NASA/TM—2016–216632

Use of the NLPQLP Sequential Quadratic Programming Algorithm to Solve Rotorcraft Aeromechanical Constrained Optimisation Problems

Jane Anne Leyland
Ames Research Center
Moffett Field, California

April 2016

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM—2016–216632

Use of the NLPQLP Sequential Quadratic Programming Algorithm to Solve Rotorcraft Aeromechanical Constrained Optimisation Problems

Jane Anne Leyland
Ames Research Center
Moffett Field, California

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, CA 94035-1000

April 2016

Available from:

NASA STI Support Services
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
757-864-9658

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312
webmail@ntis.gov
703-605-6000

This report is also available in electronic form at

http://ntrs.nasa.gov

Table of Contents

Table of Contents (cont.)

Appendix A: NLPQLP: A Fortran Implementation of a Sequential Quadratic Programming Algorithm with Distributed and Non-Monotone Line Search – User's Guide, Version 3.1

Separate Volumes:

Nomenclature

C_1 Input coefficient for the first term in the equation that defines $T_{q,p}$.

C_2 Input coefficient for the second term in the equation that defines $T_{q,p}$.

C_3 Input coefficient for the first term in the equation that defines $\theta_{p_{Initial}}$.

C_4 Input coefficient for the second term in the equation that defines $\theta_{p_{Initial}}$.

C_5 Input coefficient for the first term in the equation that defines ΔZ_{A_q}.

C_6 Input coefficient for the second term in the equation that defines ΔZ_{A_q}.

$g[Z(\theta)]$ Scalar performance index function that defines the performance index J. $g[Z(\theta)]$ is a function of the plant output measurement $Z-$vector, which is a function of the control $\theta-$vector. In general, these functions can be non-linear.

I_Z Set of all $q \ni Z_q \in Z$.

I_θ Set of all $p \ni \theta_p \in \theta$.

IMSL International Mathematics and Statistics Library, Inc.—a commercial collection of software libraries of numerical analysis functionality that are implemented in C, Java, C#.NET, and Fortran computer programming languages.

$\mathrm{ISEED1}_k$ Input seed argument for the $k-th$ call to the random number generator function RAN$(\bullet)$ in the first term in the equation that defines $T_{q,p}$, and updated automatically on completion of the generation of the random number. This argument should initially be set to a large odd-integer value.

$\mathrm{ISEED2}_k$ Input seed argument for the $k-th$ call to the random number generator function RAN$(\bullet)$ in the first term in the equation that defines $\theta_{p_{Initial}}$, and updated automatically on completion of the generation of the random number. This argument should initially be set to a large odd-integer value.

Nomenclature (cont.)

ISEED3_k — Input seed argument for the $k-th$ call to the random number generator function RAN$(\bullet)$ in the first term in the equation that defines ΔZ_{A_q}, and updated automatically on completion of the generation of the random number. This argument should initially be set to a large odd-integer value.

J — Scalar performance index that is defined by $g[Z(\theta)]$. In general, this function can be non-linear. For the problems considered in this research, J is a scalar performance index that is a quadratic function of the plant output measurement vector (i.e., the Z – vector). In this case, Z is a linear function of the control $\theta-$vector, and J is a quadratic function of the control $\theta-$vector.

JSEED1_l — Input seed argument for the $l-th$ call to the random number generator function RAN$(\bullet)$ in the second term in the equation that defines $T_{q,p}$, and updated automatically on completion of the generation of the random number. This argument should initially be set to a large odd-integer value.

JSEED2_l — Input seed argument for the $l-th$ call to the random number generator function RAN$(\bullet)$ in the second term in the equation that defines $\theta_{p_{Initial}}$, and updated automatically on completion of the generation of the random number. This argument should initially be set to a large odd-integer value.

JSEED3_l — Input seed argument for the $l-th$ call to the random number generator function RAN$(\bullet)$ in the second term in the equation that defines ΔZ_{A_q}, and updated automatically on completion of the generation of the random number. This argument should initially be set to a large odd-integer value.

k — Index number for the input seed argument for calls to the random number generator function RAN$(\bullet)$ in the first term in the equations that defines $T_{q,p}$, $\theta_{p_{Initial}}$, and ΔZ_{A_q}, where: $k \in [1, 2, 3, \bullet, \bullet, \bullet, \bullet, \rightarrow +\infty)$.

l — Index number for the input seed argument for calls to the random number generator function RAN$(\bullet)$ in the second term in the equations that defines $T_{q,p}$, $\theta_{p_{Initial}}$, and ΔZ_{A_q}, where: $l \in [1, 2, 3, \bullet, \bullet, \bullet, \bullet, \rightarrow +\infty)$.

N_{EQ} — Number of elements (dimension) in the Equality Constraint $\phi(\theta)$–vector.

Nomenclature (cont.)

N_{IEQ}	Number of elements (dimension) in the Inequality Constraint $\psi(\theta)$ – vector. $N_{IEQ} = N_{IEQ_1} + N_{IEQ_2}$.
N_{IEQ_1}	Number of elements in the First Inequality Constraint ${}^1\psi(\theta)$ – sub-vector.
N_{IEQ_2}	Number of elements in the Second Inequality Constraint ${}^2\psi(\theta)$ – sub-vector.
N_Z	Number of elements (dimension) in the predicted measurement Z – vector.
N_θ	Number of elements (dimension) in the control θ – vector.
NLP	Non-Linear Programming algorithm.
NLPQLP	Non-Linear Programming (NLP) algorithm that employs the Sequential Quadratic Programming (SQP) algorithm as its core algorithm.
p	Index number for the control θ – vector elements.
q	Index number for the predicted measurement Z – vector elements.
r	Index number for the Equality Constraint $\phi(\theta)$ – vector elements.
RAN(●)	Uniform Random Number Distribution Function that yields a uniformly random real number $\in [0, 1)$.
s	Index number for the Inequality Constraint $\psi(\theta)$ – vector elements.
SQP	Sequential Quadratic Programming (SQP) algorithm.
T	System or transfer $(N_Z \times N_\theta)$ matrix either defined by direct input or synthetically determined.
$T_{q,p}$	The $(q, p) - th$ element of the system or transfer $(N_Z \times N_\theta)$ matrix.
W_Z	Diagonal $(N_Z \times N_Z)$ weighting matrix in the performance index. The default setting that is the identity matrix; can be redefined by input.

Nomenclature (cont.)

W_{θ}	Diagonal $\left(N_{\theta} \text{ x } N_{\theta}\right)$ weighting matrix in the θ term of the regulator performance index J. The default setting that is the null matrix; can be redefined by input.
$W_{\dot{\theta}}$	Diagonal $\left(N_{\theta} \text{ x } N_{\theta}\right)$ weighting matrix in the $\dot{\theta}$ term of the regulator performance index J. The default setting that is the null matrix; can be redefined by input.
Z	Equals $Z(\theta)$ and is used when the omission of the explicit dependence on the control $\theta-$vector does not present any confusion, ambiguity, or vagueness.
$Z(\theta)$	Predicted measurement $Z-$vector $\left(N_Z \text{ x } 1\right)$ evaluated during the optimisation or regulator process and is a function of the control θ – vector. In general, this function can be non-linear. For the problems analysed in this research, $Z(\theta)$ is a linear function of the control $\theta-$vector.
Z_A	Actual measurement $Z-$vector $\left(N_Z \text{ x } 1\right)$ that would normally be evaluated during the previous duty cycle or at a reference epoch time. Z_A is either directly input or synthetically determined.
Z_q	The $q-th$ element of the predicted measurement $Z-$vector.
ΔZ_A	Random component of the synthetically determined actual measurement Z_A-vector.
ΔZ_{A_q}	Random component of the $q-th$ element of the synthetically determined actual measurement Z_A-vector.
ε	Input small value constant selected to prevent $\theta_{0_{Initial}}$ from being identical on a bound (i.e., the least upper bound [l.u.b.] or the greatest lower bound [g.l.b.]).
θ	Control $\theta-$vector $\left(N_{\theta} \text{ x } 1\right)$.
$\dot{\theta}$	Time rate of change of the control $\theta-$vector $\left(N_{\theta} \text{ x } 1\right)$.
θ_0	Solution θ_{Sol} – vector $\left(N_{\theta} \text{ x } 1\right)$ that would normally be evaluated during the previous duty cycle or at a reference epoch time. θ_0 is either directly input or synthetically determined.

Nomenclature (cont.)

θ_{MAX_p} Least upper bound (l.u.b.) for the $p-th$ element of the control θ – vector $\left(\text{N}_\theta \text{ x } 1\right)$.

θ_{MIN_p} Greatest lower bound (g.l.b.) for the $p-th$ element of the control θ – vector $\left(\text{N}_\theta \text{ x } 1\right)$.

θ_p The $p-th$ element of the control $\theta-$vector $\left(\text{N}_\theta \text{ x } 1\right)$.

$\dot{\theta}_p$ The $p-th$ element of the time rate of change of the control $\theta-$vector $\left(\text{N}_\theta \text{ x } 1\right)$.

$\theta_{p_{Initial}}$ The $p-th$ element of the solution θ_{Sol} – vector $\left(\text{N}_\theta \text{ x } 1\right)$ that would normally be evaluated during the previous duty cycle or at a reference epoch time. θ_0 is synthetically determined.

θ_{Sol} Solution θ_{Sol} – vector $\left(\text{N}_\theta \text{ x } 1\right)$ evaluated during the optimisation or regulator process.

θ_{Sol_p} The $p-th$ element of the solution control θ_{Sol} – vector $\left(\text{N}_\theta \text{ x } 1\right)$ evaluated during the optimisation or regulator process.

$\phi(\theta)$ Equality Constraint $\phi(\theta)-$vector $\left(\text{N}_{\text{EQ}} \text{ x } 1\right)$ function; in general, can be dependent on the $\theta-$vector and be non-linear.

$\phi_r(\theta)$ The $r-th$ element of the Equality Constraint $\phi(\theta)-$vector $\left(\text{N}_{\text{EQ}} \text{ x } 1\right)$ function.

$\psi(\theta)$ Complete Inequality Constraint $\psi(\theta)-$vector $\left(\left[\text{N}_{\text{IEQ}} = \text{N}_{\text{IEQ}_1} + \text{N}_{\text{IEQ}_2}\right] \text{ x } 1\right)$ function; in general, can be dependent on the $\theta-$vector and be non-linear. $\psi(\theta)$ is comprised of the two sub-vector functions ${}^1\psi(\theta)$ and ${}^2\psi(\theta)$. Specifically:

$$\psi(\theta) = \begin{bmatrix} {}^1\psi(\theta) \\ {}^2\psi(\theta) \end{bmatrix}$$

Nomenclature (concluded)

${}^{1}\psi(\theta)$	First Inequality Constraint ${}^{1}\psi(\theta)-$sub-vector $\left(N_{IEQ_1} \text{ x } 1\right)$ function with elements of the First Inequality Constraint Form.
${}^{2}\psi(\theta)$	Second Inequality Constraint ${}^{2}\psi(\theta)-$sub-vector $\left(N_{IEQ_2} \text{ x } 1\right)$ function with elements of the Second Inequality Constraint Form.
$\psi_s(\theta)$	The $s-th$ element of the complete Inequality Constraint vector $\left(N_{IEQ} \text{ x } 1\right)$ $\psi(\theta)-$function.

Summary

Optimisation of a control vector, an aerodynamic surface design, or an aircraft configuration potentially offers significant performance enhancement to rotorcraft systems. These problems typically include various types of constraints. Previous research and analysis indicated that non-linear programming methods that solve a sequence of related quadratic-programming sub-problems could be used successfully to solve these problems. Accordingly, a licence for one of the latest versions of Professor Klaus Schittkowski's very successful Sequential Quadratic Programming NLPQLP software was obtained and used to experiment with and analyse typical optimisation problems encountered in various rotorcraft wind tunnel and flight tests. Emphasis was directed toward obtaining efficiency, robustness, and speed in computation.

The NLPQLP software was installed on both a mainframe computer and a desktop computer. Stand-alone main driver codes were developed to task the NLPQLP software to solve non-linear programming (NLP) problems. The required input data was synthesised to facilitate this analysis. Solution to the classic regulator problem was included to provide verification of the NLP solutions to the unconstrained optimisation problems. The mainframe computer was used to develop the driver codes and to experiment with the various models and tune the associated input. The desktop computer was used to refine the process and to develop software for laptop computers that can be used in austere test environments including wind tunnels.

The problems solved with this analysis were of the type encountered in rotorcraft applications where there is a linear dependence (i.e., a T-Matrix plant model) of the measurement vector on the control vector. These problems ranged from a relatively simple, unconstrained 4-vector control to a relatively large, constrained 60-vector control problem. Five different control vector dimensions ranging from 4 to 60 were analysed, and solutions were obtained for each of these with no imposition of constraints, imposition of only equality constraints, imposition of only inequality constraints, and imposition of both equality and inequality constraints. The smaller 4-, 6-, and 8-dimension control vector problems are representative of actual rotorcraft control problems. The solutions to these problems were sufficiently fast to be included in real-time duty cycles. The larger 30- and 60-dimension control vector problems are representative of aerodynamic surface design or aircraft configuration problems and, although they were solved rapidly, they are more suitable to non-real-time design applications. Solutions were obtained for all problems considered, and verification of the solutions to all of the unconstrained problems was obtained by solving the regulator problem. Although tuning and some input adjustments were required to successfully solve the large constrained 60-vector control problems, the NLPQLP System proved to be an efficient and reliable method to solve these problems.

1.0 Introduction

Previous research on the feasibility and desirability of using a constrained optimisation technique to define the optimal control vector and plant model for rotorcraft was accomplished on a mainframe computer not part of actual wind tunnel and/or flight-test experiments. This research indicated that constrained optimisation methodology provides better controller performance in many cases compared to results obtained with the widely used solution to the regulator problem for this application (ref. 1), and would be useful in defining the optimal configuration and required constants for a non-linear neural-network plant model (refs. 2–5). The initial research (ref. 1) on the development, design, and use of optimal controllers, both open- and closed-loop, to optimise rotorcraft aeromechanical behaviour indicated that the general non-linear programming method coded as the NCONF/DNCONF subroutine system, which is available in the IMSL MATH/LIBRARY (ref. 6), worked very well for solving the required constrained optimisation problems and was significantly superior to methods previously used for solving problems of this type. This IMSL NCONF/DNCONF subroutine system, which dates back to 1989, was used successfully on subsequent research studies (refs. 2–5) and likewise worked very well for solving the required constrained optimisation problems.

This IMSL NCONF/DNCONF subroutine system is based on the work by Schittkowski, Gill et al., Powell, and Stoer (refs. 7–13). This method solves the general non-linear programming problem by solving a sequence of related quadratic programming sub-problems. One advantage of this method is that quadratic programming problems can be solved efficiently. A very important property of quadratic programming problems is that if the quadratic coefficient matrix in the performance index is positive definite, the problem has a unique solution, which is, of course, the global solution. This means that the sequence of solutions to the quadratic programming sub-problems will converge to the global solution of the general problem in the limit providing that the quadratic coefficient matrix in the performance index remains positive definite in the process.

To conduct a wind tunnel experiment, the optimisation code is installed on a computer within the wind tunnel or on a portable computer such as a laptop that could be brought into the wind tunnel. Research on available suitable optimisation codes revealed that Professor Klaus Schittkowski had revised and updated his sequential quadratic programming method, which is part of the IMSL MATH/LIBRARY, via several versions of the code that improved performance and eliminated errors, and that it was available by licence as a stand-alone code (i.e., not part of a library). NASA obtained a licence, and Version 3.1, dated February 2010, was installed on both the Mac Pro desktop computer and the Hewlett-Packard Alpha mainframe computer. The code that was installed on the Mac Pro desktop computer was transportable to a Mac laptop computer for use in the wind tunnel. Version 3.1 of the NLPQLP System (ref. 14 and Appendix A) was used for the study described herein.

The linear global plant model, which linearly relates the measurement vector to the control vector and is widely used for rotorcraft aeromechanical behaviour studies, was assumed for the problems that were solved as part of this research. For this research, two options could be used to define the T-Matrix: (1) use actual test data to define these elements, or (2) synthetically define these elements using a uniformly distributed random function. The second option was used for the results presented herein.

This method was computationally implemented by writing stand-alone main driver codes assuming the linear global plant model. These drivers included the widely used solution to the regulator problem, used for comparison purposes. These codes were written to solve several typical problems of rotorcraft aeromechanical behaviour and are specific to the dimension of the T-Matrix (i.e., the dimensions of the measurement vector and the control vector) and the computer (i.e., the Hewlett-Packard Alpha mainframe or the Mac Pro desktop) that the problem was run on.

The very general non-linear programming problem that is solved by the NLPQLP System (ref. 14) is defined in section 2.1. The specific problems that were solved using the NLPQLP System for this study are defined in section 2.2. A description of the synthetic data generation process is presented in section 2.3, and the definition of, and solution to, the regulator problem are described in section 2.4.

The (6 x 4), (6 x 6), and (24 x 8) T-Matrix non-linear programming (NLP) control problems are representative of actual rotorcraft control problems. The solutions to these problems were sufficiently fast to be included in real-time duty cycles. The (90 x 30) and (90 x 60) T-Matrix NLP problems are representative of aerodynamic surface design and/or aircraft configuration problems and, although they were solved rapidly, they are more suitable to non-real-time design applications.

The results are shown in Appendix B and Appendix C, separate volumes of this report.

Listings of the command (DCL) file code and the Fortran main driver code for the Hewlett-Packard Alpha mainframe computer, and the input and output for the four sub-problems that were part of the (6 x 4), (6 x 6), (24 x 8), (90 x 30), and (90 x 60) T-Matrix NLP control problems solved using the Hewlett-Packard Alpha mainframe computer, are presented in Appendix B. A listing of the Fortran main driver code for the Mac Pro desktop computer, and the input and output for the four sub-problems that were part of the (6 x 4), (6 x 6), (24 x 8), (90 x 30), and (90 x 60) T-Matrix NLP control problems solved using the Mac Pro desktop computer, are presented in Appendix C.

2.0 Technical

This study documents the testing, experimentation, and evaluation of Version 3.1 of the NLPQLP System (ref. 14 and Appendix A) to ascertain its suitability to solve rotorcraft optimisation problems. This NLPQLP System is designed to solve a very general NLP problem by solving a sequence of related quadratic programming sub-problems. One advantage of this method is that quadratic programming problems can be solved efficiently. A very important property of quadratic programming problems is that if the quadratic coefficient matrix in the performance index is positive definite, then the problem has a unique solution, which is, of course, the global solution. This means that the sequence of solutions to the quadratic programming sub-problems will converge to the global solution of the general problem in the limit providing that the quadratic coefficient matrix in the performance index remains positive definite in the process.

This very general NLP problem is described in section 2.1. The problems solved in this analysis are encountered in various rotorcraft applications where there is a linear dependence (i.e., a T-Matrix plant model) of the measurement vector (the measurement Z – vector) on the control vector (i.e., the control θ – vector). The general form of the T-Matrix NLP problems considered in this study is referred to herein as the General T-Matrix NLP control problem, and is defined in section 2.2. Specific (6 x 4) T-Matrix NLP control problems (see section 2.2.1), (6 x 6) T-Matrix NLP control problems (see section 2.2.2), (24 x 8) T-Matrix NLP control problems (see section 2.2.3), (90 x 30) T-Matrix NLP control problems (see section 2.2.4), and (90 x 60) T-Matrix NLP control problems (see section 2.2.5) were analysed and solved. Each of these problems had four sub-problems: (1) unconstrained optimisation, (2) optimisation with only equality constraints, (3) optimisation with only inequality constraints, and (4) optimisation with both equality constraints and inequality constraints.

In an actual wind tunnel or flight test experiment or optimisation application, the required input would be the actual test data, which perhaps might be reformatted for computer compatibly. To expedite the testing, experimentation, and evaluation of the NLPQLP System, the input data was synthetically determined. This synthesis process is described in section 2.3 and its sub-sections.

Finally, the classic regulator problem is solved as an appendix to the unconstrained optimisation NLP problems to provide a means to verify their solutions. The classic regulator problem is defined in section 2.4.

In this study the NLPQLP System was coded in Fortran 77 and installed on both a Hewlett-Packard Alpha Server GS1280 mainframe computer with the Open VMS Version 8.2 Operating System and a Mac Pro desktop computer with the Mac OS X, Version 10.5.8 Operating System. The VMS FORTRAN Compiler was used to compile the code on the mainframe computer, and the G95 FORTRAN Compiler was used to compile the code on the Mac Pro desktop computer. It is noted that these codes will, in general, produce different numerical results for these computer systems when synthetic input data is generated because the random number generator function algorithms used for data synthesis are different for these computer systems. The codes for both the Hewlett-Packard Alpha mainframe computer and the Mac Pro desktop computer are stand-alone codes and do not require any special software libraries. Accordingly, the associated software and codes installed on the Mac Pro desktop computer should be transportable to a Mac laptop computer for use in the wind tunnel.

2.1 General Non-Linear Programming Problem

The general optimisation problem that can be solved with the NLPQLP System is:

Determine the θ_{Sol} *– vector that solves the problem:*

$$\underset{\theta_p \in \theta}{Minimise} \quad J = g\left[Z(\theta)\right] \qquad for \quad p \in I_\theta$$

$$where \qquad Z = Z(\theta) = \left[\bullet \;\bullet\; \bullet \;\; \left\{Z_q \,\middle|\, q \in I_{\mathrm{Z}}\right\} \;\; \bullet \;\bullet\; \bullet\right]^{\mathrm{T}}$$

$$and \qquad \theta = \left[\bullet \;\bullet\; \bullet \;\; \left\{\theta_p \,\middle|\, p \in I_\theta\right\} \;\; \bullet \;\bullet\; \bullet\right]^{\mathrm{T}}$$

$$I_{\mathrm{Z}} = \left\{\bullet \;\bullet\; \bullet \;\; \left\{\forall \;\, q \;\ni\; Z_q \;\in\; Z\right\} \;\; \bullet \;\bullet\; \bullet\right\}$$

$$I_\theta = \left\{\bullet \;\bullet\; \bullet \;\; \left\{\forall \;\, p \;\ni\; \theta_p \;\in\; \theta\right\} \;\; \bullet \;\bullet\; \bullet\right\}$$

Subject to:

$$\left.\begin{array}{l} \theta_{\mathrm{MIN}_p} \;\le\; \theta_p \;\le\; \theta_{\mathrm{MAX}_p} \\ \theta_{\mathrm{MIN}_p} \in \left(-\infty \,,\, +\infty\right) \\ \theta_{\mathrm{MAX}_p} \in \left(-\infty \,,\, +\infty\right) \end{array}\right\} \left\{\begin{array}{l} \text{Direct Constraints on the Control } \theta - \text{vector} \\ \text{Elements } \theta_p \text{ for: } \quad p \in I_\theta \end{array}\right.$$

$$\phi(\theta) \;=\; 0 \qquad \Big\} \left\{\begin{array}{l} \text{General Equality Constraint Vector Function} \\ \text{with Dimension } \mathrm{N_{EQ}} \end{array}\right.$$

$$\psi(\theta) \;\ge\; 0 \qquad \Big\} \left\{\begin{array}{l} \text{General Inequality Constraint Vector Function} \\ \text{with Dimension } \mathrm{N_{IEQ}} \end{array}\right.$$

2.2 Specific Problems Solved as Part of This Research

The specific problems that were solved during this study have a general form (i.e., the General T-Matrix NLP control problem) that is a specific application of the general non-linear programming problem defined in section 2.1. In this case, the general $g\left[Z(\theta)\right]$ scalar performance index function is replaced with a quadratic function of a T-Matrix plant model. The T-Matrix itself is a linear plant model that relates the measurement Z – vector to the control $\theta -$ vector. The performance index, J, is a quadratic function of the measurement Z – vector. Limits as per the general non-linear programming problem are imposed on the elements of the control θ – vector. Both the equality $\phi(\theta)$ and inequality $\psi(\theta)$ constraint functions are non-linear functions of the elements of the control θ – vector. The elements of the equality $\phi(\theta)$ –vector constraint function are pseudo-generalisations of the elements of a vector cross product. The Inequality Constraint $\psi(\theta)$ –vector function includes two forms of inequality constraints: (1) the First Inequality Constraint ${}^{1}\psi(\theta)$ –vector sub-vector function is comprised of elements that are pseudo-generalisations of an amplitude constraint on a matched harmonic pair (i.e., the sine and cosine elements of a specific harmonic control signal) of control elements, and (2) the Second Inequality Constraint ${}^{2}\psi(\theta)$ –vector sub-vector function is comprised of elements that are pseudo-generalisations of a rate-of-change constraint on an element of the control vector. Each of these problems has four sub-problems that were analysed and solved. These sub-problems are: (1) unconstrained optimisation, (2) optimisation with only equality constraints, (3) optimisation with only inequality constraints, and (4) optimisation with both equality constraints and inequality constraints. This General T-Matrix NLP control problem with both equality and inequality constraints is defined as:

Determine the θ_{Sol} – vector that solves the problem:

$$\underset{\theta_p \in \theta}{Minimise} \quad J = Z^{\mathrm{T}} W_Z Z \qquad for \quad p \in I_\theta$$

where

$$Z = Z(\theta) = Z_{\mathrm{A}} + \mathrm{T}\left(\theta - \theta_0\right)$$

$$Z = Z(\theta) = \left[\bullet \;\; \bullet \;\; \bullet \;\; \left\{Z_q \,\middle|\, q \in I_{\mathrm{Z}}\right\} \;\; \bullet \;\; \bullet \;\; \bullet\right]^{\mathrm{T}}$$

and

$$\theta = \left[\bullet \;\; \bullet \;\; \bullet \;\; \left\{\theta_p \,\middle|\, p \in I_\theta\right\} \;\; \bullet \;\; \bullet \;\; \bullet\right]^{\mathrm{T}}$$

$$I_{\mathrm{Z}} = \left\{\bullet \;\; \bullet \;\; \bullet \;\; \left\{\forall \;\; q \;\; \ni \;\; Z_q \;\in\; Z\right\} \;\; \bullet \;\; \bullet \;\; \bullet\right\}$$

$$I_\theta = \left\{\bullet \;\; \bullet \;\; \bullet \;\; \left\{\forall \;\; p \;\; \ni \;\; \theta_p \;\in\; \theta\right\} \;\; \bullet \;\; \bullet \;\; \bullet\right\}$$

Subject to:

$$\theta_{\text{MIN}p} \le \theta_p \le \theta_{\text{MAX}p}$$
$$\theta_{\text{MIN}p} \in (-\infty, +\infty)$$
$$\theta_{\text{MAX}p} \in (-\infty, +\infty)$$

$\left\{\begin{array}{l}\text{Direct Constraints on the Control } \theta - \text{vector} \\ \text{Elements } \theta_p \text{ for: } \quad p \in I_\theta\end{array}\right.$

$$\phi_r(\theta) = \theta_i\theta_l - \theta_j\theta_k = 0$$

$\left\{\begin{array}{l}\text{Form of the Equality Constraint } \phi(\theta) - \text{vector} \\ \text{Elements } \phi_r(\theta) \text{ for: } \; r \in [1, \text{N}_{\text{EQ}}] \text{ and} \\ i, j, k, l \in I_\theta \text{ and } i < j < k < l\end{array}\right.$

$$\psi_s(\theta) = \psi_{s_{\text{MAX}}} - \sqrt{\theta_i^2 + \theta_j^2} \ge 0$$

$\left\{\begin{array}{l}\text{Form of the First Inequality Constraint} \\ {}^1\psi(\theta) - \text{sub-vector Elements } \psi_s(\theta) \text{ for:} \\ s \in [1, \text{N}_{\text{IEQ}_1}], \; i, j \in I_\theta \text{ and } i < j\end{array}\right.$

$$\psi_s(\theta) = \psi_{s_{\text{MAX}}} - \text{Abs}(\theta_i - \theta_{0i}) \ge 0$$

$\left\{\begin{array}{l}\text{Form of the Second Inequality Constraint} \\ {}^2\psi(\theta) - \text{sub-vector Elements } \psi_s(\theta) \text{ for:} \\ s \in [(\text{N}_{\text{IEQ}_1} + 1), (\text{N}_{\text{IEQ}_1} + \text{N}_{\text{IEQ}_2})] \\ \text{and } \; i \in I_\theta\end{array}\right.$

2.2.1 (6 x 4) T-Matrix NLP Control Problems

For the (6 x 4) T-Matrix NLP control problem, the number of elements (dimension) in the control θ – vector is four (i.e., N_θ = 4), and the number of elements (dimension) in the predicted measurement Z – vector is six (i.e., N_Z = 6). Correspondingly, the system or transfer T-Matrix is (6 x 4). The number of elements (dimension) in the Equality Constraint $\phi(\theta)$ –vector is one (i.e., N_{EQ} = 1), the number of elements in the First Inequality Constraint ${}^1\psi(\theta)$ –sub-vector function is two (i.e., N_{IEQ_1} = 2), the number of elements in the Second Inequality Constraint ${}^2\psi(\theta)$ –sub-vector function is four (i.e., N_{IEQ_2} = 4), and the dimension of the Inequality Constraint $\psi(\theta)$ –vector is six (i.e., $\text{N}_{\text{IEQ}} = \text{N}_{\text{IEQ}_1} + \text{N}_{\text{IEQ}_2}$ = 6). The (6 x 4) T-Matrix NLP control problem with all constraints is defined as:

Determine the θ – vector, θ_{Sol} , that solves the problem:

$$\underset{\theta_p \in \theta}{Minimise} \quad J = Z^{\mathsf{T}} W_Z Z \qquad for \quad p \in I_\theta$$

where $$Z = Z(\theta) = Z_{\mathsf{A}} + \mathsf{T}\left(\theta - \theta_0\right)$$

$$Z = Z(\theta) = \left[\bullet \ \bullet \ \bullet \ \left\{Z_q \,\middle|\, q \in I_{\mathsf{Z}}\right\} \ \bullet \ \bullet \ \bullet\right]^{\mathsf{T}}$$

and $$\theta = \left[\bullet \ \bullet \ \bullet \ \left\{\theta_p \,\middle|\, p \in I_\theta\right\} \ \bullet \ \bullet \ \bullet\right]^{\mathsf{T}}$$

$$I_{\mathsf{Z}} = \{1,\ 2,\ 3,\ 4,\ 5,\ 6\} \quad \text{where} \quad \mathsf{N}_{\mathsf{Z}} = 6$$

$$I_\theta = \{1,\ 2,\ 3,\ 4\} \quad \text{where} \quad \mathsf{N}_\theta = 4$$

then $$Z = Z(\theta) = \left[Z_1,\ Z_2,\ Z_3,\ Z_4,\ Z_5,\ Z_6\right]^{\mathsf{T}}$$

$$\theta = \left[\theta_1,\ \theta_2,\ \theta_3,\ \theta_4\right]^{\mathsf{T}}$$

Subject to:

$$\left.\begin{array}{l} \theta_{\mathsf{MIN}_p} \le \theta_p \le \theta_{\mathsf{MAX}_p} \\ \theta_{\mathsf{MIN}_p} \in (-\infty, +\infty) \\ \theta_{\mathsf{MAX}_p} \in (-\infty, +\infty) \end{array}\right\} \begin{cases} \text{Direct Constraints on the Control} \\ \theta - \text{vector Elements } \theta_p \text{ for: } \quad p \in I_\theta \end{cases}$$

$$\left[\phi_1(\theta) = \theta_1\theta_4 - \theta_2\theta_3 = 0\right] \Big\} \begin{cases} \text{Equality Constraint } \phi(\theta) - \text{vector for:} \\ \qquad \mathsf{N}_{\mathsf{EQ}} = 1 \end{cases}$$

$$\left[\begin{array}{l} \psi_1(\theta) = \psi_{1_{\mathsf{MAX}}} - \sqrt{\theta_1^2 + \theta_2^2} \ge 0 \\ \psi_2(\theta) = \psi_{2_{\mathsf{MAX}}} - \sqrt{\theta_3^2 + \theta_4^2} \ge 0 \end{array}\right] \Bigg\} \begin{cases} \text{First Inequality Constraint} \\ {}^1\psi(\theta) - \text{sub-vector for: } \quad \mathsf{N}_{\mathsf{IEQ}_1} = 2 \end{cases}$$

$$\left.\begin{bmatrix} \psi_3(\theta) & = & \psi_{3_{\text{MAX}}} & - & \left|\theta_1 - \theta_{0_1}\right| & \geq & 0 \\ \psi_4(\theta) & = & \psi_{4_{\text{MAX}}} & - & \left|\theta_2 - \theta_{0_2}\right| & \geq & 0 \\ \psi_5(\theta) & = & \psi_{5_{\text{MAX}}} & - & \left|\theta_3 - \theta_{0_3}\right| & \geq & 0 \\ \psi_6(\theta) & = & \psi_{6_{\text{MAX}}} & - & \left|\theta_4 - \theta_{0_4}\right| & \geq & 0 \end{bmatrix}\right\} \begin{cases} \text{Second Inequality Constraint} \\ {}^{2}\psi(\theta) - \text{sub-vector for:} \quad N_{IEQ_2} = 4 \end{cases}$$

The (6 x 4) T-Matrix NLP control problem has four sub-problems: (1) unconstrained optimisation, (2) optimisation with only equality constraints, (3) optimisation with only inequality constraints, and (4) optimisation with both equality constraints and inequality constraints. It was assumed that the tasking of these T-Matrix NLP control problems occurred within the framework of real-time controller duty cycles. To expedite the testing, experimentation, and evaluation of the NLPQLP System, the required input data was synthetically determined using a process designed expressly for this analysis. Specifically, a previously identified T-Matrix and an actual control θ_0 – vector/actual measurement Z_A – vector pair from a previous duty cycle were synthesised for use as input to these problems. Additionally, these already synthesised values were directly input to the (6 x 4) T-Matrix NLP control problems for comparison purposes. The classic regulator problem was solved to verify the NLP solutions to the unconstrained optimisation NLP problems. Agreement was obtained in all cases.

Listings of the command (DCL) file code and the Fortran main driver code for the Hewlett-Packard Alpha mainframe computer, and the input and output for the four sub-problems that were part of the (6 x 4) T-Matrix NLP control problems solved using the Hewlett-Packard Alpha mainframe computer, are presented in Appendix B, section B.1. A listing of the Fortran main driver code for the Mac Pro desktop computer, and the input and output for the four sub-problems that were part of the (6 x 4) T-Matrix NLP control problems solved using the Mac Pro desktop computer, are presented in Appendix C, section C.1.

These (6 x 4) T-Matrix NLP control problems are representative of actual rotorcraft control problems. The solutions to these problems were sufficiently fast to be included in real-time duty cycles.

2.2.2 (6 x 6) T-Matrix NLP Control Problems

For the (6 x 6) T-Matrix NLP control problem, the number of elements (dimension) in the control θ – vector is six (i.e., N_{θ} = 6), and the number of elements (dimension) in the predicted measurement Z – vector is also six (i.e., N_{Z} = 6). Correspondingly, the system or transfer T-Matrix is (6 x 6). The number of elements (dimension) in the Equality Constraint $\phi(\theta)$ –vector is one (i.e., N_{EQ} = 1), the number of elements in the First Inequality Constraint ${}^{1}\psi(\theta)$ –sub-vector is three (i.e., N_{IEQ_1} = 3), the number of elements in the Second Inequality Constraint ${}^{2}\psi(\theta)$ –sub-vector function is six (i.e., N_{IEQ_2} = 6), and the dimension of the Inequality Constraint $\psi(\theta)$ –vector is nine (i.e., $N_{IEQ} = N_{IEQ_1} + N_{IEQ_2} = 9$). The (6 x 6) T-Matrix NLP control problem with all constraints is defined as:

Determine the θ – vector, θ_{Sol}, that solves the problem:

$$\underset{\theta_p \in \theta}{Minimise} \quad J = Z^{\mathrm{T}} W_Z \, Z \qquad for \quad p \in I_{\theta}$$

where

$$Z = Z(\theta) = Z_{\mathrm{A}} + \mathrm{T}\left(\theta - \theta_0\right)$$

$$Z = Z(\theta) = \left[\bullet \;\; \bullet \;\; \bullet \;\; \left\{ Z_q \,\middle|\, q \in I_{\mathrm{Z}} \right\} \;\; \bullet \;\; \bullet \;\; \bullet \right]^{\mathrm{T}}$$

and

$$\theta = \left[\bullet \;\; \bullet \;\; \bullet \;\; \left\{ \theta_p \,\middle|\, p \in I_{\theta} \right\} \;\; \bullet \;\; \bullet \;\; \bullet \right]^{\mathrm{T}}$$

$$I_{\mathrm{Z}} = \{1, \; 2, \; 3, \; 4, \; 5, \; 6\} \quad \text{where} \quad \mathrm{N}_{\mathrm{Z}} = 6$$

$$I_{\theta} = \{1, \; 2, \; 3, \; 4, \; 5, \; 6\} \quad \text{where} \quad \mathrm{N}_{\theta} = 6$$

then

$$Z = Z(\theta) = \left[Z_1, \; Z_2, \; Z_3, \; Z_4, \; Z_5, \; Z_6 \right]^{\mathrm{T}}$$

$$\theta = \left[\theta_1, \; \theta_2, \; \theta_3, \; \theta_4, \; \theta_5, \; \theta_6 \right]^{\mathrm{T}}$$

Subject to :

$$\left.\begin{aligned}&\theta_{\mathrm{MIN}_p} \le \theta_p \le \theta_{\mathrm{MAX}_p}\\&\theta_{\mathrm{MIN}_p} \in (-\infty\,,+\infty)\\&\theta_{\mathrm{MAX}_p} \in (-\infty\,,+\infty)\end{aligned}\right\}\left\{\begin{array}{l}\text{Direct Constraints on the Control}\\ \theta-\text{vector Elements } \theta_p \text{ for: } \quad p \in I_\theta\end{array}\right.$$

$$\left[\ \phi_1(\theta) = \theta_1\theta_4 - \theta_2\theta_3 = 0\ \right]\ \Big\}\left\{\begin{array}{l}\text{Equality Constraint } \phi(\theta)-\text{vector for:}\\ \qquad N_{\mathrm{EQ}} = 1\end{array}\right.$$

$$\left[\begin{aligned}\psi_1(\theta) &= \psi_{1_{\mathrm{MAX}}} - \sqrt{\theta_1^2+\theta_2^2} \ge 0\\ \psi_2(\theta) &= \psi_{2_{\mathrm{MAX}}} - \sqrt{\theta_3^2+\theta_4^2} \ge 0\\ \psi_3(\theta) &= \psi_{3_{\mathrm{MAX}}} - \sqrt{\theta_5^2+\theta_6^2} \ge 0\end{aligned}\right]\left\}\left\{\begin{array}{l}\text{First Inequality Constraint}\\ {}^1\psi(\theta)-\text{sub-vector for:} \quad N_{\mathrm{IEQ}_1} = 3\end{array}\right.\right.$$

$$\left[\begin{aligned}\psi_4(\theta) &= \psi_{4_{\mathrm{MAX}}} - \left|\theta_1-\theta_{0_1}\right| \ge 0\\ \psi_5(\theta) &= \psi_{5_{\mathrm{MAX}}} - \left|\theta_2-\theta_{0_2}\right| \ge 0\\ \psi_6(\theta) &= \psi_{6_{\mathrm{MAX}}} - \left|\theta_3-\theta_{0_3}\right| \ge 0\\ \psi_7(\theta) &= \psi_{7_{\mathrm{MAX}}} - \left|\theta_4-\theta_{0_4}\right| \ge 0\\ \psi_8(\theta) &= \psi_{8_{\mathrm{MAX}}} - \left|\theta_5-\theta_{0_5}\right| \ge 0\\ \psi_9(\theta) &= \psi_{9_{\mathrm{MAX}}} - \left|\theta_6-\theta_{0_6}\right| \ge 0\end{aligned}\right]\left\}\left\{\begin{array}{l}\text{Second Inequality Constraint}\\ {}^2\psi(\theta)-\text{sub-vector for:} \quad N_{\mathrm{IEQ}_2} = 6\end{array}\right.\right.$$

The (6 x 6) T-Matrix NLP control problem has four sub-problems: (1) unconstrained optimisation, (2) optimisation with only equality constraints, (3) optimisation with only inequality constraints, and (4) optimisation with both equality constraints and inequality constraints. It was assumed that the tasking of these T-Matrix NLP control problems occurred within the framework of real-time controller duty cycles. To expedite the testing, experimentation, and evaluation of the NLPQLP System, the required input data was synthetically determined using a process designed expressly for this analysis. Specifically, a previously identified T-Matrix and an actual control θ_0– vector/actual measurement Z_{A}– vector pair from a previous duty cycle were synthesised for use as input to these problems. Additionally, these already synthesised values were directly input to the (6 x 6) T-Matrix NLP control problems for comparison purposes. The classic regulator problem was solved to verify the NLP solutions to the unconstrained optimisation NLP problems. Agreement was obtained in all cases.

Listings of the command (DCL) file code and the Fortran main driver code for the Hewlett-Packard Alpha mainframe computer, and the input and output for the four sub-problems that were part of the (6 x 6) T-Matrix NLP control problems solved using the Hewlett-Packard Alpha mainframe computer, are presented in Appendix B.2. A listing of the Fortran main driver code for the Mac Pro desktop computer, and the input and output for the four sub-problems that were part of the (6 x 6) T-Matrix NLP control problems solved using the Mac Pro desktop computer, are presented in Appendix C.2.

These (6 x 6) T-Matrix NLP control problems are representative of actual rotorcraft control problems. The solutions to these problems were sufficiently fast to be included in real-time duty cycles.

2.2.3 (24 x 8) T-Matrix NLP Control Problems

For the (24 x 8) T-Matrix NLP control problem, the number of elements (dimension) in the control θ – vector is 8 (i.e., N_θ = 8), and the number of elements (dimension) in the predicted measurement Z – vector is 24 (i.e., N_Z = 24). Correspondingly, the system or transfer T-Matrix is (24 x 8). The number of elements (dimension) in the Equality Constraint $\phi(\theta)$ –vector is 2 (i.e., N_{EQ} = 2), the number of elements in the First Inequality Constraint ${}^1\psi(\theta)$ – sub-vector is 4 (i.e., N_{IEQ_1} = 4), the number of elements in the Second Inequality Constraint ${}^2\psi(\theta)$ –sub-vector function is 8 (i.e., N_{IEQ_2} = 8), and the dimension of the Inequality Constraint $\psi(\theta)$ –vector is 12 (i.e., $N_{IEQ} = N_{IEQ_1} + N_{IEQ_2}$ = 12). The (24 x 8) T-Matrix NLP control problem with all constraints is defined as:

Determine the θ – vector, θ_{Sol} , that solves the problem:

$$\underset{\theta_p \in \theta}{Minimise} \quad J = Z^{T} W_Z Z \qquad for \quad p \in I_\theta$$

$$where \qquad Z = Z(\theta) = Z_A + \mathrm{T}\left(\theta - \theta_0\right)$$

$$Z = Z(\theta) = \left[\bullet \;\; \bullet \;\; \bullet \;\; \left\{Z_q \,\middle|\, q \in I_Z\right\} \;\; \bullet \;\; \bullet \;\; \bullet\right]^T$$

$$and \qquad \theta = \left[\bullet \;\; \bullet \;\; \bullet \;\; \left\{\theta_p \,\middle|\, p \in I_\theta\right\} \;\; \bullet \;\; \bullet \;\; \bullet\right]^T$$

$$I_Z = \left\{1, \;\; 2, \;\; 3, \;\; \bullet \;\; \bullet \;\; \bullet \;\; 24\right\} \quad \text{where} \quad N_Z = 24$$

$$I_\theta = \left\{1, \;\; 2, \;\; 3, \;\; \bullet \;\; \bullet \;\; \bullet \;\; 8\right\} \quad \text{where} \quad N_\theta = 8$$

then $$Z = Z(\theta) = \left[Z_1, \ \ Z_2, \ \ Z_3, \ \ \bullet \ \bullet \ \bullet \ \ Z_{24} \right]^{\mathrm{T}}$$

$$\theta = \left[\theta_1, \ \ \theta_2, \ \ \theta_3, \ \ \bullet \ \bullet \ \bullet \ \ \theta_8 \right]^{\mathrm{T}}$$

Subject to :

$$\left.\begin{array}{l} \theta_{\mathrm{MIN}p} \le \theta_p \le \theta_{\mathrm{MAX}p} \\ \theta_{\mathrm{MIN}p} \in (-\infty \, , +\infty) \\ \theta_{\mathrm{MAX}p} \in (-\infty \, , +\infty) \end{array}\right\} \left\{ \begin{array}{l} \text{Direct Constraints on the Control} \\ \theta-\text{vector Elements } \theta_p \text{ for: } \quad p \in I_\theta \end{array}\right.$$

$$\left[\begin{array}{lcl} \phi_1(\theta) & = & \theta_1 \theta_4 - \theta_2 \theta_3 = 0 \\ \phi_2(\theta) & = & \theta_5 \theta_8 - \theta_6 \theta_7 = 0 \end{array}\right] \left\} \left\{ \begin{array}{l} \text{Equality Constraint } \phi(\theta) - \text{vector for:} \\ \qquad \mathrm{N_{EQ}} = 2 \end{array}\right.\right.$$

$$\left[\begin{array}{lcl} \psi_1(\theta) & = & \psi_{1_{\mathrm{MAX}}} - \sqrt{\theta_1^2 + \theta_2^2} \ge 0 \\ \psi_2(\theta) & = & \psi_{2_{\mathrm{MAX}}} - \sqrt{\theta_3^2 + \theta_4^2} \ge 0 \\ \psi_3(\theta) & = & \psi_{3_{\mathrm{MAX}}} - \sqrt{\theta_5^2 + \theta_6^2} \ge 0 \\ \psi_4(\theta) & = & \psi_{4_{\mathrm{MAX}}} - \sqrt{\theta_7^2 + \theta_8^2} \ge 0 \end{array}\right] \left\} \left\{ \begin{array}{l} \text{First Inequality Constraint} \\ {}^1\psi(\theta) - \text{sub-vector for:} \qquad \mathrm{N_{IEQ_1}} = 4 \end{array}\right.\right.$$

$$\left[\begin{array}{lcl} \psi_5(\theta) & = & \psi_{5_{\mathrm{MAX}}} - \left|\theta_1 - \theta_{0_1}\right| \ge 0 \\ \psi_6(\theta) & = & \psi_{6_{\mathrm{MAX}}} - \left|\theta_2 - \theta_{0_2}\right| \ge 0 \\ \psi_7(\theta) & = & \psi_{7_{\mathrm{MAX}}} - \left|\theta_3 - \theta_{0_3}\right| \ge 0 \\ & \bullet & \\ & \bullet & \\ & \bullet & \\ \psi_{12}(\theta) & = & \psi_{12_{\mathrm{MAX}}} - \left|\theta_8 - \theta_{0_8}\right| \ge 0 \end{array}\right] \left\} \left\{ \begin{array}{l} \text{Second Inequality Constraint} \\ {}^2\psi(\theta) - \text{sub-vector for:} \qquad \mathrm{N_{IEQ_2}} = 8 \end{array}\right.\right.$$

The (24 x 8) T-Matrix NLP control problem has four sub-problems: (1) unconstrained optimisation, (2) optimisation with only equality constraints, (3) optimisation with only inequality constraints, and (4) optimisation with both equality constraints and inequality constraints. It was assumed that the tasking of these T-Matrix NLP control problems occurred within the framework of real-time controller duty cycles. To expedite the testing, experimentation, and evaluation of the NLPQLP System, the required input data was synthetically determined using a process designed expressly for this analysis. Specifically, a previously identified T-Matrix and an actual control θ_0– vector/actual measurement Z_A– vector pair from a previous duty cycle were synthesised for use as input to these problems. Additionally, these already synthesised values were directly input to the (24 x 8) T-Matrix NLP control problems for comparison purposes. The classic regulator problem was solved to verify the NLP solutions to the unconstrained optimisation NLP problems. Agreement was obtained in all cases.

Listings of the command (DCL) file code and the Fortran main driver code for the Hewlett-Packard Alpha mainframe computer, and the input and output for the four sub-problems that were part of the (24 x 8) T-Matrix NLP control problems solved using the Hewlett-Packard Alpha mainframe computer, are presented in Appendix B, section B.3. A listing of the Fortran main driver code for the Mac Pro desktop computer, and the input and output for the four sub-problems that were part of the (24 x 8) T-Matrix NLP control problems solved using the Mac Pro desktop computer, are presented in Appendix C, section C.3.

These (24 x 8) T-Matrix NLP control problems are representative of actual rotorcraft control problems. The solutions to these problems were sufficiently fast to be included in real-time duty cycles.

2.2.4 (90 x 30) T-Matrix NLP Control Problems

For these problems, the number of elements (dimension) in the control θ – vector is 30 (i.e., N_θ = 30), and the number of elements (dimension) in the predicted measurement Z – vector is 90 (i.e., N_Z = 90). Correspondingly, the system or transfer T-Matrix is (90 x 30). The number of elements (dimension) in the Equality Constraint $\phi(\theta)$ –vector is 7 (i.e., N_{EQ} = 7), the number of elements in the First Inequality Constraint ${}^1\psi(\theta)$ –sub-vector is 15 (i.e., N_{IEQ_1} = 15), the number of elements in the Second Inequality Constraint ${}^2\psi(\theta)$ –sub-vector function is 30 (i.e., N_{IEQ_2} = 30), and the dimension of the Inequality Constraint $\psi(\theta)$ –vector is 45 (i.e., $N_{IEQ} = N_{IEQ_1} + N_{IEQ_2}$ = 45). The (90 x 30) T-Matrix NLP control problem with all constraints is defined as:

Determine the θ – vector, θ_{Sol} , that solves the problem:

$$\underset{\theta_p \in \theta}{Minimise} \quad J = Z^{\mathrm{T}} W_Z\, Z \qquad for \quad p \in I_\theta$$

where

$$Z = Z(\theta) = Z_{\mathrm{A}} + \mathrm{T}\left(\theta - \theta_0\right)$$

$$Z = Z(\theta) = \left[\bullet \;\; \bullet \;\; \bullet \;\; \left\{ Z_q \,\middle|\, q \in I_{\mathrm{Z}} \right\} \;\; \bullet \;\; \bullet \;\; \bullet \right]^{\mathrm{T}}$$

and

$$\theta = \left[\bullet \;\; \bullet \;\; \bullet \;\; \left\{ \theta_p \,\middle|\, p \in I_\theta \right\} \;\; \bullet \;\; \bullet \;\; \bullet \right]^{\mathrm{T}}$$

$$I_{\mathrm{Z}} = \left\{ 1, \;\; 2, \;\; 3, \;\; \bullet \;\; \bullet \;\; \bullet \;\; 90 \right\} \quad \text{where} \quad \mathrm{N_Z} = 90$$

$$I_\theta = \left\{ 1, \;\; 2, \;\; 3, \;\; \bullet \;\; \bullet \;\; \bullet \;\; 30 \right\} \quad \text{where} \quad \mathrm{N}_\theta = 30$$

then

$$Z = Z(\theta) = \left[Z_1, \;\; Z_2, \;\; Z_3, \;\; \bullet \;\; \bullet \;\; \bullet \;\; Z_{90} \right]^{\mathrm{T}}$$

$$\theta = \left[\theta_1, \;\; \theta_2, \;\; \theta_3, \;\; \bullet \;\; \bullet \;\; \bullet \;\; \theta_{30} \right]^{\mathrm{T}}$$

Subject to :

$$\left.\begin{array}{l} \theta_{\mathrm{MIN}_p} \le \theta_p \le \theta_{\mathrm{MAX}_p} \\ \theta_{\mathrm{MIN}_p} \in \left(-\infty \,,\, +\infty\right) \\ \theta_{\mathrm{MAX}_p} \in \left(-\infty \,,\, +\infty\right) \end{array}\right\} \left\{\begin{array}{l} \text{Direct Constraints on the Control} \\ \theta - \text{vector Elements } \theta_p \text{ for : } \quad p \in I_\theta \end{array}\right.$$

$$\begin{bmatrix} \phi_1(\theta) & = & \theta_1\theta_4 - \theta_2\theta_3 & = & 0 \\ \phi_2(\theta) & = & \theta_5\theta_8 - \theta_6\theta_7 & = & 0 \\ \phi_3(\theta) & = & \theta_9\theta_{12} - \theta_{10}\theta_{11} & = & 0 \\ \phi_4(\theta) & = & \theta_{13}\theta_{16} - \theta_{14}\theta_{15} & = & 0 \\ \phi_5(\theta) & = & \theta_{17}\theta_{20} - \theta_{18}\theta_{19} & = & 0 \\ \phi_6(\theta) & = & \theta_{21}\theta_{24} - \theta_{22}\theta_{23} & = & 0 \\ \phi_7(\theta) & = & \theta_{25}\theta_{28} - \theta_{26}\theta_{27} & = & 0 \end{bmatrix} \left\} \begin{array}{l} \text{Equality Constraint } \phi(\theta) - \text{vector for:} \\ \qquad N_{EQ} = 7 \end{array} \right.$$

$$\begin{bmatrix} \psi_1(\theta) & = & \psi_{1_{MAX}} & - & \sqrt{\theta_1^2 + \theta_2^2} & \geq & 0 \\ \psi_2(\theta) & = & \psi_{2_{MAX}} & - & \sqrt{\theta_3^2 + \theta_4^2} & \geq & 0 \\ \psi_3(\theta) & = & \psi_{3_{MAX}} & - & \sqrt{\theta_5^2 + \theta_6^2} & \geq & 0 \\ & & & \bullet & & & \\ & & & \bullet & & & \\ & & & \bullet & & & \\ \psi_{15}(\theta) & = & \psi_{15_{MAX}} & - & \sqrt{\theta_{29}^2 + \theta_{30}^2} & \geq & 0 \end{bmatrix} \left\} \begin{array}{l} \text{First Inequality Constraint} \\ {}^1\psi(\theta) - \text{sub-vector for:} \qquad N_{IEQ_1} = 15 \end{array} \right.$$

$$\begin{bmatrix} \psi_{16}(\theta) & = & \psi_{16_{MAX}} & - & \left|\theta_1 - \theta_{0_1}\right| & \geq & 0 \\ \psi_{17}(\theta) & = & \psi_{17_{MAX}} & - & \left|\theta_2 - \theta_{0_2}\right| & \geq & 0 \\ \psi_{18}(\theta) & = & \psi_{18_{MAX}} & - & \left|\theta_3 - \theta_{0_3}\right| & \geq & 0 \\ & & & \bullet & & & \\ & & & \bullet & & & \\ & & & \bullet & & & \\ \psi_{45}(\theta) & = & \psi_{45_{MAX}} & - & \left|\theta_{30} - \theta_{0_{30}}\right| & \geq & 0 \end{bmatrix} \left\} \begin{array}{l} \text{Second Inequality Constraint} \\ {}^2\psi(\theta) - \text{sub-vector for:} \qquad N_{IEQ_2} = 30 \end{array} \right.$$

The (90 x 30) T-Matrix NLP control problem has four sub-problems: (1) unconstrained optimisation, (2) optimisation with only equality constraints, (3) optimisation with only inequality constraints, and (4) optimisation with both equality constraints and inequality constraints. It was assumed that the tasking of these T-Matrix NLP control problems occurred within the framework of real-time controller duty cycles. To expedite the testing, experimentation, and evaluation of the NLPQLP System, the required input data was synthetically determined using a process designed expressly for this analysis. Specifically, a previously identified T-Matrix and an actual control θ_0– vector/actual measurement Z_A– vector pair from a previous duty cycle were synthesised for use as input to these problems. Additionally, these already synthesised values were directly input to the (90 x 30) T-Matrix NLP control problems for comparison purposes. The classic regulator problem was solved to verify the NLP solutions to the unconstrained optimisation NLP problems. Agreement was obtained in all cases.

Listings of the command (DCL) file code and the Fortran main driver code for the Hewlett-Packard Alpha mainframe computer, and the input and output for the four sub-problems that were part of the (90 x 30) T-Matrix NLP control problems solved using the Hewlett-Packard Alpha mainframe computer, are presented in Appendix B, section B.4. A listing of the Fortran main driver code for the Mac Pro desktop computer, and the input and output for the four sub-problems that were part of the (90 x 30) T-Matrix NLP control problems solved using the Mac Pro desktop computer, are presented in Appendix C, section C.4.

These (90 x 30) T-Matrix NLP control problems are representative of aerodynamic surface design and/or aircraft configuration problems and, although they were solved rapidly, they are more suitable to non-real-time design applications.

2.2.5 (90 x 60) T-Matrix NLP Control Problems

For these problems, the number of elements (dimension) in the control θ – vector is 60 (i.e., N_θ = 60), and the number of elements (dimension) in the predicted measurement Z – vector is 90 (i.e., N_Z = 90). Correspondingly, the system or transfer T-Matrix is (90 x 60). The number of elements (dimension) in the Equality Constraint $\phi(\theta)$ –vector is 15 (i.e., N_{EQ} = 15), the number of elements in the First Inequality Constraint ${}^1\psi(\theta)$ –sub-vector is 30 (i.e., N_{IEQ_1} = 30), the number of elements in the Second Inequality Constraint ${}^2\psi(\theta)$ –sub-vector function is 60 (i.e., N_{IEQ_2} = 60), and the dimension of the Inequality Constraint $\psi(\theta)$ –vector is 90 (i.e., $N_{IEQ} = N_{IEQ_1} + N_{IEQ_2}$ = 90). The (90 x 60) T-Matrix NLP control problem with all constraints is defined as:

Determine the θ – vector, θ_{Sol}, that solves the problem:

$$\underset{\theta_p \in \theta}{Minimise} \quad J = Z^{\mathsf{T}} W_Z Z \quad for \quad p \in I_\theta$$

where

$$Z = Z(\theta) = Z_A + \mathsf{T}(\theta - \theta_0)$$

$$Z = Z(\theta) = \left[\bullet \ \bullet \ \bullet \ \left\{ Z_q \,\middle|\, q \in I_Z \right\} \ \bullet \ \bullet \ \bullet \right]^{\mathsf{T}}$$

and

$$\theta = \left[\bullet \ \bullet \ \bullet \ \left\{ \theta_p \,\middle|\, p \in I_\theta \right\} \ \bullet \ \bullet \ \bullet \right]^{\mathsf{T}}$$

$$I_Z = \{1, \ 2, \ 3, \ \bullet \ \bullet \ \bullet \ 90\} \quad \text{where} \quad N_Z = 90$$

$$I_\theta = \{1, \ 2, \ 3, \ \bullet \ \bullet \ \bullet \ 60\} \quad \text{where} \quad N_\theta = 60$$

then

$$Z = Z(\theta) = \left[Z_1, \ Z_2, \ Z_3, \ \bullet \ \bullet \ \bullet \ Z_{90}\right]^{\mathsf{T}}$$

$$\theta = \left[\theta_1, \ \theta_2, \ \theta_3, \ \bullet \ \bullet \ \bullet \ \theta_{60}\right]^{\mathsf{T}}$$

Subject to:

$$\left.\begin{array}{l} \theta_{\text{MIN}_p} \le \theta_p \le \theta_{\text{MAX}_p} \\ \theta_{\text{MIN}_p} \in (-\infty, +\infty) \\ \theta_{\text{MAX}_p} \in (-\infty, +\infty) \end{array}\right\} \left\{\begin{array}{l} \text{Direct Constraints on the Control} \\ \theta-\text{vector Elements } \theta_p \text{ for: } \quad p \in I_\theta \end{array}\right.$$

$$
\begin{bmatrix}
\phi_1(\theta) & = & \theta_1\theta_4 - \theta_2\theta_3 & = & 0 \\
\phi_2(\theta) & = & \theta_5\theta_8 - \theta_6\theta_7 & = & 0 \\
\phi_3(\theta) & = & \theta_9\theta_{12} - \theta_{10}\theta_{11} & = & 0 \\
 & & \bullet & & \\
 & & \bullet & & \\
 & & \bullet & & \\
\phi_{15}(\theta) & = & \theta_{57}\theta_{60} - \theta_{58}\theta_{59} & = & 0
\end{bmatrix}
\left\{ \begin{array}{l} \text{Equality Constraint } \phi(\theta) - \text{vector for:} \\ \qquad N_{EQ} = 15 \end{array} \right.
$$

$$
\begin{bmatrix}
\psi_1(\theta) & = & \psi_{1_{MAX}} & - & \sqrt{\theta_1^2 + \theta_2^2} & \geq & 0 \\
\psi_2(\theta) & = & \psi_{2_{MAX}} & - & \sqrt{\theta_3^2 + \theta_4^2} & \geq & 0 \\
\psi_3(\theta) & = & \psi_{3_{MAX}} & - & \sqrt{\theta_5^2 + \theta_6^2} & \geq & 0 \\
 & & & \bullet & & & \\
 & & & \bullet & & & \\
 & & & \bullet & & & \\
\psi_{30}(\theta) & = & \psi_{30_{MAX}} & - & \sqrt{\theta_{59}^2 + \theta_{60}^2} & \geq & 0
\end{bmatrix}
\left\{ \begin{array}{l} \text{First Inequality Constraint} \\ {}^1\psi(\theta) - \text{sub-vector for:} \quad N_{IEQ_1} = 30 \end{array} \right.
$$

$$
\begin{bmatrix}
\psi_{31}(\theta) & = & \psi_{31_{MAX}} & - & \left|\theta_1 - \theta_{0_1}\right| & \geq & 0 \\
\psi_{32}(\theta) & = & \psi_{32_{MAX}} & - & \left|\theta_2 - \theta_{0_2}\right| & \geq & 0 \\
\psi_{33}(\theta) & = & \psi_{33_{MAX}} & - & \left|\theta_3 - \theta_{0_3}\right| & \geq & 0 \\
 & & & \bullet & & & \\
 & & & \bullet & & & \\
 & & & \bullet & & & \\
\psi_{90}(\theta) & = & \psi_{90_{MAX}} & - & \left|\theta_{60} - \theta_{0_{60}}\right| & \geq & 0
\end{bmatrix}
\left\{ \begin{array}{l} \text{Second Inequality Constraint} \\ {}^2\psi(\theta) - \text{sub-vector for:} \quad N_{IEQ_2} = 60 \end{array} \right.
$$

The (90 x 60) T-Matrix NLP control problem has four sub-problems: (1) unconstrained optimisation, (2) optimisation with only equality constraints, (3) optimisation with only inequality constraints, and (4) optimisation with both equality constraints and inequality constraints. It was assumed that the tasking of these T-Matrix NLP control problems occurred within the framework of real-time controller duty cycles. To expedite the testing, experimentation, and evaluation of the NLPQLP System, the required input data was synthetically determined using a process designed expressly for this analysis. Specifically, a previously identified T-Matrix and an actual control θ_0– vector/actual measurement Z_A– vector pair from a previous duty cycle were synthesised for use as input to these problems. Additionally, these already synthesised values were directly input to the (90 x 60) T-Matrix NLP control problems for comparison purposes. The classic regulator problem was solved to verify the NLP solutions to the unconstrained optimisation NLP problems. Agreement was obtained in all cases.

Listings of the command (DCL) file code and the Fortran main driver code for the Hewlett-Packard Alpha mainframe computer, and the input and output for the four sub-problems that were part of the (90 x 60) T-Matrix NLP control problems solved using the Hewlett-Packard Alpha mainframe computer, are presented in Appendix B, section B.5. A listing of the Fortran main driver code for the Mac Pro desktop computer, and the input and output for the four sub-problems that were part of the (90 x 60) T-Matrix NLP control problems solved using the Mac Pro desktop computer, are presented in Appendix C, section C.5.

These (90 x 60) T-Matrix NLP control problems are representative of aerodynamic surface design and/or aircraft configuration problems and, although they were solved rapidly, they are more suitable to non-real-time design applications.

2.3 Synthetic Data

The specific problems that were solved during this study were representative of actual problems to be solved during experimentation and/or testing of various rotorcraft configurations. It is assumed that the tasking of these T-Matrix NLP control problems occurs within the framework of real-time controller duty cycles. Tasking of one of these T-Matrix NLP control problems during the current duty cycle requires: (1) a previously identified T-Matrix, and (2) the actual control θ_0– vector and the actual measurement Z_A– vector pair determined during a previous duty cycle or at a reference epoch time. Preparations for such experimentation and/or testing requires the definition of the specific NLP problem (i.e., the performance index, the dimension and elements of the control vector and the measurement vector, and all constraints) to be solved. The main driver program, which defines the problem to be solved and tasks the NLPQLP System to solve it, must be coded, verified for proper functioning, and tuned-up so that it reliably and efficiently solves the required problems during actual tests. This, of course, must precede the test and the generation of actual associated test data.

In an actual wind tunnel or flight test, the identification and/or acquisition of the required data would normally be tasked during a previous duty cycle or at a reference epoch time and would be transmitted to the NLP program host computer for the solution of the NLP problem. The solution control θ_{Sol}– vector would then be transmitted back to the controller during the current duty cycle. It is necessary to have input/output (I/O) interface/compatibility between the controller and the NLP host computer in order to successfully transmit usable data between the controller and the NLP host computer. This issue is not addressed in this document.

Although it might be possible to use similar test data from another test for verification and tuning purposes, in general this is cumbersome and can be unnecessarily time consuming. In addition, there could be I/O interface/compatibility issues associated with use of test data from another test. Synthetic determination of the T-Matrix, the actual control θ_0– vector, and the actual measurement Z_A– vector data avoids problems associated with use of test data from another test and provides a simple, rapid method to obtain the required data during verification and tuning.

A synthesis procedure was designed to provide the T-Matrix and a previous duty cycle "actual" control θ_0– vector/"actual" measurement Z_A– vector pair required for the verification of the NLP main driver codes and subsequent tuning of the operation of these codes to solve the type of problems expected during the test. In order to develop a realistic test of the main driver codes and the NLPQLP System, a small degree of randomness was included in the synthetic modelling. Uniformly distributed pseudo-random numbers were selected for this synthesis process (see section 2.3.1).

First, the T-Matrix is determined by defining its elements to be pseudo-random numbers between –1.0 inclusive and +1.0 exclusive (see section 2.3.2). Scaling coefficients are provided to ensure that the norm to the T-Matrix is within acceptable limits. Next, the "actual" control θ_0– vector is likewise determined by defining its elements to be pseudo-random numbers between –1.0 inclusive and +1.0 exclusive (see section 2.3.2). In this case, the elements are constrained to be within specified limits à la "external limiting." Although the definition of the T-Matrix and the "actual" control θ_0– vector was accomplished using pseudo-random numbers, these elements did not have to be defined randomly and could have been directly input or generated otherwise. These elements were generated randomly for convenience. The essential requirements for these elements are that the scaling and limits be reasonable and the resulting values are realistic. The degree of randomness to the model is introduced in the definition of the "actual" measurement Z_A– vector. The "actual" measurement Z_A– vector is defined by adding uniformly distributed pseudo-random numbers to the Tθ_0 product used in the definition of the Z_A– vector (see section 2.3.2).

2.3.1 Definition of the Random Number Generator Function RAN(SEED)

The random number generator function, RAN(●), employed for data synthesis from both the Hewlett-Packard Alpha mainframe computer and Mac Pro desktop computer, produces a real uniformly pseudo-random number between 0.0 inclusive and 1.0 exclusive (i.e., $\text{RAN}(\bullet) \in [0,1)$). Because the algorithm employed by RAN(●) on the Hewlett-Packard Alpha mainframe computer is a VMS System subroutine, it is not necessary to provide this subroutine for the main driver code on that computer. In order to provide nearly identical main driver codes, and because the corresponding G95 intrinsic uniform pseudo-random number generator function is named RAND(●), a Fortran code for a RAN(●) subroutine that calls RAND(●) is provided in addition to the main driver code for the Mac Pro desktop computer. It is, however, necessary to define this RAN(●) with EXTERNAL and REAL statements in the Mac Pro main driver codes. The provision of the EXTERNAL and REAL statements in the Mac Pro main driver codes is the only difference from the corresponding main driver codes for the Hewlett-Packard Alpha mainframe computer. These RAN(●) codes employ different algorithms to generate the pseudo-random numbers, and correspondingly will, in general, produce different numerical values.

Both of these RAN(●) random number generators require provision of a seed in the calling argument. This argument (i.e., the seed) should initially be set to a large odd-integer value for the first call to RAN(●). Its value will be updated during this call to RAN(●) to be used in the next call to RAN(●).

2.3.2 Determination of the Synthetic T-Matrix, the "Actual" Control θ_0– Vector, and the "Actual" Measurement Z_A–Vector

The methodology employed to synthetically determine the T-Matrix, the actual control θ_0– vector, and the actual measurement Z_A– vector data is:

First, synthesise the T-Matrix *according to:*

$$T_{q,p} = C_1\left[2 \bullet RAN\left(ISEED1_k\right) - 1\right] + C_2\left[2 \bullet RAN\left(JSEED1_l\right) - 1\right]$$

where

$$k = \textit{is the index number of the input seed for the first call to } RAN\left(\bullet\right).$$

$$l = \textit{is the index number of the input seed for the second call to } RAN\left(\bullet\right).$$

and

$$T = \begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & T_{q,p} & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} \textit{ is } \left(N_Z \times N_\theta\right) \quad \forall\ q \in I_Z \quad \textit{and} \quad \forall\ p \in I_\theta$$

Next, synthesise the "actual" θ_0 – Vector *according to*:

$$\theta_{p_{Initial}} = C_3\left[2 \bullet RAN\left(ISEED2_k\right) - 1\right] + C_4\left[2 \bullet RAN\left(JSEED2_l\right) - 1\right] \quad \forall\ p \in I_\theta$$

Subject to:

$$\left.\begin{matrix} \theta_{MIN_p} \le \theta_{p_{Initial}} \le \theta_{MAX_p} \\ \theta_{MIN_p} \in \left(-\infty, +\infty\right) \\ \theta_{MAX_p} \in \left(-\infty, +\infty\right) \end{matrix}\right\} \left\{\begin{matrix} \text{Direct Constraints on the Control} \\ \theta-\text{vector Elements } \theta_p \text{ for:} \end{matrix}\right. \quad p \in I_\theta$$

if $\theta_{p_{Initial}} \le \theta_{MIN_p} + \varepsilon$ *then* $\theta_{p_{Initial}} = \theta_{MIN_p} + \varepsilon$

if $\theta_{p_{Initial}} \ge \theta_{MAX_p} - \varepsilon$ *then* $\theta_{p_{Initial}} = \theta_{MIN_p} - \varepsilon$

otherwise there is no change to the value of $\theta_{p_{Initial}}$ *as determined by the random equation above.*

then

$$\theta_0 = \left[\bullet \;\bullet\; \bullet \;\left\{\theta_{p_{Initial}} \,\middle|\, p \in I_\theta\right\}\; \bullet \;\bullet\; \bullet\right]^{\mathrm{T}}$$

Finally, synthesise the "actual" Z_{A} – Vector *according to*:

$$\Delta Z_{\mathrm{A}_q} = \mathrm{C}_5\left[2 \bullet \mathrm{RAN}\left(\mathrm{ISEED3}_k\right) - 1\right] \;+\; \mathrm{C}_6\left[2 \bullet \mathrm{RAN}\left(\mathrm{JSEED3}_l\right) - 1\right]$$

$$\forall \;\; q \in I_{\mathrm{Z}}$$

then

$$\Delta Z_{\mathrm{A}} = \left[\bullet \;\bullet\; \bullet \;\left\{\Delta Z_{\mathrm{A}_q} \,\middle|\, q \in I_{\mathrm{Z}}\right\}\; \bullet \;\bullet\; \bullet\right]^{\mathrm{T}}$$

and

$$Z_{\mathrm{A}} = \mathrm{T}\,\theta_0 \;+\; \Delta Z_{\mathrm{A}}$$

where

$\mathrm{RAN}\left(\,\bullet\,\right)$ *is the Uniform Random Number Distribution Function which yields a uniformly random real number* $\in \left[0, 1\right)$

$$I_{\mathrm{Z}} = \left\{\bullet \;\bullet\; \bullet \;\left\{\forall \;\; q \;\ni\; Z_q \in Z\right\}\; \bullet \;\bullet\; \bullet\right\}$$

$$I_\theta = \left\{\bullet \;\bullet\; \bullet \;\left\{\forall \;\; p \;\ni\; \theta_p \in \theta\right\}\; \bullet \;\bullet\; \bullet\right\}$$

2.4 The Regulator Problem

Unlike the non-linear programming problems that are constrained/unconstrained optimisation problems, the regulator problem is a steady-state problem; its solution process seeks maintenance of a steady-state condition with minimal control and, in some cases, minimal control rate of change. Because the regulator problem solution is analytically explicit and known, its solution process is fast and has a relatively small computational load compared to the constrained/unconstrained optimisation solution processes. In some cases, “External Limiting” constraints (i.e., direct maximum limits on control vector elements imposed *after* the analytic explicit solution is obtained) are imposed on control vector elements. Correspondingly, early attempts (i.e., circa the 1950s) to solve constrained/unconstrained optimisation control problems formulated these problems as regulator problems because the algorithms for constrained/ unconstrained optimisation solution processes and computer technology were in their early stages of development and not sufficiently reliable or efficient for this purpose. As computer technology advances occurred and efficient, reliable optimisation techniques were developed, use of numerical optimisation techniques became feasible for actual test applications.

A control vector metric, and a rate of change of the control vector metric if required, are adjoined to a steady-state excursion metric to form the performance index for the regulator problem. By appropriately defining the steady-state excursion metric, which is the first term in the performance index, and carefully tuning the weighting coefficients for all the terms in the performance index, a pseudo-optimal solution can be obtained that satisfies, or nearly satisfies, any required constraints. This tuning must, of course, be accomplished before actual test applications.

As in the case of the General T-Matrix NLP control problem described in section 2.2, a T-Matrix linear plant model that relates the measurement $Z-$vector to the control $\theta-$vector is assumed for the regulator problem described below. The first-term performance index is the steady-state excursion metric and a quadratic function of a T-Matrix plant model, and correspondingly a quadratic function of the control $\theta-$vector. The second term in the performance index is simply a weighted control $\theta-$vector quadratic. If required, the third term in the performance index is a weighted time rate of change of the control $\theta-$vector quadratic. The regulator problem is:

Determine the θ – vector, θ_{Sol} , that solves the problem:

$$\underset{\theta_p \in \theta}{\text{Minimise}} \qquad J = Z^{\mathsf{T}} W_Z Z + \theta^{\mathsf{T}} W_\theta \theta + \dot{\theta}^{\mathsf{T}} W_{\dot{\theta}} \dot{\theta} \qquad \textit{for} \quad p \in I_\theta$$

where $\quad Z = Z(\theta) = Z_{\mathsf{A}} + \mathsf{T}\left(\theta - \theta_0\right)$

$$Z = Z(\theta) = \left[\bullet \;\bullet\; \bullet \;\left\{Z_q \,\middle|\, q \in I_{\mathsf{Z}}\right\}\; \bullet\; \bullet\; \bullet\right]^{\mathsf{T}}$$

and $\quad \theta = \left[\bullet \;\bullet\; \bullet \;\left\{\theta_p \,\middle|\, p \in I_\theta\right\}\; \bullet\; \bullet\; \bullet\right]^{\mathsf{T}}$

$$\dot{\theta} = \left[\bullet \;\bullet\; \bullet \;\left\{\dot{\theta}_p \,\middle|\, p \in I_\theta\right\}\; \bullet\; \bullet\; \bullet\right]^{\mathsf{T}}$$

$$I_{\mathsf{Z}} = \left\{\bullet \;\bullet\; \bullet \;\left\{\forall\; q \ni Z_q \in Z\right\}\; \bullet\; \bullet\; \bullet\right\}$$

$$I_\theta = \left\{\bullet \;\bullet\; \bullet \;\left\{\forall\; p \ni \theta_p \in \theta\right\}\; \bullet\; \bullet\; \bullet\right\}$$

Subject to NO constraints per se during the regulator problem solution process to determine θ_{Sol} .

The Explicit Solution θ_{Sol} *– vector is* :

$$\theta_{Sol} = \left(D\, W_{\dot{\theta}} + D\, \mathsf{T}^{\mathsf{T}} W_Z\, \mathsf{T}\right)\theta_0 - \alpha D\, \mathsf{T}^{\mathsf{T}} W_Z Z_{\mathsf{A}}$$

where $\quad D = \left(\mathsf{T}^{\mathsf{T}} W_Z\, \mathsf{T} + W_{\dot{\theta}} + W_\theta\right)^{-1}$

and $\quad \alpha \in \left[0,\, 1\right]$

In some cases, "External Limiting" constraints are imposed on θ_{Sol} *after the explicit analytic solution for* θ_{Sol} *is determined. Specifically:*

$$\left.\begin{array}{l} \theta_{\mathrm{MIN}_p} \le \theta_{Solp} \le \theta_{\mathrm{MAX}_p} \\ \theta_{\mathrm{MIN}_p} \in (-\infty , +\infty) \\ \theta_{\mathrm{MAX}_p} \in (-\infty , +\infty) \end{array}\right\} \begin{cases} \text{Direct Constraints on the Control} \\ \theta_{Sol} - \text{vector Elements } \theta_{Solp} \text{ for: } p \in I_\theta \end{cases}$$

$$\textit{if} \quad \theta_{Sol_p} \le \theta_{\mathrm{MIN}_p} \qquad \textit{then set} \quad \theta_{Sol_p} = \theta_{\mathrm{MIN}_p}$$

$$\textit{if} \quad \theta_{Sol_p} \ge \theta_{\mathrm{MAX}_p} \qquad \textit{then set} \quad \theta_{Sol_p} = \theta_{\mathrm{MIN}_p}$$

otherwise there is no change to the value of θ_{Sol_p} *as determined by the equation for the explicit solution for* θ_{Sol_p}.

then

$$\theta_{Sol} = \left[\bullet \quad \bullet \quad \bullet \quad \left\{ \theta_{Sol_p} \,\middle|\, p \in I_\theta \right\} \quad \bullet \quad \bullet \quad \bullet \right]^{\mathrm{T}}$$

3.0 Results and Conclusions

The newly acquired Version 3.1 of the NLPQLP System was used to solve several typical constrained and unconstrained optimisation problems of the type encountered in various rotorcraft wind tunnel and flight tests on both the Hewlett-Packard Alpha mainframe computer and the Mac Pro desktop computer. The associated software and codes installed on the Mac Pro desktop computer should be transportable to a Mac laptop computer for use in a wind tunnel. A linear dependence (i.e., a T-Matrix plant model) of the measurement vector (the measurement Z – vector) on the control vector (i.e., the control θ – vector) was assumed. These problems ranged from a relatively simple, unconstrained 4-vector control problem, to a relatively large, constrained 60-vector control problem. Solutions were obtained for all problems considered. Although tuning and some input adjustments were required to successfully solve the large, constrained 60-vector control problem, the NLPQLP System proved to be an efficient and reliable method to solve these problems.

The problems solved in this analysis included: (6 x 4) T-Matrix NLP control problems, (6 x 6) T-Matrix NLP control problems, (24 x 8) T-Matrix NLP control problems, (90 x 30) T-Matrix NLP control problems, and (90 x 60) T-Matrix NLP control problems. Each of these problems had four sub-problems: (1) unconstrained optimisation, (2) optimisation with only equality constraints, (3) optimisation with only inequality constraints, and (4) optimisation with both equality constraints and inequality constraints. It was assumed that the tasking of these T-Matrix NLP control problems occurred within the framework of real-time controller duty cycles. To expedite the testing, experimentation, and evaluation of the NLPQLP System, the required input data was synthetically determined using a process designed expressly for this analysis. Specifically, a previously identified T-Matrix and an actual control θ_0– vector/actual measurement Z_A– vector pair from a previous duty cycle, or at a reference epoch time, were synthesised for use as input to these problems. Additionally, these already synthesised values were directly input to the (6 x 4), (6 x 6), and (24 x 8) T-Matrix NLP control problems for comparison purposes. The classic regulator problem was solved to verify the NLP solutions to the unconstrained optimisation NLP problems. Agreement was obtained in all cases.

The (6 x 4), (6 x 6), and (24 x 8) T-Matrix NLP control problems are representative of actual rotorcraft control problems. The solutions to these problems were sufficiently fast to be included in real-time duty cycles. The (90 x 30) and (90 x 60) T-Matrix NLP problems are representative of aerodynamic surface design and/or aircraft configuration problems and, although they were solved rapidly, they are more suitable to non-real-time design applications.

The results are shown in Appendix B and Appendix C, separate volumes of this report.

Listings of the command (DCL) file code and the Fortran main driver code for the Hewlett-Packard Alpha mainframe computer, and the input and output for the four sub-problems that were part of the (6 x 4), (6 x 6), (24 x 8), (90 x 30), and (90 x 60) T-Matrix NLP control problems solved using the Hewlett-Packard Alpha mainframe computer, are presented in Appendix B. A listing of the Fortran main driver code for the Mac Pro desktop computer, and the input and output for the four sub-problems that were part of the (6 x 4), (6 x 6), (24 x 8), (90 x 30), and (90 x 60) T-Matrix NLP control problems solved using the Mac Pro desktop computer, are presented in Appendix C.

4.0 References

1. Leyland, J. A.: A Higher Harmonic Optimal Controller to Optimise Rotorcraft Aeromechanical Behaviour. NASA Technical Memorandum 110390, Mar. 1996.

2. Leyland, J. A.: A Closed-Loop Optimal Neural-Network Controller to Optimise Rotorcraft Aeromechanical Behaviour, Volume 1, Theory and Methodology. NASA TM-2001-209622, Mar. 2001.

3. Leyland, J. A.: A Closed-Loop Optimal Neural-Network Controller to Optimise Rotorcraft Aeromechanical Behaviour, Volume 2, Output from Two Sample Cases. NASA TM-2001-209623, Mar. 2001.

4. Leyland, J. A.: A General Regularisation Functional to Enhance the Update Convergence of a Neural-Network Rotorcraft Behaviour Model, Volume 1, Theory and Methodology. NASA TM-2002-211843, Oct. 2002.

5. Leyland, J. A.: A General Regularisation Functional to Enhance the Update Convergence of a Neural-Network Rotorcraft Behaviour Model, Volume 2, Output From Two Sample Cases. NASA TM-2002-211844, Oct. 2002.

6. Anon.: IMSL MATH/LIBRARY User's Manual, FORTRAN Subroutines for Mathematical Applications, Volume 3, Chapter 8: Nonlinearly Constrained Minimization Using Finite Difference Gradients. Version 1.1, MALB-USM-UNBND-EN8901-1.1, pp. 895–902, Jan. 1989.

7. Schittkowski. K.: Non-linear Programming Codes. *Lecture Notes in Economics and Mathematics*, **183**, Springer-Verlag, Berlin, Germany, 1980.

8. Schittkowski, K.: On the Convergence of a Sequential Quadratic Programming Method With an Augmented Lagrangian Line Search Function. *Mathematik Operationsforschung und Statistik, Serie Optimization*, **14**, pp. 197–216, 1983.

9. Schittkowski, K.: NLPQL: a FORTRAN Subroutine solving Constrained Non-linear Programming Problems. (Clyde L. Monma, Ed.), *Annals of Operations Research*, **5**, pp. 485–500, 1986.

10. Gill, P. E.; Murray, W.; Saunders, M. A.; and Wright, M. H.: Model Building and Practical Aspects of Non-Linear Programming. *Computational Mathematical Programming*, (K. Schittkowski, Ed.), NATO ASI Series, **15**, Springer-Verlag, Berlin, Germany, 1985.

11. Powell, M. J. D.: A Fast Algorithm for Non-linearly Constrained Optimisation Calculations. *Numerical Analysis Proceedings*, Dundee, 1977.

12. Powell, M. J. D.: A Fast Algorithm for Non-linearly Constrained Optimisation Calculations. *Lecture Notes in Economics and Mathematics*, **630**, Springer-Verlag, Berlin, Germany, pp. 144–157, 1978.

13. Stoer, J.: Principles of Sequential Quadratic Programming Methods in Solving Non-Linear Problems. *Computational Mathematical Programming*, (K. Schittkowski, Ed.), NATO ASI Series, **15**, Springer-Verlag, Berlin, Germany, 1985.

14. Schittkowski, K.: NLPQLP: a FORTRAN Implementation of a Sequential Quadratic Programming Algorithm With Distributed and Non-Monotone Line Search – User's Guide, Version 3.1. Department of Computer Science, University of Bayreuth, Germany, Feb. 2010.

15. Anon.: Programming in VAX FORTRAN, Software Version: V4.0. *VAX/VMS Manual No. AA-D034D-TE*, Digital Equipment Corporation (DEC), Maynard, Mass., Sept. 1984.

16. Vaught, A.: G95 FORTRAN Compiler, Mesa, Ariz., Oct. 2006.