ICASE

SUBOPT - A CAD PROGRAM FOR SUBOPTIMAL LINEAR REGULATORS

Peter J. Fleming

Contract No.  NASI-17070
February 1985

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association
SUBOPT - A CAD PROGRAM FOR SUBOPTIMAL LINEAR REGULATORS

Peter J. Fleming
University College of North Wales

Abstract

This interactive software package provides design solutions for both standard linear quadratic regulator (LQR) and suboptimal linear regulator problems. Intended for time-invariant continuous systems the package is easily modified to include sampled-data systems. LQR designs are obtained by established techniques while the large class of suboptimal problems containing controller and/or performance index options is solved using a robust gradient minimization technique. Numerical examples demonstrate features of the package and recent developments are described.


Research was supported in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-17070 while the author was in residence at the Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23665.
Introduction

In addition to providing CAD solutions to the linear quadratic regulator (LQR) problem for time-invariant continuous systems this package offers the designer a range of attractive "suboptimal" alternatives. These alternatives were first suggested in the early seventies to offset certain disadvantages of the LQR approach which include its reliance on full state feedback or state reconstruction techniques and its sensitivity to plant specifications. The so-called "suboptimal" designs optimize problems which are related to the LQR formulation but which are not bound by its restrictive framework. In addition to handling the basic LQR design SUBOPT accommodates options which permit both alternative controller configurations and performance index (PI) descriptions.

The design options made available are:

Controller Options

* Full state feedback
* Output feedback
* Dynamic compensation

PI Options

* Sensitivity reduction
* Virtual model-following
* Implicit model following

One controller option is selected by the designer and may be used simply with the conventional PI containing quadratic terms in state and control or an extended PI which includes sensitivity reduction and/or model-following.
terms. With this more flexible design approach it is easy to compare different design philosophies within the package. A further feature, the "gain-fixing," permits the designer to fix one or more controller parameters to have zero or non-zero values. This has proved useful, for example, (i) in eliminating redundant gain elements, (ii) for investigating certain failure modes and (iii) for realizing minimal parameter compensators.

Methodology

Plant parameters are input according to the system description

\[
\dot{x}_p = A_p x_p + B_p u_p, \quad y_p = C_p x_p
\]

and the desired controller configuration is selected:

(i) \( u_p = K_s x_p \) - full state feedback

(ii) \( u_p = K_0 y_p \) - output feedback

(iii) \( u_p = A_c y_p + B_c x_c \) - dynamic compensation

where \( \dot{x}_c = C_c y_p + D_c x_c, \quad x_c(0) = 0, \)

\( x_p \) is the plant state vector,

\( y_p \) is the output state vector,

\( u_p \) is the control vector and

\( x_c \) is the compensator state vector.
Elements of controller gain matrices $K_s$, $K_0$ or $A_C$, $B_C$, $C_C$ and $D_C$ are the minimizing parameters of the PI.

The conventional quadratic infinite-time PI

$$J = \int_0^\infty (x_p^T Q_p x_p + u_p^T R_p u_p) dt$$

is augmented with a quadratic compensator measure if controller configuration (iii) is selected.

It may be augmented further if any of the PI options are selected. If sensitivity reduction is an objective a quadratic measure of sensitivity is included, based on the sensitivity vector $x_s = \partial x / \partial \alpha$, where $\alpha$ is a varying system parameter. If model-following is to be achieved a quadratic measure of the difference between plant and model states for virtual model-following (VMF) or derivatives for implicit model-following (IMF) replaces the usual state measure in the PI, where the model is defined thus:

$$\dot{x}_m = A_m x_m, y_m = C_m x_m,'$$

$x_m$ is the model state vector and

$y_m$ is the model output vector.

The general suboptimal linear regulator problem (and LQR problem) is formulated as follows:

Given the system

$$\dot{x} = A x, x(0) = x_0$$
find the feedback controller, \( u_p \), of specified configuration which minimizes the PI
\[
J = \int_0^\infty x' Q x \, dt.
\]

Vector \( x \) may simply contain \( x_p \) or depending on the option(s) selected may be automatically augmented to include \( x_c, x_g \), and/or \( x_m \). Accordingly the program builds the appropriate \( A \) and \( Q \).

Option choices which lead to the basic LQR design are identified by the program and the resulting matrix Riccati equation is solved using Kleinman's iterative technique [1]. Otherwise, the optimization problem is solved using a quasi-Newton gradient minimization algorithm to minimize \( J \). Both approaches require an initial controller estimate which results in a stable system. Such an estimate is usually available; if not, one can be obtained within the program package employing the method suggested by Koenigsberg and Frederick [2].

The program realization of the problem formulation permits the specification of fixed elements within the controller structure so that not all of the controller gain elements need be minimizing parameters; this introduces additional flexibility to controller configurations.

The minimization algorithm fails only if the user-selected convergence termination criterion is not satisfied after a prespecified number of PI evaluations. Such a failure will occur

(i) if an insufficient number of PI evaluations have been made in pursuit of the solution, or

(ii) if the termination requirement demands a numerical precision for which the particular combination of program package and host machine is unsuited.
A "soft" failure mode has been adopted thus permitting

(a) immediate re-entry to the routine in the event of (i), using the most recent controller estimate, or

(b) specification of a less stringent termination criterion in the event of (ii).

High precision in the solution is rarely needed when the design goal is viewed within the control engineering context.

An attractive feature of the LQR design approach is that its solution is independent of plant initial conditions. If these are unknown the initial condition dependency of the suboptimal solution is adequately accommodated by permitting the designer to

(a) minimize $J$ for an average set of initial conditions,

(b) minimize $J$ for a worst-case set of initial conditions, or

(c) minimize $J$, given some statistical knowledge of the initial conditions.

**Software Details**

The program package contains robust, efficient numerical algorithms which have been thoroughly tested in industrial and academic environments. It is a conversational-mode package, having versions in FORTRAN-10 and FORTRAN 77 with the graphics component based on GHOST. Source code is fully commented and the software documented [3], [4]. Array sizes in code are parameterized in order that the user may tailor the program to fit design and host machine requirements. Typically, an overall system of order 30 will require 190K bytes of executable code (excluding graphics). A typical run-time on a DEC-10
machine for a tenth-order suboptimal linear regulator problem having 4 variable gain elements is 17 seconds.

Design evaluation tools include closed-loop eigenvalue determination, a COST routine to permit investigation of individual components of the PI, e.g.,

\[ \int_0^\infty x_s^T Q_s x_s \, dt \quad \text{or} \quad \int_0^\infty u_s^2 \, dt, \]

and a simulation facility for graphical display of system responses. A series of design solutions may be obtained through modification of weighting matrices, controller/PI options, etc. The evaluation tools, therefore, provide a ready comparison between designs, e.g., contrasting an LQR solution with an output feedback design.

Design Examples

Helicopter Regulation

Helicopter longitudinal dynamics are described in [5] for a plant which has four states, \( x_p^T = [u_x, u_z, \theta, \delta]^T \), and two controls, \( u_p^T = [u_1, u_2]^T \). An LQR design, incorporating full state feedback, obtained for \( Q_p = \text{diag}(0.1,0.1,0,0) \) and \( R_p = \text{diag}(0.01,0.01) \), yields a satisfactory model response:

\[ \dot{x}_m = (A_p + B_p K^*_s)x_m = A_m x_m, \]

where \( \dot{x}_m = K_s x_p \). However, the goal is to satisfactorily regulate \( u_x \) and \( u_z \) (forward and vertical velocities) employing feedback from only two states, \( \theta \) and \( \delta \) (pitch angle and pitch rate), since the use of airspeed sensors for \( u_x \) and \( u_z \) is undesirable.
Such a design may be effected by minimization of the PI:

\[
J = \int_0^\infty (x_p - x_m)^T Q_p (x_p - x_m) + u_p^T R_p u_p \, dt
\]

which includes a virtual model-following term and by specifying controller feedback from \( \theta \) and \( \dot{\theta} \) alone. The VMF term forces this new reduced feedback design to closely match the response of the original LQR design. Setting \( Q_p \) and \( R_p \) as given above yields controller \( A \):

\[
\begin{bmatrix}
0.146 & 0.074 \\
-0.279 & 0.134
\end{bmatrix}
\begin{bmatrix}
\theta \\
\dot{\theta}
\end{bmatrix}
\]

and from the resulting responses (see Figure 1) it is clear that reasonable regulation can be achieved with comparable control effort to the LQR design without resorting to use of velocity sensors.

Assume now a possible pitch rate sensor failure, in which case we wish to maintain reasonable regulation with pitch angle feedback alone. Setting the gain elements corresponding to pitch rate feedback to zero and minimizing \( J \) with respect to the two remaining control gains reveals that although there is some degradation in system performance using the resulting controller \( B \):

\[
\begin{bmatrix}
0.157 \\
-0.301
\end{bmatrix}
\begin{bmatrix}
\theta
\end{bmatrix}
\]

the quality of regulation is still acceptable. Now if we accept this as a base or "worst-case" controller an improved controller can be obtained by fixing the pitch angle gain elements at these values and minimizing \( J \) with respect to the pitch rate gain elements. The resulting "robust" controller,
Desensitizing a Gas Turbine Engine Controller

Consider now the regulation of the two spool speeds ($x_3$ and $x_4$) of a twin-spool gas turbine engine. An output feedback control design which conforms to rigid engineering constraints, has already been effected for this 2-control 5-state problem via a nonlinear optimization technique (for details and model data, see [6]). Particular attention has been paid to control magnitudes and the step responses of $x_3$ and $x_4$, which must have little or no overshoot since overspeeding of the engine rotors will lead to blade failure. In this design, however, no account could be taken of plant parameter variations.

The engine is particularly susceptible to variations in the low pressure spool speed time constant, $\tau_{LP}$ (nominal value $\tau_{LP_0}$), and its effect on $x_3$ is shown in Figure 2(a); variations have little effect on $x_4$. When $\tau_{LP} = 0.5 \tau_{LP_0}$, the low pressure spool speed, $x_3$, has an unacceptable overshoot, and for $\tau_{LP} = 1.5 \tau_{LP_0}$ the response is rather sluggish. The object, then, is to design a controller which is less sensitive to variations in $\tau_{LP}$ and which will produce a similar nominal response. This nominal response is characterized by a model matrix, $A_m$, derived from the original design.

One approach which suggests itself is to include a sensitivity term in the PI; however, experience has indicated that simple inclusion of this term
leads to design difficulties since it can prove difficult to choose suitable weighting matrices to simultaneously yield sensitivity reduction and a desirable system response. It has been found easier to replace the state term by a model-following term thus:

\[ J = \int_0^\infty \{ (\dot{x}_p - \dot{x}_m)^T Q_p (\dot{x}_p - \dot{x}_m) + x_s^T Q_s x_s + u_p^T R_p u_p \} dt. \]

IMF has been employed here because, although it is less accurate than VMF, it leads to a lower-order problem and has the added advantage that \( R_p \) may be set to zero and PI minimization will still lead to a finite controller, thereby releasing the designer from the chore of balancing \( Q_p, Q_s, \) and \( R_p \) to achieve the desired result.

Setting \( Q_p = 1000 I_5 \) and \( Q_s = \text{diag}(0,0,10^5,0,0) \) leads to the controller:

\[ u = \begin{bmatrix} 0.09 & -8.88 \\ -13.70 & 61.44 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \]

and the desensitized response for \( x_3 \) is shown in Figure 2(b). State dispersion of \( x_3 \) is considerably reduced, and the overshoot when \( \tau_{LP} = 0.5 \tau_{LP0} \) has been eliminated using a controller of comparable size to that used in the original design.

The performance has been further improved by employing a dynamic compensator (see Figure 2(c)):

\[ u_p = \begin{bmatrix} -2.28 & 8.61 \\ -8.49 & 37.28 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} -0.37 & -14.56 \\ -13.60 & 7.54 \end{bmatrix} x_c \]

where
\[
\dot{x}_c = \begin{bmatrix}
1 & -8.57 \\
1 & -5.76
\end{bmatrix}
\begin{bmatrix}
x_3 \\
x_4
\end{bmatrix}
+ \begin{bmatrix}
-3.77 & 0 \\
0 & -4.32
\end{bmatrix} x_c, \ x_c(0) = 0
\]

which also results in a controller of comparable size. IMF is unsuitable in connection with dynamic compensation, so VMF has been implemented in this case where the PI is

\[
J = \int_0^\infty \left\{ (y_p - y_m)^T Q_p (y_p - y_m) + x_s^T Q_s x_s + u_p^T R_p u_p + x_c^T R_c x_c \right\} dt.
\]

The controller was found by setting \( Q_p = \text{diag}(3000,3000) \), \( Q_s = \text{diag}(0,0,100,0,0) \), \( R_p = 100 I_2 \) and \( R_c = I_2 \). (It is necessary for \( R_c \) to be positive definite although its contribution here to the final design is negligible.) It will be noted from the controller description above that a minimal parameter canonical form has been adopted: using the gain fixing facility, elements \((1,1)\) and \((2,1)\) of \( C_c \) are set to unity and elements \((1,2)\) and \((2,1)\) of \( D_c \) are set to zero prior to minimization.

Recent Developments

Both the LQR and suboptimal regulator approaches rely on a trial-and-error approach to design in which weighting matrices are adjusted iteratively in a design loop until the final goal is realized. In a new approach [7], the linear quadratic constrained regulator (LQCR), the design is recast as a constrained optimization problem in which both design objective and constraints are quadratic cost functions. This formulation permits the separate consideration of, for example, model-following errors, sensitivity
measures and control energy as objectives to be minimized or limits to be observed. The result is a one-pass design procedure whose solution, which includes identification of active constraints at the minimum, reveals useful insights into the design problem (see [8]).

The LQCR technique is available as research code, the software having been constructed from the SUBOPT package. Present work is engaged in investigating multi-objective optimization strategies in which multiple goals are specified and design solutions sought through nonlinear programming techniques.

**Acknowledgement**

The author is indebted to Mr. A. G. Shutler (Rolls Royce) for valuable discussions in connection with gas turbine engine design.
References


Figure 1
Figure 2
This interactive software package provides design solutions for both standard linear quadratic regulator (LQR) and suboptimal linear regulator problems. Intended for time-invariant continuous systems the package is easily modified to include sampled-data systems. LQR designs are obtained by established techniques while the large class of suboptimal problems containing controller and/or performance index options is solved using a robust gradient minimization technique. Numerical examples demonstrate features of the package and recent developments are described.