, we used randomized algorithms for the former parameters and developed efficient algorithms for the latter. This idea is considered to be effective in various control problems.

Subsequent research will be carried on along several directions. In particular, we plan to extend the results of this paper to stabilization of plants affected by a delay, and to classes of multi-input multi-output systems.

## 7. ACKNOWLEDGEMENTS

This research work is carried out as a part of the UKIERI project “Towards Smart Reliable and Adaptable Air Vehicles” sanctioned by the British Council to the second author, along with University of Leicester, UK, IISc Bangalore, IIT Bombay, and NAL Bangalore. Travel and registration for the conference have been obtained from the project grant. The authors express their sincere thanks to UKIERI authorities, colleagues in the project, and
the Director NIT Trichy.

## 8. REFERENCES

[1] V. L. Syrmos, C. T. Abdallah, P. Dorato, and K. Grigoriadis, “Static output feedback – A survey,” Automatica, 33, 125 (1997).

[2] M. Fu and Z.-Q. Luo, “Computational complexity of a problem arising in fixed order output feedback design,” Syst. Control Lett., 30, 209 (1997).

[3] D. Henrion, M. . ˇ Sebek, and V. Kuˇcera, “Positive polynomials and robust stabilization with fixed-order controllers,” IEEE Trans. Automat. Control, 48, 1178 (2003).

[4] C. W. J. Hol and C. W. Scherer, “A sum-ofsquares approach to fixed-order H1 synthesis,” in Positive Polynomials in Control,(ed. D. Henrion and A. Garulli) (Springer-Verlag, Berlin, Germany) 45 (2005).

[5] E. N. Gryazina & B. T. Polyak, “Stability regions in the parameter space:D-decomposition revisited,” Automatica, 42, 13

(2006).

[6] K. J. Astrom and T. Agglund, PID Control – Theory, Design and Tuning, 2nd ed. Research Triangle Park, NC: Instrument Society of America, (1995).

[7] A. Datta, M.-T. Ho, and S. P. Bhattacharyya, Structure and Synthesis of PID Controllers, Springer-Verlag, London, U.K. (2000).

[8] "Special Issue on PID 2006," IEEE Control Syst. Mag., 26(1), (2006).

[9] J. Ackermann and D. Kaesbauer, "Stable polyhedra in parameter space," Automatica, 39, 937 (2003).

[10] M. T. Soylemez, N. Munro, and H. Baki, "Fast calculation of stabilizing PID controllers," Automatica, 39, 121 (2003).

[11] F. Blanchini, A. Lepschy, S. Miani, and U. Viaro, "Characterization of PID and lead/lag compensators satisfying given H1 specifications," IEEE Trans. Automat. Control, 49(5), 736 (2004).

[12] M. Vidyasagar and V. D. Blondel, "Probabilistic solutions to some NP-hard matrix problems," Automatica, 37, 397 (2001).

[13] R. Motwani and P. Raghavan, Randomized Algorithms. Cambridge Univ. Press, Cambridge, U.K (1995)
[14] R. Tempo, G. Calafiore, and F. Dabbene, Randomized Algorithms for Analysis and Control of Uncertain Systems, Springer-Verlag, London, UK, (2005).

[15] Y. Fujisaki, F. Dabbene, and R. Tempo, "Probabilistic design of LPV control systems," Automatica, 39 1323 (2003).

[16] W. A. Malik, S. Darbha, and S. P. Bhattacharyya, "A linear programming approach to the synthesis of fixed-structure controllers," IEEE Trans. Automat. Control, 53, 1341 (2008).

[17] M.-T. Ho, A. Datta, and S. P. Bhattacharyya, "Robust and non-fragile PID controller design," Int. J. Robust Nonlin. Control, 11, 681(2001).