LAGRANGIAN MULTIPLIER / MATRIX PARTITIONING APPROACH FOR THE SOLUTION OF THE MUSCLE LOAD DISTRIBUTION PROBLEM

James Shippen[1] and Barbara May[1]

Department of Industrial Design, Coventry School of Art and Design, Coventry University, United Kingdom[1]

The paper presents a musculoskeletal biomechanical model which resides in the Simulink / MATLAB environment and is suitable for the analysis of sporting activities. The model can be used to calculate ground reaction forces, muscles, joint and skeletal component loadings. The model is currently being used at over 150 research institutions and because of its short learning curve is especially suitable for undergraduate teaching. The calculation of the muscle loading distribution is calculated by minimising an objective function. The optimisation approach implemented uses a Lagrangian multiplier technique and is supplemented by a matrix partitioning method to apply inequality constraints. This technique is capable of calculating the muscle force distribution in near real time using the limited resources of a laptop computer.

KEY WORDS: Matlab, Simulink, musculoskeletal modelling, real time

INTRODUCTION: The modelling of the musculoskeletal system is becoming a widely used technique in sports biomechanical simulation. Most models consist of rigid bodies with appropriate mass / inertia and geometric properties to represent the major skeletal components. These segments are connected by joints which represent the kinematics of their anatomical counterparts [Delp et al 2007]. Muscles are included in the model enabling the muscle forces and joint contact forces corresponding to a prescribed movement and external forces to be calculated (inverse dynamics). The inverse dynamics solution method commences with a calculation of the joint torques which correspond to the observed motion [Koopman et al 1995]. Equating the torques at the joints to the torques generated by the muscles results in a number of equality constraints; the number of equality constraints equals the number of joint torques considered in the model. However most musculoskeletal models possess many more muscles than joint torques; for example the BoB model [Shippen and May 2010] has 606 locomotor muscles but only 30 joint torques to satisfy. Hence the system contains many redundancies and consequently there is not a unique solution for the muscle loading problem by only satisfying the equality constraints. Therefore it is necessary to introduce an objective function to choose the optimal solution from the infinite number of possible solutions for the muscle loading distribution. This objective function should be based on a physiological basis, for example, minimising fatigue. Numerous objective functions have been proposed and implemented [Crowninshield and Brand, 1981, Thelen et al., 2003] and Modenese et al (2011) found that an objective function based on the minimisation of the sum of the quadratic of the muscles' activation provided the best fit of the calculated muscle activity to the measurements of muscle activity using EMG methods.

Additionally, inequality constraints arise as muscles cannot push and hence the instantaneous force must be greater than zero. Also, the maximum force which a muscle can generate is limited and hence the instantaneous force must be less than this value which introduces further inequality constraints.

Minimising the objective function subject to equality and inequality constraints can be solved by various numeric approaches but this paper presents a novel, computationally efficient method suitable for solving the muscle load distribution in a full body musculoskeletal system in real time on limited computing facilities. This enables the production of a real-time biomechanical system providing feedback to a subject on the activation of muscles and loads occurring in the muscles and joints.

METHOD: The loads in the body's muscles will be calculated as the distribution which minimises an objective function whilst being subject to equality and inequality constraints. A Lagrange multiplier method approach will be used [Arfken 1985] to minimise the objective function subject to equality constraints together with an iterative matrix partitioning approach to accommodate the inequality constraints. The objective function to be minimised, f(x), is defined as the sum of the squares of the muscles' activations where muscle activation is defined as the instantaneous force divided by the maximum isometric force of the muscle:

$$f(x) = \sum_j x_i^2$$

where x_i = muscle activation

$$= \frac{F_i}{F_{imax}}$$

F_i = the instantaneous force generated in the muscle

F_{imax} = the maximal isometric force in the muscle modified by optimal length effects and contraction rate effects [Zajac 1989]

Equality constraints, g(x), are defined which relate the torques generated by the muscles surrounding the joints to equal the torque required to articulate the joint in the observed manner as calculated by ann inverse dynamical analysis:

$$g(x) = \sum_j \left(r_i \text{ x } (F_{imax} \,.\, x_i)\right) - T_j = 0$$

where r_i = the radius of the lever arm of action of the ith muscle about jth rotation axis through the jth joint centre

T_j = the torque occurring at the joint due to the surrounding muscles about jth rotation axis

For an instantaneous configuration, r_i can be considered to be a constant therefore g(x) can be expressed as:

$$g(\mathbf{x}) = A_{eq} \,.\, \mathbf{x} - \mathbf{T} = \mathbf{0}$$

where A_{eq} = is a matrix of lever arms for the the ith muscle about the about jth rotation axis through the jth joint centre times the ith muscle's maximal isometric force

The minimum of the objective function subject to the equality constraint occurs at:

$$\frac{\partial f}{\partial x_i} = \sum_j \lambda_j \frac{\partial g}{\partial x_i}$$

where λ_j = jth Lagrangian multiplier

Expressing the Lagrangian expression in matrix form and including the condition that the equality constraints are valid results in:

$$\begin{Bmatrix} 0 \\ \mathbf{T} \end{Bmatrix} = \begin{bmatrix} 2I & A_{eq}^T \\ A_{eq} & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{Bmatrix}$$

It should be noticed that the above matrix is square and symmetric and therefore efficient methods can be employed for the solution of **x** and $\boldsymbol{\lambda}$, for example LU factorization with partial pivoting. To ensure that the solution for the muscle force lie between the upper and lower limits, partitions of the matrices will be defined. Define a g-set which consists of all of the variables; ie the muscles activations (**x**) and the Lagrange multipliers (**λ**):

$$k_{gg} = \begin{bmatrix} 2I & A_{eq}^T \\ A_{eq} & 0 \end{bmatrix}$$

$$\mathbf{v}_g = \begin{Bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{Bmatrix}; \qquad \mathbf{f}_g = \begin{Bmatrix} \mathbf{0} \\ \mathbf{T} \end{Bmatrix}$$

The g-set can be partitioned into 2 sets: the f-set (the variables which are within their prescribed limits as defined by the inequality constraints) and the s-set (the variables which are outside their prescribed limits as defined by the inequality constraints):

$$k_{gg} = \begin{bmatrix} k_{ff} & k_{fs} \\ k_{sf} & k_{ss} \end{bmatrix}$$

$$\mathbf{v}_g = \begin{Bmatrix} \mathbf{v}_f \\ \mathbf{v}_s \end{Bmatrix}; \qquad \mathbf{f}_g = \begin{Bmatrix} \mathbf{f}_f \\ \mathbf{f}_s \end{Bmatrix}$$

Assigning the limiting values of $\mathbf{v}_s$ which are known, and solving for $\mathbf{v}_f$:

$$\mathbf{v}_f = [\mathbf{f}_f - k_{sf}^T . \mathbf{v}_s]^{-1} k_{ff}$$

$$\mathbf{f}_s = k_{sf} . \mathbf{v}_f + k_{ss} . \mathbf{v}_s$$

The s-set is further partitioned into u-set (the variables which exceed their prescribed limit) and the l-set (the variables which are lower than their prescribed limit). Remove from the u-set, and hence the s-set, the elements which correspond to an entry in $\mathbf{f}_s$ less than zero. Remove from the l-set, and hence the s-set, the elements of the correspond to an entry in $\mathbf{f}_s$ which are greater than zero. Iteratively repeat for the solution of $\mathbf{v}_f$ until there is no modifications to the s-set. $\mathbf{v}_f$ will then contain the solution for the minimisation of the objective function subject to equality and inequality constraints together with the Lagrange multipliers.

RESULTS: The BoB musculoskeletal modelling system was used to generate the equality and inequality constraint equations for a full body model in a number of arbitrary poses subject to an arbitrary set of external forces; an example is shown in figure 1. Other examples can be viewed by searching for mendip89 on YouTube The musculoskeletal system consisted of 606 muscle forces and 30 joint torques.

The muscle force distribution was calculated using 3 methods:
1) The above described Lagrange multiplier / partitioning based method.
2) An active set algorithm [Gill 1981]
3) An interior point convex algorithm [Gould and Toint 2004]

All three methods calculated the same muscle loading distribution to within machine precision. However there was a significant difference in the demand on computational resource between the various methods. For the above trials, the Lagrange multiplier results were derived from translated Matlab [Mathworks, Natick, MA, USA] m-code whereas the active set and interior point convex algorithms were implemented using compiled code and hence the compilation of the former is expected to return even greater speed. Table 1 lists the solution times for the full body muscle load distribution problem running on an i7 laptop:

Table 1
Solution times for differing methods

Method	Solution time
Lagrange multiplier	0.052s
Interior point convex	0.647s
Active set	18.785s

CONCLUSION: A method has been described which is capable of solving the full body muscle load distribution which occur during arbitrary sporting activities within approximately one twentieth of a second on a laptop computer. This speed of solution is commensurate with the requirements of a system providing real time feedback to a subject undergoing a biomechanical sports analysis of muscle, skeletal and joint loads. The approach lends itself to compact, robust code development. The Matlab m-code implementation of the above method consisted of 54 lines of arithmetic and command control code. If a search method is to be implemented to minimise an alternative objective function, it is suggested that the above method be used as a starting position for the search due to its low computational cost.

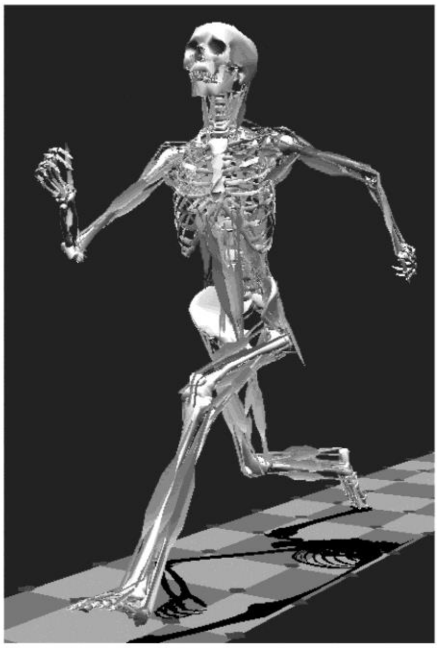

Fig 1: Example of calculated muscle activation

References

Arfken, G. (1985) Lagrange Multipliers *Section 17.6 in Mathematical Methods for Physicists*, (3rd ed). Orlando, FL: Academic Press, pp945-950

Crowninshield, R.D., Brand, R.A., (1981). A physiologically based criterion of muscle force prediction in locomotion. *Journal of Biomechanics* 14,793–801.

Delp, S.L, Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Guendelman, & E., Thelen, D.G., (2007). OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, 54,1940–1950

Gill, P.E., Murray, W. and Wright, M.H., (1981) *Practical Optimization*, Academic Press, London, UK

Gould, N. & Toint, P.L. (2004). Preprocessing for quadratic programming, *Math. Programming, Series B,* 100, 95–132,

Koopman, B., Grootenboer, H.J. & de Jongh, H.J. (1995). An inverse dynamics model for the analysis, reconstruction and prediction of bipedal walking. *J Biomechanics* 28, 1369–1376 DOI: 10.1016/0021-9290(94)00185-7

Modenese, L., Phillips, A.T.M. & Bull, A.M.J. (2011). An open source lower limb model: Hip joint validation. *J Biomechanics* 44 (12), 11, 2185–2193

Shippen, J.M. & May, B. (2010). Calculation of Muscle Loading and Joint Contact Forces in Irish Dance. *Journal of Dance Science and Medicine*, 14(1) 11-18

Thelen, D.G., Anderson, F.C., Delp, S.L., (2003). Generating dynamic simulations of movement using computed muscle control. *Journal of Biomechanics* 36, 321–328.

Zajac, F.E. (1989). Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Crit Rev Biomed Eng*. 17(4):359-411.