GENETIC APPROACH ON LQ WEIGHTING MATRICES SELECTION AND ITS APPLICATION IN THE VLS ATTITUDE CONTROL

Alexandro Garro Brito
Instituto de Aeronáutica e Espaço. Pça. Mal. Eduardo Gomes, 50. Vila das Acácias. São José dos Campos - SP. CEP: 12228-904. Brazil
alegbrito@yahoo.com.br

Waldemar de Castro Leite Filho
Instituto de Aeronáutica e Espaço. Pça. Mal. Eduardo Gomes, 50. Vila das Acácias. São José dos Campos - SP. CEP: 12228-904. Brazil
waldemar@iae.cta.br

Abstract. *A genetic methodology to select Q and R weighting matrices on linear-quadratic design is presented. The method uses an optimization process on these matrices such that many closed loop requirements, of the time and frequency domains, can be treated . This approach is being proposed to VLS (Brazilian Satellite Launcher) attitude control design.*

Keywords: *Genetic algorithms, attitude control, optimization*

1. Introduction

The linear-quadratic approach is a minimization process where the cost function weights the states and control signal by using the proper Q and R matrices (Maciejowski, 1989). The LQ state feedback controller gives an excellent stability margins for closed loop. However, the transient response is highly dependent on the choice of Q and R, and there are not analytic methods that relates performance requirements with these matrices, directly. So, the most used way to find them is a trial and error procedure based in some prior informations about the system.

In this work it will be presented a method to obtain the weighting Q and R matrices using an optimization method based on genetic algorithms. The objective is to find them such that some response requirements are achieved. Then, it is possible to combine the stability conditions provided by LQ design with constraints on time and frequency domains.

The use of genetic algorithms in controllers tuning is not new. It was applied already in on-line tuning, by the minimization of a time-domain cost function (Jones and Oliveira, 1995) and *Integral Squared Error* index (Porter and Jones, 1992). More sophisticated methods, like neural network tuning (Omatu and Deris, 1996) and adaptive generalized minimum variance control (Mitsumura et. al., 1997), was proposed too. However, it does not seem useful for linear parameter varying systems, similar to VLS attitude control.

The proposed methodology, named GA LQ design, will be compared against the traditional LQ design, that was performed for early flights. The GA LQ optimization could lead to an improvement on closed loop response.

2. VLS attitude control

The VLS is a launch vehicle with 4 solid propellant stages. It has a orbit insertion capability around 750km and can transport up to 185 kg of useful load.

Most of the model coefficients are variable according to the flight time, because the fuel consumption, variations and the stage loss. A previous knowledge about these variations is very important for trajectory and control design. Besides, the controller and bending mode rejection filters must be variable such that the stability is guaranteed.

For control system design, a rigid body model for the launcher is used (Ramos, Leite Filho and Moreira, 2003), as stated in 3^{rd} order T.F. (1)

$$\frac{\dot{\Theta}(s)}{B_z(s)} = \frac{-\mu_\beta s^2 - \left(\frac{\mu_\beta z_\alpha - \mu_\alpha z_\beta}{u}\right) s}{s^3 + \left(\mu_q + \frac{z_\alpha}{u}\right) s^2 + \left(\frac{\mu_q z_\alpha}{u} - \mu_\alpha\right) s + \frac{\mu_\alpha g}{u}} \tag{1}$$

where $\beta_z(t) = \mathcal{L}^{-1}\{B_z(s)\}$ is actuator deflection and $\theta(t) = \mathcal{L}^{-1}\{\Theta(s)\}$ is the attitude output.

The VLS control system is composed by a PI controller with an angular velocity loop, such that it performs like the PID configuration presented in Fig. 1. The bending modes and notch filter are represented too.

Fixed PID gains for all flight are not usual since the rigid body parameters are varying with time. Then, it is adopted the *frozen poles methodology*, where a controller, that is designed for a critical time, produces a pole location which must be maintained along the flight.

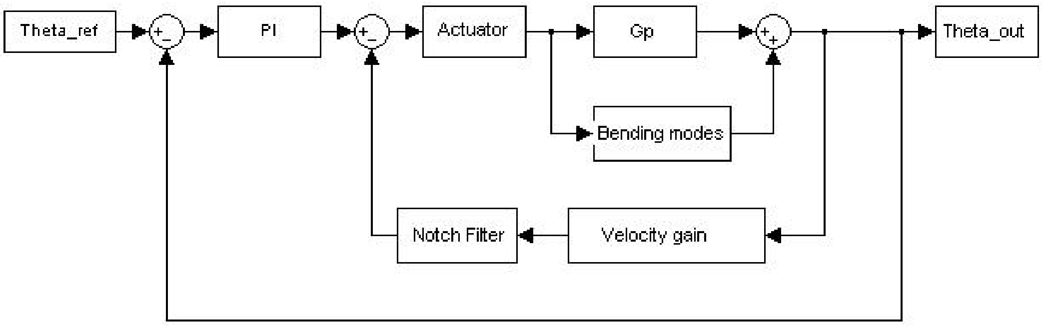

Figure 1. Complete VLS control system.

3. LQ approach

For design purpose, a simplified rigid body model of VLS is used. Assuming that the velocity u is very high and the M_q term can be neglected, the T.F (1) simplifies to (2) below,

$$G_P = \frac{-\mu_\beta s}{s^2 - \mu_\alpha}. \tag{2}$$

Because $\mu_\beta < 0$ and $\mu_\alpha > 0$ the VLS launcher is a minimum phase unstable model.

Taking this simplified model and neglecting the bending modes and filters of complete VLS control system, the closed loop which is used on controller design is presented in Fig. 2.

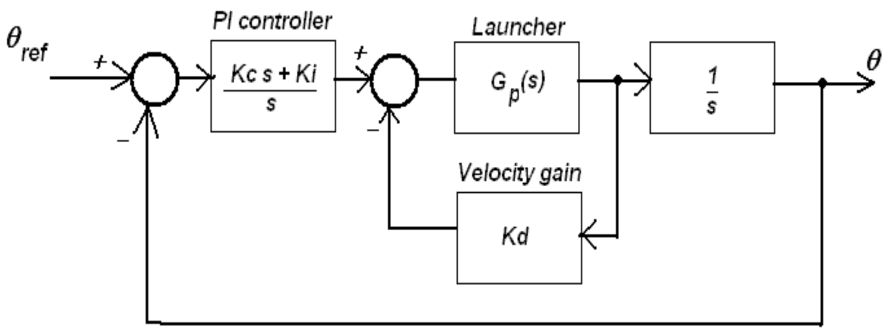

Figure 2. Simplified VLS control system.

The design begins with the search of a linear-quadratic controller for the Fig. 2 in a critical time of flight. This critical situation occurs when one of the launcher poles reaches the biggest positive real part, which means that the system has its worst stability condition.

The closed loop can be represent by

$$\begin{bmatrix} \ddot{\theta}(t) \\ \dot{\theta}(t) \\ \dot{c}(t) \end{bmatrix} = \begin{bmatrix} 0 & \mu_\alpha & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}(t) \\ \theta(t) \\ c(t) \end{bmatrix} + \begin{bmatrix} -\mu_\beta \\ 0 \\ 0 \end{bmatrix} \beta(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \theta_{ref}(t), \tag{3}$$

where $c(t) = \int [\theta_{ref}(t) - \theta(t)]dt$ and

$$\beta(t) = \begin{bmatrix} -K_d & -K_c & K_i \end{bmatrix} \begin{bmatrix} \dot{\theta}(t) \\ \theta(t) \\ c(t) \end{bmatrix} + K_c \theta_{ref}(t). \tag{4}$$

The optimum LQ problem is based on finding a controller that minimizes the follow cost function (5)

$$J = \int_0^\infty [z^T(t)Qz(t) + \beta(t)^2 R]dt, \tag{5}$$

where $z(t)$ is the state vector in (3); Q and R are weighting matrices on the states and input, respectively.

The optimization problem above has an exact solution, which can be obtained by solving the *Algebraic Riccati Equation* (Maciejowski, 1989). So, if it is possible to obtain a semidefinite-positive symmetric matrix W such that

$$A^TW + WA - BR^{-1}B^TW + Q = 0 \tag{6}$$

where

$$A = \begin{bmatrix} 0 & \mu_\alpha & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \text{ and } \quad B = \begin{bmatrix} -\mu_\beta \\ 0 \\ 0 \end{bmatrix}, \tag{7}$$

then the optimum state feedback controller is

$$K = R^{-1}B^TW. \tag{8}$$

After one evaluates the LQ optimization for the critical time, it is obtained the PID gains that lead to the follow transfer function (9) of Fig. 2,

$$G_{cl} = \frac{\Theta(s)}{\Theta_{ref}(s)} = \frac{-\mu_\beta(K_c s + K_i)}{s^3 - K_d\mu_\beta s^2 - (\mu_\alpha + \mu_\beta K_c)s - K_i\mu_\beta}, \tag{9}$$

that could be represented by the 3^{rd} order T.F. (10)

$$G_{cl} = \frac{\Theta(s)}{\Theta_{ref}(s)} = \frac{K(s + \eta p_0)}{(s + p_0)(s^2 + 2\zeta\omega_d s + \omega_d^2)}. \tag{10}$$

So it is possible to find the closed loop pole locations determined by ζ, ω_d and p_0 at this time.

The *frozen poles concept* consists on maintain in this pole locations in other instants of flight. Then the PID gains that lead to these pole locations can be calculated by (11), in a specific time (Ramos, Leite Filho and Moreira, 2003)

$$\begin{array}{l} K_c = -(\mu_\alpha + 2\zeta\omega_d p_0 + \omega_d^2)/(\mu_\beta) \\ K_i = -(p_0\omega_d^2)/(\mu_\beta) \\ K_d = -(p_0 + 2\zeta\omega_d)/(\mu_\beta) \end{array} . \tag{11}$$

The LQ approach has excellent stability margins, at least in the critical time. Its temporal performance is very dependent of the choice of Q and R weighting matrices. In early flights, this choice was searched by a trial and error procedure. The best empirical values for Q and R were

$$Q = \begin{bmatrix} 0.1 & & \\ & 1 & \\ & & 0.2 \end{bmatrix} \quad R = 0.4. \tag{12}$$

However, these matrices lead to poor transient performance. Then, it is important to adopt an iterative procedure when choosing of Q and R matrices such that the LQ approach used on VLS design associates temporal requirements. This methodology is presented on next sections.

4. GA concepts

The **Genetic Algorithms** are optimization processes based on evolution ideas taken of the nature. Then, the GA's are characterized by the evolution of an initial set of solutions, named *population*, according to stochastic rules that lead the actual population to the next, in a *generation* sequence.

The basic GA operations are:

i crossover: it combines the informations contained in two or more elements, such that new solutions are created. This operator is useful to guide the population for a possible global minimum, after some generations.

ii mutation: a new solution is created by using a stochastic rule on the informations of an element. The mutation operator guarantees the diversity of solution set. Then, new regions in the search region can be explored.

iii selection: some elements are replicated and continue to the next generation according to a fitness function. The others are discarded.

iv elitism: if the best solution is not selected to the next generation it can be inserted by replacing another element, which is chosen randomly.

After each generation a stop test is performed. The algorithm is finished case the solutions are sufficiently close of the minimum. Otherwise, new random elements are added to maintain constant the population size and the optimization process goes on. Fig. 3 presents a flowchart of GA optimization technique.

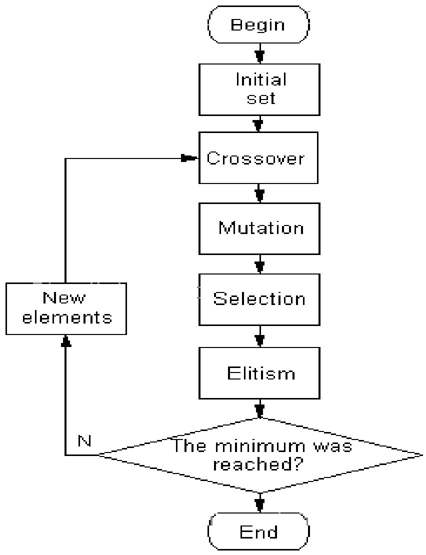

Figure 3. Flowchart of GA optimization technique

5. QR search

The QR search is made by using a GA, as discussed above. For this purpose, some assumptions are important.

Each element in the population is composed by the entries of Q and R matrices. Only diagonal matrices are considered for Q since it gives a more physical insight. For example, a element E_i is represented by

$$E_i = [\ q_{11} \quad q_{22} \quad q_{33} \quad r\ ]. \tag{13}$$

The value of the entries are limited in a range, which is constant in all optimization process. The GA operates on these entries such that the cost function reaches its minimum value. The initial set, $\mathcal{E} = \{E_i \ \forall \ i = 1, \ldots, n\}$is chosen randomly. The number of elements in this population must be big, such that the search region is represented properly.

The optimization is realized in an instant of flight. In general, the preferred instant is the one when the launcher has its worst stability condition - it is supposed that in other situations the vehicle has better stability margins. At the chosen time of flight, a LQ design is made for each element on the initial set, according to framework discussed in LQ approach section. After, a step response simulation is performed in each closed loop to obtain the transient response.

The cost function, which is used to evaluate the time domain performance of each element, must be defined such that the most important requirements are weighted properly. Herein, it is applied a mono-objective optimization, where the result of the cost function is a single value. Multi-objective optimization concepts are not considered in (Takahashi, 2004). For the VLS case, the cost function should consider the following aspects:

- minimizing the percent overshoot, PO, since it is more seriously affected requirement on the LQ approach. Without an optimization process, the VLS percent overshoot is bigger than specified on the vehicle conception;
- limiting the rise time, t_r, on a proper range. Very fast temporal response could excite the bending modes and this could make the launcher unstable.
- limiting the maximum value of settling time, t_s, such that the system is sufficiently fast.

Then, the optimization problem for VLS case is minimizing the percent overshoot in the restrict region limited by rise time and settling time ranges above - many other formulations are possible, but this gives best results. The functional requirements were created as explained. The rise time function, f_r, is defined as

$$f_r = \begin{cases} 1000t_r + 800 & \text{if} \quad t_r < t_{r-min} \\ 0 & \text{if} \quad t_{r-min} < t_r < t_{r-max} \\ 1000t_r - 1400 & \text{if} \quad t_r > t_{r-max} \end{cases} . \tag{14}$$

For the settling time, the f_s is defined as

$$f_s = \begin{cases} 0 & \text{if} \quad t_s < t_{s-max} \\ 1000t_s - 9900 & \text{if} \quad t_s > t_{s-max} \end{cases} . \tag{15}$$

The final cost function is

$$F_{cost} = PO + f_r + f_s. \tag{16}$$

The elements are sorted according to its final cost function values. After the genetic operations, the new generation is produced and the cycle goes on. When a small set of solutions reaches a local minimum, a stop criterion is set and the algorithm finishes.

6. Results

As it was exposed above, the optimization is made only in the instant with the worst stability condition. By the using the simplified model of VLS with frozen parameters in this time, it is possible to find Q and R for all the initial population in the GA optimization.

The search region for Q and R was limited such that each E_i entry in (13) is contained in $[0, 1]$. The constraints on *settling time* and *rise time* are based in the performance requirements necessary for the VLS closed loop. For security reasons, explicit values for this constraints and percent overshoot are not presented. However, weighted indexes are defined for each requirement to provide an efficient analysis of the results.

The *Percent Overshoot Index*, POI, is defined by

$$POI = \frac{PO(t)}{PO_{req}} \tag{17}$$

where $PO(t)$ is the percent overshoot for a step input in frozen instant t, and PO_{req} is the required percent overshoot for the closed loop. This value is defined by the proper performance for the launcher control design.

In the same way, the *Settling Time Index*, STI, and the *Rise Time Index*, RTI, are defined respectively by

$$STI = \frac{t_s(t)}{t_{s-max}} \tag{18}$$

and

$$RTI = \frac{t_r(t)}{t_{r-min}}. \tag{19}$$

The GA optimization of (16) constrained by search region above, leads to the optimum Q and R matrices

$$Q_{opt} = \begin{bmatrix} 0.798 & & \\ & 0.635 & \\ & & 0.129 \end{bmatrix} \quad R_{opt} = 0.258 \tag{20}$$

Using these matrices in LQ design, one could obtain ζ, ω_d and p_0 in (10). The complete set of controller gains is determined maintaining the parameters above for the rest of flight and computing K_c, K_i and K_g (11).

Now, a discussion about adopting (12) or (20) QR matrices in LQ design is performed. In Fig. 4 is presented the *Percent Overshoot Index* for both cases along the flight time. When (20) are used, the percent overshoot is smaller than PO_{req} for all instants. Only in a small interval, QR matrices in (12) give best performance.

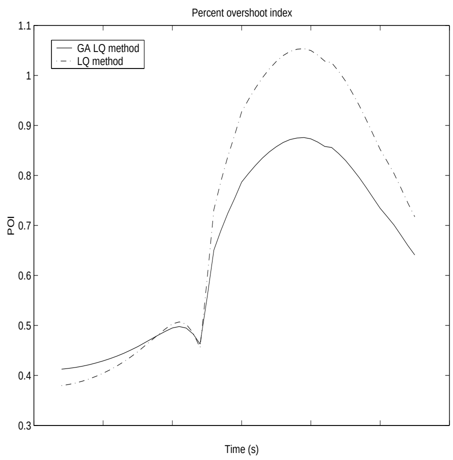

Figure 4. Percent Overshoot Index, POI, × flight time (omitted by security). Note that the GA LQ method presented maintains the percent overshoot smaller than the LQ method in almost all instants.

The rise time is analyzed in Fig. 5. The closed loop response for (12) matrices is very fast. This is not advisable, since the bending modes could be excited. When applying the optimum matrices the transient response is slower - the rise time is above the minimum recommended limit for all the flight. The maximum recommended limit for the rise time index is around 2.14.

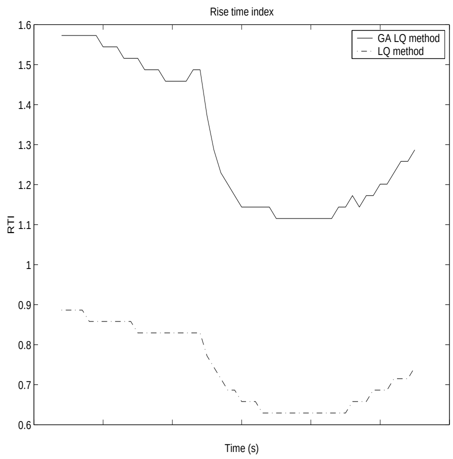

Figure 5. Rise Time Index, RTI, $\times$ flight time (omitted by security). The LQ method becomes the closed loop very fast, since RTI<1. Proper performance is achieved when AG LQ method is adopted.

The strongest requirement on settling time is about its maximum value, as it was shown in (15), such that the system response is not very slow. In Fig.(6) is presented the settling time index for both methods. As the GA LQ method prioritizes the percent overshoot optimization, it is natural that the settling time becomes bigger. However it is possible to maintain a good performance by using a proper constraint at its maximum value.

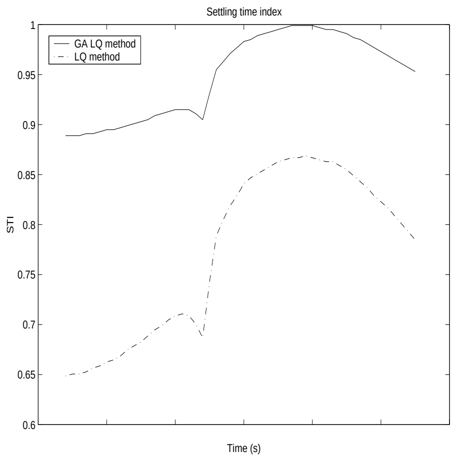

Figure 6. Settling Time Index, STI, $\times$ flight time (omitted by security). The optimization on percent overshoot leads the settling time becomes bigger, in general. Then, it is expected that the settling time in LQ method with optimized QR matrices is bigger than those in (12). However, the maximum settling time in those case is less than the required on design

7. Conclusions

In this work it was shown a strategy to find optimum Q and R matrices for LQ controller design. The common search methods for these matrices depends of an accurate knowledge of process and are based on trial and error procedure. By using a optimization algorithm in the search, a big variety of requirements could be approached.

For VLS case, it is used a Genetic Algorithm. This method is proper to find a global minimum in an unknown search region, since that it can explore accurately with the population set. Other traditional strategies like *Search Directions Algorithms* (Takahashi, 2004) could stop when it reaches a local minimum that can not correspond to desired solution. The GA optimization is applied to search QR according to temporal response requirements - minimizing the percent overshoot constrained by limits on settling time and rise time. As it was shown, adopting this procedure the closed loop performance is improved. Other requirements could be included in the optimization process by proper modifications in cost function.

8. References

Jones, A. H., Oliveira, P. B. M., 1995, "Genetic auto-tuning of PID controllers", Genetic algorithms in Eng. Syst. p.141-145.

Maciejowski, J.M., 1989, "Multivariable Feedback Design", Ed. Addison Wesley, Cornwall, Great Britain, 424 p.

Mitsukura, Y., Yamamoto, T., Kaneda, M., 1997, "A genetic tuning algorithm of PID parameters", Int. Conf. Systems, Man and Cybern., vol.1, p.923-928.

Omatu, S., Deris, S., 1996, "Stabilization of inverted pendulum by the genetic algorithm", Conference on Emerging Tech. and Factory Automation, vol.1, p.282-287.

Porter, B., Jones, A. H., 1992, "Genetic tuning of digital PID controllers", Electronic Letters, vol.28, n.9, p.843-844.

Ramos, F.O., Leite Filho, W.C. and Moreira, F.J.O., 2003, "Gain computation strategy for an attitude control system", 17th Brazilian Congress of Mechanical Engineering, São Paulo, Brazil.

Takahashi, R.H.C., 2004, "Otimização Escalar e Vetorial", Notas de Aula, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil.