

An Investigation of the Influence of Pseudoinverse
Matrix Calculations on Multibody Dynamics
Simulations by Means of the Udwadia-Kalaba
Formulation

D. de Falco^{*}, E. Pennestrì[†], and L. Vita[†]

^{*}Dipartimento di Ingegneria Spaziale e Aeronautica

II Università di Napoli

Real Casa dell'Annunziata, Via Roma,29 81031 Aversa - Italy

[†] Dipartimento di Ingegneria Meccanica

Università di Roma Tor Vergata

Facoltà Ingegneria, via del Politecnico, 1 00133 Roma, Italy

July 11, 2008

ABSTRACT

The formulation of the dynamic equations of motion proposed by Udwadia-Kalaba is discussed from the point of view of numerical efficiency. Since this formulation requires the computation of a pseudoinverse matrix, it is investigated the influence of the method of pseudoinverse computation on the dynamic simulation of an overconstrained linkage. Finally, it has been proposed a new

dynamic equation which explicitly computes accelerations also in the case of mathematical models with rank deficient Jacobian and mass matrices.

INTRODUCTION

In a recent series of papers *e.g.* [Udwadia Kalaba 1992] and in a textbook [Udwadia and Kalaba 1996], starting from the Gauss Principle of Least Constraint [Gauss 1829, Lanczos 1970, Chetaev 1993], deduced a new equation of dynamics for a system of constrained particles or rigid bodies.

The main features of this formulation are:

- the equations of motion can be reduced to a system of ordinary differential equations (ODE), even when a redundant set of coordinates is used;
- variations of degrees-of-freedom due to the change of topology or other causes are allowed and do not require special effort in computer programming;
- rheonomic and scleronomic constraints are treated in the same way;
- forward and inverse dynamics problems can be solved within the same tool;
- easy computer implementation, provided that a subroutine for computing the pseudoinverse matrix is available.

The above features make the formulation very attractive. In fact, one of the shortcomings associated with the use of a redundant set of coordinates is the integration of a mixed system of differential-algebraic equations (DAE). These systems are different from ordinary differential equations (ODE) and require specialised numerical methods for their solution. The matrices are of high order,

but sparse. The computer programming effort required by redundant coordinates dynamic formulations is relatively low and within the capabilities of the average mechanical engineering student.

A fundamental step of the Udwadia-Kalaba (UK) formulation is the computation of the Moore-Penrose generalized inverse or pseudoinverse matrix.

In [Arabyan and Wu 1998] the advantages of this formulation within the framework of multibody dynamics are numerically investigated and compared the number of floating point operations (flops) required for computing the pseudoinverse with Gram-Schmidt (GS) orthogonalization and Singular Value Decomposition (SVD).

However, the authors of the present paper believe that the comparison of the methods for computing the pseudoinverse only on the basis of flops operations required, although important, is somewhat limited. For this reason we decided to compare different methods by observing other parameters such as the reliability, accuracy of results and speed of computation.

Thus, one of the purposes of this paper is to report about numerical tests, for accuracy and computational efficiency, of different pseudoinverse matrix calculation algorithms with explicit reference to multibody dynamics simulations. These tests do not seem to be readily available in literature.

The paper is organized in three parts. The first and second parts are mainly tutorial. In the first part the main steps of the following methods for computing the pseudoinverse matrix are summarized:

- Singular value decomposition (SVD) [Golub and Van Loan 1996];
- Varga's algorithm [Varga 2001] with QR decomposition computed by means of the Householder factorization [IMSL Math library 1997];
- Greville's algorithm [Udwadia and Kalaba 1999];

- Least squares solution with modified Gram-Schmidt (GS) QR factorization [Golub and Van Loan 1996];
- Least squares solution with Householder QR factorization. [Golub and Van Loan 1996]

In the second part of the paper, the fundamental equation reported in [Udwadia and Kalaba 1996] is concisely deduced for the particular case of mass and Jacobian matrices with full rank. In order to deal also with multibody formalisms with singular mass matrix, it has been proposed a new dynamic equation where these assumptions have been removed. Singular mass matrices follow when the kinematics of the rigid bodies is described by means of redundant number of coordinates (*e.g.* [Wehage and Haug 1982, Nikravesh 1988, Haug 1989]).

The third part describes the test problem and reports the numerical results.

REVIEW OF METHODS FOR COMPUTING THE MOORE-PENROSE PSEUDOINVERSE MA- TRIX

This section, for completeness, summarizes the main properties of the Moore-Penrose pseudoinverse matrix and the main steps of the different algorithms tested during the dynamic simulations.

Definitions

The main properties of the Moore-Penrose pseudoinverse matrix $[A]^+$ of a matrix $[A]$ are:

$$\left([A] [A]^+\right)^T = [A] [A]^+ \quad (1a)$$

$$\left([A]^+ [A]\right)^T = [A]^+ [A] \quad (1b)$$

$$[A] [A]^+ [A] = [A] \quad (1c)$$

$$[A]^+ [A] [A]^+ = [A]^+ \quad (1d)$$

When $[A]$ is a square matrix with full rank, then its pseudoinverse coincides with the inverse.

The Moore-Penrose pseudoinverse matrix is associated with the least squares solution of the linear system of equations

$$[A] \{x\} = \{b\} , \quad (2)$$

where the number m of equations is not equal to the number n of unknowns and $[A]$ does not necessarily has full rank.

If we momentarily assume that $[A]$ has a full rank, the following cases are distinguished:

Overdetermined system of equations ($m > n$)

By requiring that

$$h \equiv \|[A] \{x\} - \{b\}\|_2^2 \quad (3)$$

is a minimum, one obtains

$$[A]^T [A] \{x\} = [A]^T \{b\} . \quad (4)$$

Therefore, the solution of (4) can be stated as

$$\{x\} = [A]^+ \{b\} , \quad (5)$$

where

$$[A]^+ = \left([A]^T [A] \right)^{-1} [A]^T \quad (6)$$

is the *left pseudoinverse matrix*. Obviously this definition is valid only if $\left([A]^T [A] \right)^{-1}$ exists *i.e.* matrix $[A]$ has full column rank.

Undetermined system of equations ($m < n$)

The solution is obtained imposing the minimum of the Euclidean norm

$$g \equiv \|x\|_2^2 , \quad (7)$$

with $\{x\}$ subjected to (2). Thus, introduced the new objective function,

$$g' \equiv g + \{\lambda\}^T ([A] \{x\} - \{b\}) \quad (8)$$

the solution is achieved solving the system

$$\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix} \begin{Bmatrix} x \\ \lambda \end{Bmatrix} = \begin{Bmatrix} 0 \\ b \end{Bmatrix} , \quad (9)$$

or

$$\{x\} = [A]^+ \{b\} , \quad (10)$$

where

$$[A]^+ = [A]^T \left([A] [A]^T \right)^{-1} , \quad (11)$$

is the *right pseudoinverse matrix*. This definition is valid only if $\left([A] [A]^T \right)^{-1}$ exists, *i.e.* matrix $[A]$ has full row rank.

In the more general case of $[A]$ of rank r , the matrix can be partitioned as follows

$$\begin{aligned}
[A] &= [Q] \begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix} [P] \\
&= \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \end{bmatrix} \\
&= [Q_1] [L] [P_1]^T
\end{aligned} \tag{12}$$

where $[Q]_{m \times m}$ and $[P]_{n \times n}$ are orthogonal matrices and $[L]_{r \times r}$ is a lower triangular matrix of rank r .

By substitution into (1) one can verify that

$$[A]^+ = [P] [B]^+ [Q]^T \tag{13}$$

where

$$[B]^+ = \begin{bmatrix} L^{-1} & 0 \\ 0 & 0 \end{bmatrix}. \tag{14}$$

From the numerical point of view, one should avoid the use of (6) or (11) also in the case of full rank of $[A]$. In fact, the matrices $[A]^T [A]$ and $[A] [A]^T$ could be ill-conditioned and their direct inversion prone to round-off errors. The evaluation of pseudoinverses through matrix decomposition is considered numerically more robust.

The *least squares* method

Since there is abundance of software procedures for computing the least squares solution of a system of algebraic equations

$$[A] \{x\} = \{b\} , \quad (15)$$

the computation of the pseudoinverse can be reduced to such solution. These procedures are often based on QR decomposition by means of Householder reflections or GS orthogonalization.

Let

$$\begin{aligned} \{b_1\} &= \left\{ \begin{array}{cccc} 1 & 0 & 0 & \dots & 0 \end{array} \right\}^T , \\ \{b_2\} &= \left\{ \begin{array}{cccc} 0 & 1 & 0 & \dots & 0 \end{array} \right\}^T , \\ \{b_3\} &= \left\{ \begin{array}{cccc} 0 & 0 & 1 & \dots & 0 \end{array} \right\}^T , \\ &\dots\dots\dots \\ \{b_m\} &= \left\{ \begin{array}{cccc} 0 & 0 & 0 & \dots & 1 \end{array} \right\}^T . \end{aligned}$$

The procedure differs according to the dimensions of $[A]$.

Case $m > n$

1. Solve m times the following system

$$[A]^T [A] \{x\} = [A]^T \{b\} . \quad (16)$$

Actually the system is solved only once. The factor matrices used for the first solution are saved and used also for the remaining ones.

2. From the m solutions $\{x_1\}, \{x_2\}, \dots, \{x_m\}$ one can form the pseudoinverse matrix

$$[A]^+ = \begin{bmatrix} \{x_1\} & \{x_2\} & \{x_3\} & \dots & \{x_m\} \end{bmatrix} \quad (17)$$

Case $n \geq m$

1. Solve m times the following system

$$\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix} \begin{Bmatrix} x \\ \lambda \end{Bmatrix} = \begin{Bmatrix} 0 \\ b \end{Bmatrix} \quad (18)$$

Also in this case the pseudoinverse is given by equation (17).

The square matrices in (16) and (18) are singular or ill conditioned. Thus their solution requires special care. As mentioned, an appropriate way to solve least squares problems is by means of GS orthogonalization or Householder QR factorization.

Singular Value Decomposition method

1. Apply the Singular Value Decomposition (SVD) to matrix $[A]$

$$[A] = [U][\Sigma][V]^T,$$

where $[U][U]^T = [I]_{m \times m}$, $[V][V]^T = [I]_{n \times n}$

$$[\Sigma]_{m \times n} = \begin{bmatrix} \sigma & 0 \\ 0 & 0 \end{bmatrix}$$

where

$$[\sigma]_{r \times r} = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_r \end{bmatrix}$$

is a diagonal matrix.

2. Compute

$$[A]^+ = [V][\Sigma]^+[U]^T, \quad (19)$$

where

$$[\Sigma]^+ = \begin{bmatrix} \sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

Varga's method

1. Apply the QR factorization to matrix $[A]$. Thus, one obtains:

$$[A] = [U][G] = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} G_1 \\ 0 \end{bmatrix}$$

with $[G_1]$ matrix with full row rank and $[U]^T[U] = [I_m]$.

2. Apply the QR factorization to $[G_1]^T$

$$[G_1]^T = [V]^T \begin{bmatrix} G_2^T \\ 0 \end{bmatrix}, \quad (20)$$

with $[G_2]^T$ invertible matrix and $[V][V]^T = [I_n]$.

3. Since

$$[A] = [U] \begin{bmatrix} G_2 & 0 \\ 0 & 0 \end{bmatrix} [V],$$

the pseudoinverse matrix is readily obtained from

$$[A]^+ = [V]^T \begin{bmatrix} G_2^{-1} & 0 \\ 0 & 0 \end{bmatrix} [U]^T . \quad (21)$$

Greville's method

1. Decompose the matrix $[A]_{m \times n}$ into row vectors $\{a_i\}$, ($i = 1, 2, \dots, m$)

$$[A] = \begin{bmatrix} a_1^T & a_2^T & \dots & a_m^T \end{bmatrix}$$

2. Let matrix

$$[A_i]_{i \times n} = \begin{bmatrix} A_{i-1} \\ a_i \end{bmatrix} \quad (22)$$

with $[A_1]_{1 \times n} = \{a_1\}_{1 \times n}$.

3. For $i = 2, \dots, m$ compute the matrices $[A_i]^+$ as

$$[A_i]_{n \times i}^+ = \left[[A_{i-1}]^+ - \{b_i\}^T \{d_i\} \right] \{b_i\}^T ,$$

where

$$\begin{aligned} \{d_i\}_{1 \times (i-1)} &= \{a_i\} [A_{i-1}]^+ \\ \{c_i\}_{1 \times n} &= \{a_i\} - \{d_i\} [A_{i-1}] \\ \{b_i\}_{1 \times n} &= \begin{cases} \frac{\{c_i\}}{\{c_i\}\{c_i\}^T} & (\|c_i\| \neq 0) \\ \frac{\{d_i\}[A_{i-1}]^+}{1 + \{d_i\}\{d_i\}^T} & (\|c_i\| = 0) \end{cases} \\ [A_1]^+ &= \begin{cases} \frac{\{a_1\}^T}{\{a_1\}\{a_1\}^T} & (\|a_1\| \neq 0) \\ \{a_1\}^T & (\|a_1\| = 0) \end{cases} \end{aligned}$$

4. After m repetitions $[A_m]^+$ gives the pseudoinverse $[A]_{n \times m}^+$ of matrix $[A]$.

THE FORMULATION OF UDWADIA-KALABA AND SOME ALTERNATIVES

In this section the formulation of [Udwadia and Kalaba 1996] is initially deduced following the approach of [Arabyan and Wu 1998]. Then, such formulation is generalized for the case of rank deficient Jacobian and mass matrices.

Let us denote with

- $\{F\}$ the vector of external generalized forces;
- $[M]$ the mass matrix;
- m , number of constraints on the acceleration vector;
- n , number of generalized coordinates;
- $[\Psi_q]$ the Jacobian of constraints equations;
- $\{q\}$ the vector of generalized coordinates;
- $\{\gamma\} = -([\Psi_q]\{\dot{q}\})_q\{\dot{q}\} - 2[\Psi_{qt}]\{\dot{q}\} - \{\Psi_{tt}\}$;
- $\{\lambda\}$ Lagrange's multiplier vector
- the upperscript $+$ the operation of pseudoinverse of a matrix.

When in a multibody dynamics model a redundant set of coordinates is used, the following system of differential-algebraic system of equations (DAE) is obtained [Nikravesh 1988, Haug 1989]

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix} \begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} F \\ \gamma \end{Bmatrix} \quad (23)$$

Assuming a full rank for the mass and the Jacobian matrices, the square block matrix at the left-hand of (23) can be inverted by partitioning [Stephenson 1965]

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix}^{-1} = \begin{bmatrix} M^{-1} - M^{-1}\Psi_q^T (\Psi_q M^{-1}\Psi_q^T)^{-1} \Psi_q M^{-1} & M^{-1}\Psi_q^T (\Psi_q M^{-1}\Psi_q^T)^{-1} \\ (\Psi_q M^{-1}\Psi_q^T)^{-1} \Psi_q M^{-1} & - (\Psi_q M^{-1}\Psi_q^T)^{-1} \end{bmatrix}. \quad (24)$$

Introduced the vector

$$\{\ddot{q}_f\} = [M]^{-1} \{F\}, \quad (25)$$

representing the acceleration vector of the unconstrained system, the solution of (23) follows

$$\{\ddot{q}\} = \{\ddot{q}_f\} + [M]^{-1} [\Psi_q]^T \left([\Psi_q] [M]^{-1} [\Psi_q]^T \right)^{-1} (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}) \quad (26)$$

and

$$\{\lambda\} = \left([\Psi_q] [M]^{-1} [\Psi_q]^T \right)^{-1} (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}). \quad (27)$$

If we let

$$[M]^{-1} = [M]^{-\frac{1}{2}} [M]^{-\frac{1}{2}}, \quad (28a)$$

$$[D] = [\Psi_q] [M]^{-\frac{1}{2}}, \quad (28b)$$

then equation (26) can be rewritten in the form

$$\begin{aligned} \{\ddot{q}\} &= \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} \left([M]^{-\frac{1}{2}} [\Psi_q]^T \right) \left([\Psi_q] [M]^{-\frac{1}{2}} [M]^{-\frac{1}{2}} [\Psi_q]^T \right)^{-1} (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}), \\ &= \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} [D]^T \left([D] [D]^T \right)^{-1} (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}) \end{aligned} \quad (29)$$

Taken into account the definition of right pseudoinverse matrix formerly stated,

after we let

$$[D]^+ = [D]^T \left([D] [D]^T \right)^{-1} , \quad (30)$$

the previous equation can be concisely expressed as follows [Arabyan and Wu 1998]:

$$\{\ddot{q}\} = \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} [D]^+ (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}) . \quad (31)$$

This is the dynamic formulation originally proposed by [Udwadia and Kalaba 1996] starting from Gauss Principle of Least Constraint.

If the Baumgarte stabilization is introduced, then (31) is modified as follows

$$\{\ddot{q}\} = \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} [D]^+ \left(\{\gamma\} - 2\alpha \{\dot{\Psi}\} - \beta^2 \{\Psi\} - [\Psi_q] \{\ddot{q}_f\} \right) , \quad (32)$$

where α and β are the gain parameters often chosen such that $\alpha = \beta$.

It must be pointed out that (31) is valid also when $[\Psi_q]$ does not have a full rank.

There is an apparent inconsistency in our reasoning. We started with expression (24), valid only with full rank mass and Jacobian matrices, and we ended up with the equation of motion (31) which does not demand a full rank Jacobian.

To remove this inconsistency, let us assume a full rank mass matrix. Hence, the pseudoinverse of the matrix of coefficients in equation (23) can be expressed in the form [Campbell 1979]

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix}^+ = \begin{bmatrix} M^{-1} - M^{-1} \Psi_q^T Q^+ \Psi_q M^{-1} & M^{-1} \Psi_q^T Q^+ \\ Q^+ \Psi_q M^{-1} & -Q^+ \end{bmatrix} . \quad (33)$$

where

$$[Q] = [\Psi_q] [M]^{-1} [\Psi_q]^T . \quad (34)$$

Hence, from the least squares solution of (23) we obtain

$$\{\ddot{q}\} = ([I] - [H] [\Psi_q]) [M]^{-1} \{F\} + [H] \{\gamma\} \quad (35)$$

where

$$[H] = [M]^{-1} [\Psi_q]^T [Q]^+ . \quad (36)$$

Introducing the equalities (25) and (28) into (35) one readily obtains (31).

Multibody dynamics formulations with a singular mass matrix are not rare. For instance, the well known formulation discussed in [Haug 1989] and implemented in the DADS multibody dynamics code results into a singular mass matrix. This is due to the redundancy of generalized coordinates used for the definition of each rigid body position. In this case the (31) cannot be applied.

In the following we propose a new general equation of dynamics, which maintains its validity also in the case of rank deficient Jacobian and mass matrices.

After we let

$$[E] = [I] - [\Psi_q]^+ [\Psi_q] , \quad (37)$$

$$[Q] = ([E] [M] [E])^+ , \quad (38)$$

$$[R] = [M] [\Psi_q]^+ , \quad (39)$$

the Moore-Penrose pseudoinverse of the matrix coefficient in equation (23), assuming a symmetric mass matrix, can be expressed in the form [Campbell 1979]

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix}^+ = \begin{bmatrix} 0 & \Psi_q^+ \\ \left([\Psi_q]^+\right)^T & -\left([\Psi_q]^+\right)^T [R] \end{bmatrix} + \begin{bmatrix} I \\ -R^T \end{bmatrix} [Q] \begin{bmatrix} I & -R \end{bmatrix} . \quad (40)$$

Hence, the acceleration vector is given by the expression

$$\{\ddot{q}\} = [Q] \{F\} + \left([\Psi_q]^+ - [Q] [M] [\Psi_q]^+ \right) \{\gamma\} . \quad (41)$$

An explicit form of the equations of motion for constrained mechanical systems, apt for mathematical models with singular mass matrices, is also presented in [Udwadia and Phohomsiri 2006].

If we denote with $\{C\}$ the vector describing the nature of non-ideal constraints, and define

$$[\overline{M}] = \begin{bmatrix} (I - \Psi_q^+ \Psi_q) M \\ \Psi_q \end{bmatrix} \quad (42)$$

then, according to [Udwadia and Phohomsiri 2006],

$$\{\ddot{q}\} = [\overline{M}]^+ \begin{Bmatrix} F + C \\ \gamma \end{Bmatrix} + \left([I] - [\overline{M}]^+ [\overline{M}] \right) \{\eta\} \quad (43)$$

with $\{\eta\}$ an arbitrary n vector.

It is interesting a comparison of the number of floating point operations (flops) required by equations (31), (35) (41) and (43).

The equations (31) and (35) can be applied with rank deficient Jacobian, but mass matrix positive definite, whereas equations (41) and (43) are both applicable with rank deficient Jacobian and mass matrices. It should be acknowledged that flops counting, since ignores other overheads associated with program execution, is not an accurate approach for measuring computational efficiency. However, in the field of dynamics is often adopted.

In our estimate we made the following assumptions:

1. The product of two real matrices $[A]_{m \times n}$ and $[B]_{n \times p}$ requires $2mnp$ flops.
2. The evaluation of the pseudoinverse of a matrix is equivalent, in term of

flops, to the computation of the Singular Value Decomposition of the same matrix. In fact many mathematical libraries (*e.g* IMSL, LAPACK) compute the pseudoinverse by means of the SVD algorithm. The application of the Golub-Reisch SVD algorithm to a matrix of m rows and n columns requires $4m^2n + 8mn^2 + 9n^3$ flops [Golub and Van Loan 1996].

3. The mass matrices in (31) and (35) are diagonal, hence their inversion requires only n flops. In case this assumption is not verified, one can substitute $an^3 + O(n^2)$, where a depends on the algorithm chosen.

The Table 1 summarizes the flops counting for the different dynamic formulations herein discussed. The details of the flops counting for each equation are reported in the Tables 2, 3, 4 and 5.

The Table 6 offers a direct comparison for given numerical values of m and n .

THE PROBLEM OF RANK COMPUTATION

A serious numerical problem that arises in the computation of the pseudoinverse is the accurate computation of the matrix rank. At the practical level, the user must establish a numerical treshold under which the numerical values are all considered to be zero. One may set this value equal to the hardware precision. However, since the inverse of these very small values is required by the algorithms, this inflates a perturbation in the numerical solution. Thus, a truncation or other means of regularization are compulsory to obtain acceptable results.

In all the examples discussed in the paper the numerical treshold necessary for the computation of the pseudoinverse is set equal to the error tolerance parameter TOL required by the following IMSL numerical integration subroutines

- DIVPAG, which solve an ODE initial-value problem using the Adams-Moulton or Gear's method.
- DIVPRK, which solve an ODE initial-value problem using the Runge-Kutta-Verner fifth and sixth order method.

During the integration process, both subroutines try to control the norm of the local error such that the global error is proportional to TOL.

It should be acknowledged that the adoption of different criterion may alter the results herein presented.

SUMMARY OF FORTRAN SUBROUTINES USED

Whenever possible the computation of the pseudoinverse has been executed by means of standard math libraries, such as IMSL. For instance, the Householder QR factorization has been carried out through the combined use of IMSL library subroutines DLQRRR and DLQERR. For other relevant matrix computations, the Fortran code available on the web site www.netlib.org, such as LAPACK, has been used.

All the Fortran subroutines described in this section and implemented by the authors are available upon request.

- **Pseudoinverse by means of the SVD method** The IMSL library [IMSL Math library 1997] supplies the ready-to-use DLSGRR subroutine.
- **Pseudoinverse by means of the Varga's method** The two Householder QR factorizations required are executed by means of the combined use of the IMSL library subroutines DLQRRR and DLQERR. For the inversion of the triangular matrix G_2 the IMSL routine DLINRT is used. Computation of transpose and matrix multiplications are done with DTRNRR and DMRRRR, respectively.

- **Pseudoinverse by means of Greville's method** A Fortran subroutine has been implemented with matrix operations executed with LAPACK subroutines [Anderson et al. 1995].
- **Pseudoinverse by means of Least squares solution and Householder QR factorization** For the least squares solution of a linear system of equations, it has been used the IMSL subroutines DLQRRR and DLQRSL. DLQRRR provides information on the QR factorization, whereas DLQRSL compute the least squares solution.
- **Pseudoinverse by means of Least squares solution and modified Gram-Schmidt QR factorization** For the least squares solution of a linear system of equations, it has been used the code written by [Wampler 1979].

SIMULATION RESULTS

The example chosen to test the different algorithms of pseudoinverse computation is the overconstrained mechanism shown in Figure 1. The length, mass and moment of inertia of each of the three parallel links are $L=1$, $m=1$ and $I=0.1$, respectively. The length, mass and moment of inertia of the coupler are $L_2=2$, $m_2=2$, $I_2=0.2$. The kinematic constraints were modelled by means of the method described in [Nikravesh 1988]. The position of each body is described by three coordinates, thus there are $n=12$ coordinates and $k=12$ algebraic constraints due to the presence of six revolute pairs. Due to the particular link dimension, Jacobian rank is $r = 11$. Thus the degree-of-freedom is $F = n - r = 1$. Gravity force is included and dissipative effects neglected.

The dynamic formulation adopted for all the numerical tests is the one described by equation (32).

The gain parameters $\alpha = 1000$ and $\beta = 100$ for the Baumgarte's stabilisation have been empirically chosen in order to bound the violation of constraints.

The results were monitored for $t_f=20$ s of simulation time. Sometimes the execution was halted because of precision loss or other causes.

The results of the numerical tests are summarized in Tables 7-12.

The simulations were carried out with a PC equipped with AMD Athlon 2600+, 512 MB RAM, running under Microsoft XP Professional. The CPU time reported is the average of the CPU times measured executing four times the same simulation.

CONCLUSIONS

Computing pseudoinverse by means of the Gram-Schmidt decomposition is usually less expensive than other methods based on the use of the SVD decompositions, Greville algorithm, or Householder reflections. However, these last methods require about twice as much arithmetic, but are more reliable and accurate near the limits of residual reduction. The loss of accuracy usually connected with the Gram-Schmidt method of computing the pseudoinverse requires the integration subroutine a considerable higher number of computational steps in order to maintain the error lower than a fixed limit. The other two methods discussed are very accurate and require less function evaluation calls during numerical integration.

Finally, making use of generalized inverses of partitioned matrices, new explicit equations of motion of constrained multibody systems have been deduced. The first one is valid only in the case of rank deficient Jacobian. The second one can be applied in mathematical models where both mass and Jacobian matrices are rank deficient. The number of flops required by the equations presented in [Udwadia and Kalaba 1996] [Udwadia and Phohomsiri 2006] has been com-

pared with the one required by the equations herein deduced. In general, we observe that the dynamic formulations which can handle models with mass and Jacobian matrices require a higher number of flops.

References

- [Anderson et al. 1995] Anderson, E. *et al.* (1995). *LAPACK - Users's Guide*. Society for Industrial and Applied Mathematics, Philadelphia.
- [Arabyan and Wu 1998] Ara Arabyan, Fei Wu (1998). An Improved Formulation for Constrained Mechanical Systems, *Multibody System Dynamics*, 2(1), 49-69.
- [Campbell 1979] Campbell, S.L. and Meyer Jr., C.D. (1979). *Generalized Inverses of Linear Transformations*, Dover Edition Inc., 69.
- [Chetaev 1993] Chetaev, N. (1993). *Mécanique Rationnelle*, Éditions Mir, Moscou, 251-253.
- [Gauss 1829] Gauss, C.F. (1829). Über ein neues allgemeines Grundgesetz der Mechanik, *Journal für Reine und Angewandte Mathematik*, 4, 232-235.
- [Golub and Van Loan 1996] Golub, G.H., Van Loan, C.F. (1996). *Matrix Computations*, 3rd Edition, The John Hopkins University Press.
- [Haug 1989] Haug, E., *Computer-Aided Kinematics and Dynamics of Mechanical Systems*, Allyn and Bacon (1989).
- [IMSL Math library 1997] IMSL Math library (1997), vols.I, II.
- [Lanczos 1970] Lanczos, C. (1970). *The Variational Principle of Mechanics*, Dover Publications, Inc., Fourth Edition, 106-110.

- [Nikravesh 1988] Nikravesh, P. (1988). *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall.
- [Stephenson 1965] Stephenson, G. (1965). *An Introduction to Matrices Sets and Groups for Science Students*, Dover Publications Inc., 48-51.
- [Udwadia Kalaba 1992] Udwadia, F.E., Kalaba, R.E. (1992). A new perspective on constrained motion, *Proceedings of the Royal Society of London, Series A*, 439, November, 407-410.
- [Udwadia and Kalaba 1996] Udwadia, F. and Kalaba, R. (1996). *Analytical Dynamics a New Approach*. Cambridge University Press, Cambridge.
- [Udwadia and Kalaba 1999] Udwadia, F.E., Kalaba, R.E. (1999). A Unified Approach for the Recursive Determination of Generalized Inverses, *Computers and Mathematics with Applications*, 37, 125-130.
- [Udwadia and Phohomsiri 2006] Udwadia, F.E., Phohomsiri, P. (2006). Explicit equations of motion for constrained mechanical systems with singular mass matrices and applications to multi-body dynamics, *Proceedings of the Royal Society, Series A*, 462, 2097-2117.
- [Wampler 1979] Wampler, R.H. (1979). Modified Gram-Schmidt with iterative refinement, *ACM TOMS* 5, 494-499. Fortran code available at www.netlib.org
- [Varga 2001] Varga, A. (2001). On computing generalized inverse systems using matrix pencil methods, *International Journal of Applied Mathematics and Computer Science*, 11(5), 1055-1068.
- [Wehage and Haug 1982] Wehage, R. and Haug, E. (1982). Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems. *ASME Journal of Mechanical Design*, 134(1), 247-255.

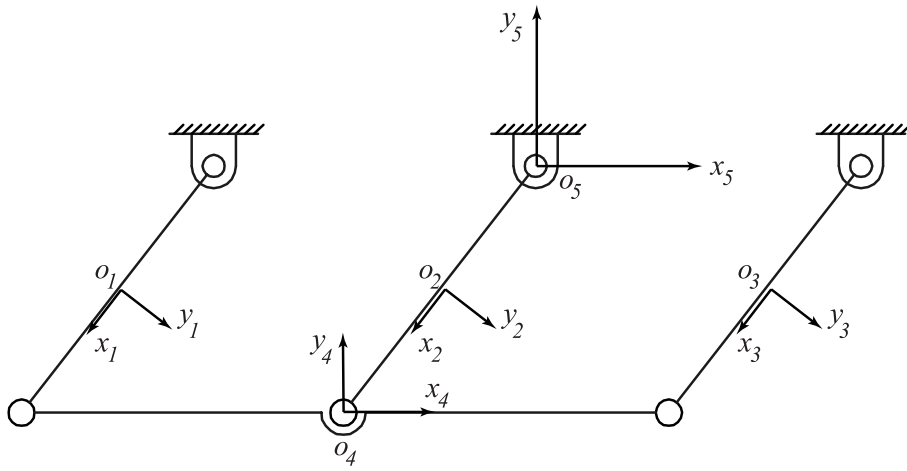


Figure 1: Five bar parallelogram linkage

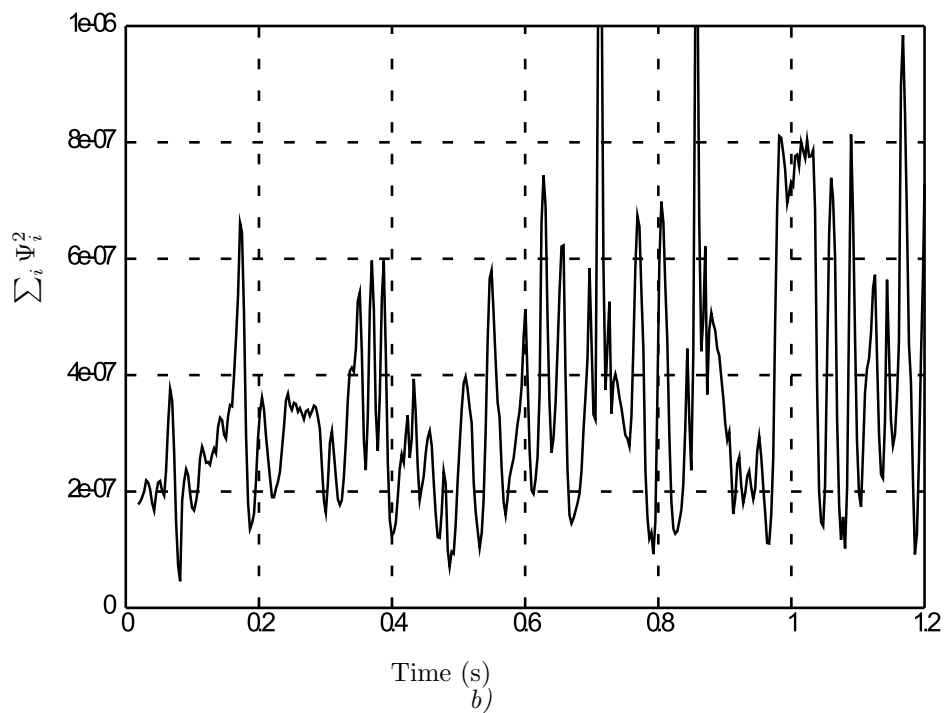
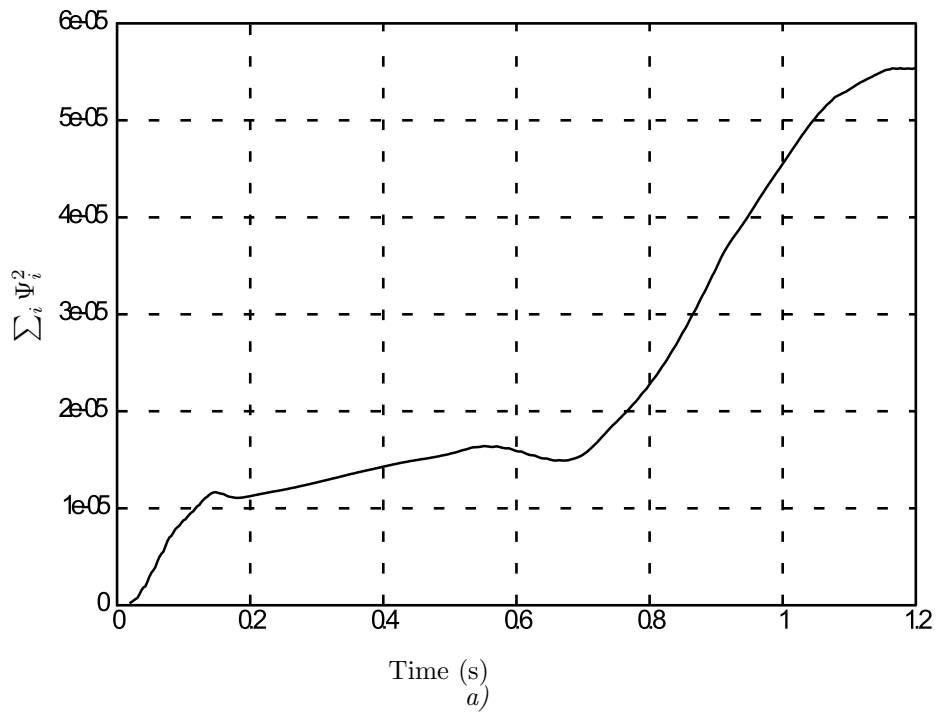


Figure 2: Violation of position constraints: a) Without stabilization b) With Baumgarte stabilization

Table 1: Comparison of flops for equations of motion of constrained multibody systems

Dynamic formulation	Number of flops
[Udwadia and Kalaba 1996], eq. (31)	$12mn^2 + 4m^2n + 9n^3 + 6n^2 + 4n$
[Udwadia and Phohomsiri 2006], eq. (43)	$4m^2n + 10mn^2 + 11n^3 + 4(n+m)^2n$ $+ (n+m)(8n^2 + 2n) + n$
(This investigation), eq. (35)	$8n^2m + 21m^3 + 2m^2n + 6n^2 + 3n$
(This investigation), eq. (41)	$4m^2n + 12mn^2 + 34n^3 + 3n + 4n^2$

Table 2: Counting of floating point operations for the equation (31) deduced in [Udwadia and Kalaba 1996]

Matrix operation	Flops
Computation of $[M]^{-1}$	n
Computation of $[M]^{-\frac{1}{2}}$	n
$[D] = [\Psi_q] [M]^{-\frac{1}{2}}$	$2n^2m$
Computation of $[D]^+$	$4m^2n + 8mn^2 + 9n^3$
$\{\ddot{q}_f\} = [M]^{-1} \{f\}$	$2n^2$
$[\Psi_q] \{q_f\}$	$2n^2$
$\{\gamma\} - [\Psi_q] \{q_f\}$	n
$[M]^{-\frac{1}{2}} [D]^+$	$2n^2m$
$[M]^{-\frac{1}{2}} [D]^+ (\{\gamma\} - [\Psi_q] \{q_f\})$	$2n^2$
$\{\ddot{q}\} = \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} [D]^+ (\{\gamma\} - [\Psi_q] \{q_f\})$	n

Table 3: Counting of floating point operations for the equation (35) (This investigation)

Matrix operation	Flops
$[M]^{-1}$	n
$[M]^{-1} [\Psi_q]^T$	$2n^2m$
$[Q] = [\Psi_q] [M]^{-1} [\Psi_q]^T$	$2n^2m$
Compute $[Q]^+$	$21m^3$
$[\Psi_q]^T [Q]$	$2m^2n$
$[H] = [M]^{-1} [\Psi_q]^T [Q]$	$2n^2m$
$[H] [\Psi_q]$	$2n^2m$
$[I] - [H] [\Psi_q]$	n
$[M]^{-1} f$	$2n^2$
$([I] - [H] [\Psi_q]) [M]^{-1} \{F\}$	$2n^2$
$[H] \{\gamma\}$	$2n^2$
$\{\tilde{q}\} = ([I] - [H] [\Psi_q]) [M]^{-1} \{F\} + [H] \{\gamma\}$	n

Table 4: Counting of floating point operations for the equation (41) (This investigation)

Matrix operation	Flops
Computation of $[\Psi_q]^+$	$4m^2n + 8mn^2 + 9n^3$
$[\Psi_q]_{n \times m}^+ [\Psi_q]_{m \times n}$	$2n^2m$
$[E]_{n \times n} = [I]_{n \times n} - [\Psi_q]^+ [\Psi_q]$	n
$[Q]_{n \times n} = [E] [M] [E]$	$4n^3$
Compute $[Q]^+$	$21n^3$
$[I] + [M]$	n
$([I] + [M]_{n \times n}) [\Psi_q]_{n \times m}^+$	$2n^2m$
$[Q] \{F\}$	$2n^2$
$([I + M]) [\Psi_q]^+$	$2n^2$
$\{\ddot{q}\} = ([I] - [H] [\Psi_q]) [M]^{-1} \{F\} + [H] \{\gamma\}$	n

Table 5: Counting of floating point operations for the equation (43) deduced in [Udwadia and Phohomsiri 2006]

Matrix operation	Flops
Computation of $[\Psi_q]^+$	$4m^2n + 8mn^2 + 9n^3$
$[\Psi_q]_{n \times m}^+ [\Psi_q]_{m \times n}$	$2mn^2$
$[R] = [I] - [\Psi_q]^+ [\Psi_q]$	n
$[R]_{n \times n} [M]_{n \times n}$	$2n^3$
Computation of $[\bar{M}]_{n \times (n+m)}^+$	$4(n+m)^2n + 8(n+m)n^2 + 9n^3$
$[\bar{M}]_{n \times (n+m)}^+ \begin{Bmatrix} Q \\ b \end{Bmatrix}_{(n+m) \times 1}$	$2(n+m)n$

Table 6: Number of flops required for the computation of the accelerations vector

Dynamic formulation	$m = 2$ $n = 3$	$m = 15$ $n = 18$	$m = 19$ $n = 28$
[Udwadia and Kalaba 1996], eq. (31)	573	129024	421568
[Udwadia and Phohomsiri 2006] , eq. (43)	1218	294102	975716
This investigation, eq. (35)	399	119853	288211
This investigation, eq. (41)	1227	274158	968772

Table 7: TOL= 10^{-10} . Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. No Baumgarte stabilization applied.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Fails ¹	$3.7 \cdot 10^{-7}$	-	-	-
Varga	Fails ²	$9.9 \cdot 10^{-2}$	-	-	-
Greville	Correct up to 17.5 s	$4.5 \cdot 10^{-5}$	2,961,955	2,633,811	0.93
LS-Householder	Correct up to 16.5 s	$2.3 \cdot 10^{-4}$	6,131,159	4,177,395	11.57
LS-Modified GS	Correct up to 17.0 s	$1.1 \cdot 10^{-4}$	3,749,766	3,015,354	36.85

¹ After 9.1 s of correct simulation, the integration subroutine is halted because of repeated error-test failures (see [IMSL Math library 1997], p. 668).

² After 8.0 s of correct simulation, the integration subroutine is halted after failing to pass the error-test failures (see [IMSL Math library 1997], p. 668).

Table 8: TOL= 10^{-8} . Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. No Baumgarte stabilization applied.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Fails ¹	$3.4 \cdot 10^{-6}$	-	-	-
Varga	Fails ²	$9.9 \cdot 10^{-2}$	-	-	-
Greville	Correct up to 17.5 s	$1.86 \cdot 10^{-4}$	1,648,055	1,422,689	0.53
LS-Householder	Correct up to 18.0 s	$7.9 \cdot 10^{-4}$	1,641,751	1,422,297	1.28
LS-Modified GS	Correct up to 18.0 s	$5.2 \cdot 10^{-4}$	1,644,341	1,424,864	6.03

¹ After 12.07 s of correct simulation, the integration subroutine is halted because a fatal error 1 from DIVPAG (see [IMSL Math library 1997], p. 668).

² After 8.06 s of correct simulation, the integration subroutine is halted because the precision loss of constraints equations.

Table 9: TOL= 10^{-6} . Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. No Baumgarte stabilization applied.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Correct up to 17.5 s	$4.1 \cdot 10^{-4}$	1,033,381	847,294	0.28
Varga	Fails ¹	$9.9 \cdot 10^{-2}$	-	-	-
Greville	Correct up to 19.0 s	$9.7 \cdot 10^{-4}$	1,017,558	1,422,689	0.37
LS-Householder	Correct	$7.9 \cdot 10^{-4}$	1,641,751	1,422,297	0.76
LS-Modified GS	Correct	$6.9 \cdot 10^{-4}$	1,013,795	839,020	3.40

¹ After 7.3 s of correct simulation, the integration subroutine is halted because the precision loss of constraints equations.

Table 10: TOL= 10^{-10} . Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. Baumgarte stabilization applied, $\alpha = 1000$, $\beta = 100$.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Fails ¹	-	-	-	-
Varga	Correct	$4.5 \cdot 10^{-5}$	78,491,398	47,074,599	11.4
Greville	Correct	$2.1 \cdot 10^{-7}$	74,043,990	44,053,874	18.3
LS-Householder	Correct	$3.3 \cdot 10^{-7}$	73,707,588	43,948,609	51.5
LS-Modified GS	Correct s	$4.1 \cdot 10^{-7}$	73,910,268	44,515,071	241.0

¹ After 1.17 s of correct simulation, the integration subroutine is halted because a fatal error 1 from DIVPAG (see [IMSL Math library 1997], p. 668).

Table 11: $\text{TOL}=10^{-8}$. Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. Baumgarte stabilization applied, $\alpha = 1000$, $\beta = 100$.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Fails ¹	-	-	-	-
Varga	Fails ²	-	-	-	-
Greville	Correct	$4.90 \cdot 10^{-7}$	80,109,155	55,569,995	19.4
LS-Householder	Correct	$3.5 \cdot 10^{-7}$	80,197,458	54,468,468	55.0
LS-Modified GS	Correct	$4.9 \cdot 10^{-7}$	80,112,706	54,604,452	290

¹ After 0.17 s of correct simulation, the integration subroutine is halted because a fatal error 1 from DIVPAG (see [IMSL Math library 1997], p. 668).

² After 14.97 s of correct simulation, the integration subroutine is halted because the precision loss of constraints equations.

Table 12: $\text{TOL}=10^{-6}$. Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. Baumgarte stabilization applied, $\alpha = 1000$, $\beta = 100$.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Fails ¹	-	-	-	-
Varga	Fails ²	-	-	-	-
Greville	Correct	$1.9 \cdot 10^{-6}$	80,483,163	57,743,095	19.5
LS-Householder	Correct	$1.14 \cdot 10^{-6}$	80,689,071	58,099,798	55
LS-Modified GS	Correct	$1.06 \cdot 10^{-6}$	80,636,983	58,010,479	291

¹ After 0.46 s of correct simulation, the integration subroutine is halted because a fatal error 1 from DIVPAG (see [IMSL Math library 1997], p. 668).

² After 9.7 s of correct simulation, the integration subroutine is halted because the precision loss of constraints equations.