HAB718 Spor Biyomekaniğinde Hareket Analizi

Serdar Arıtan

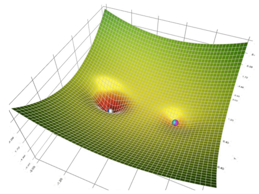
Hacettepe Üniversitesi
Spor Bilimleri Fakültesi
Biyomekanik Araştırma Grubu

serdar.aritan@hacettepe.edu.tr

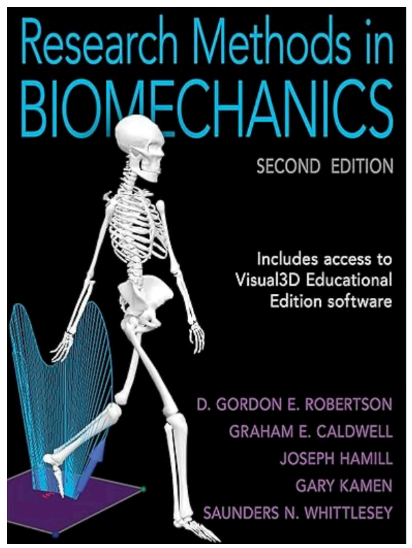
#11

- Muscle Force Optimization

HAB718 Spor Biyomekaniğinde Hareket Analizi



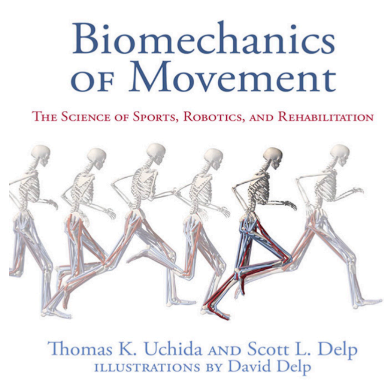
Muscle Force Optimization



Chapter 11

Musculoskeletal Modeling

Brian R. Umberger and Graham E. Caldwell

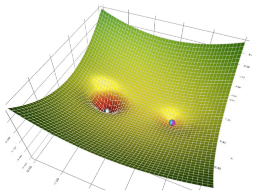


9

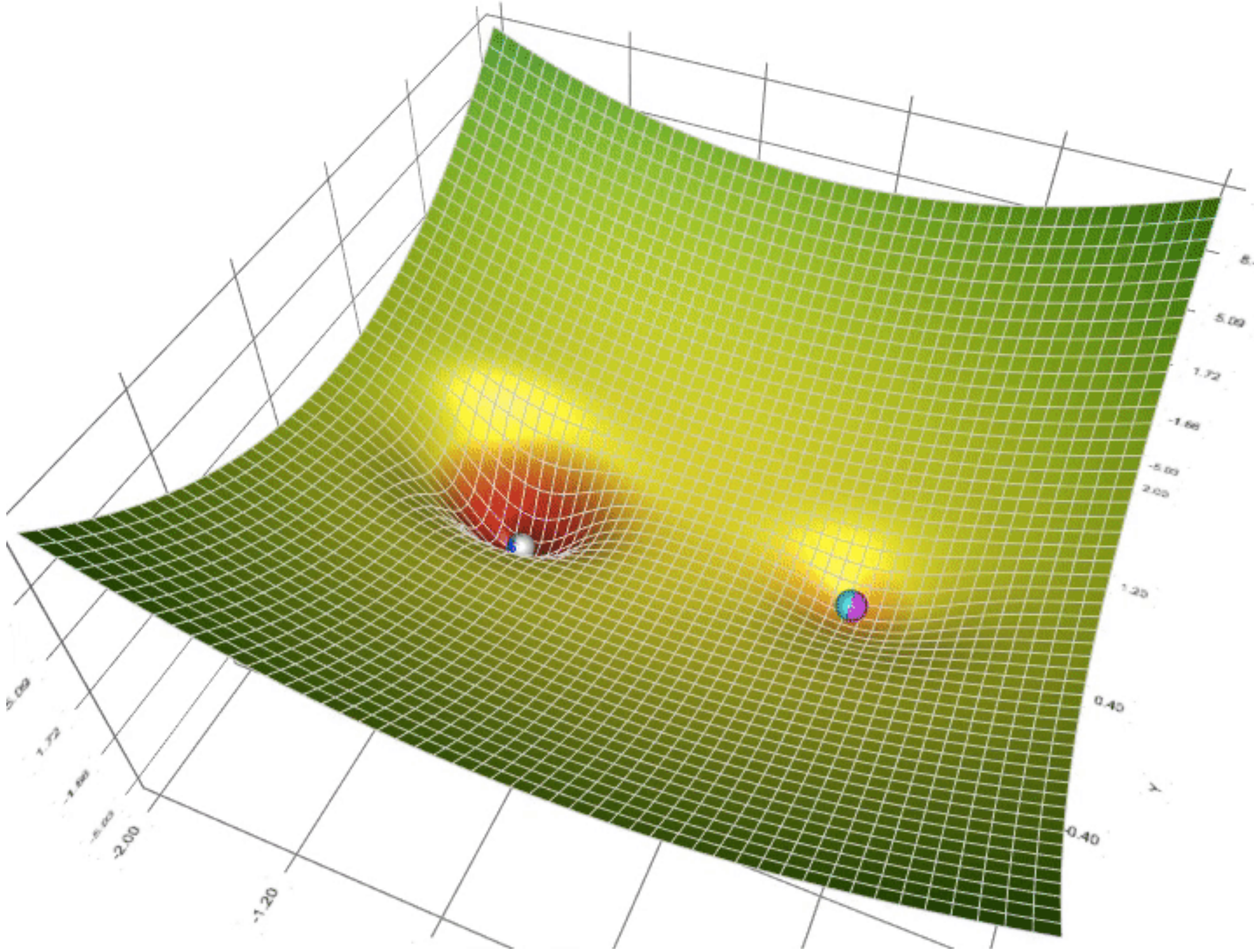
Muscle Force Optimization

There is always a well-known solution to every human problem—neat, plausible, and wrong.
—H. L. Mencken

HAB718 Spor Biyomekaniğinde Hareket Analizi

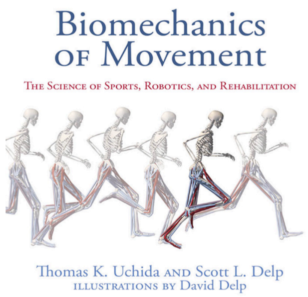


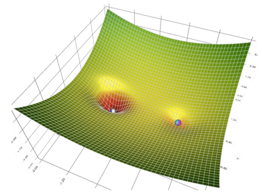
Muscle Force Optimization



Animation of 5 gradient descent methods on a surface: gradient descent (cyan), momentum (magenta), AdaGrad (white), RMSProp (green), Adam (blue). Left well is the global minimum; right well is a local minimum.

https://github.com/lilipads/gradient_descent_viz

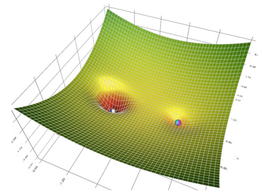




Muscle Force Optimization

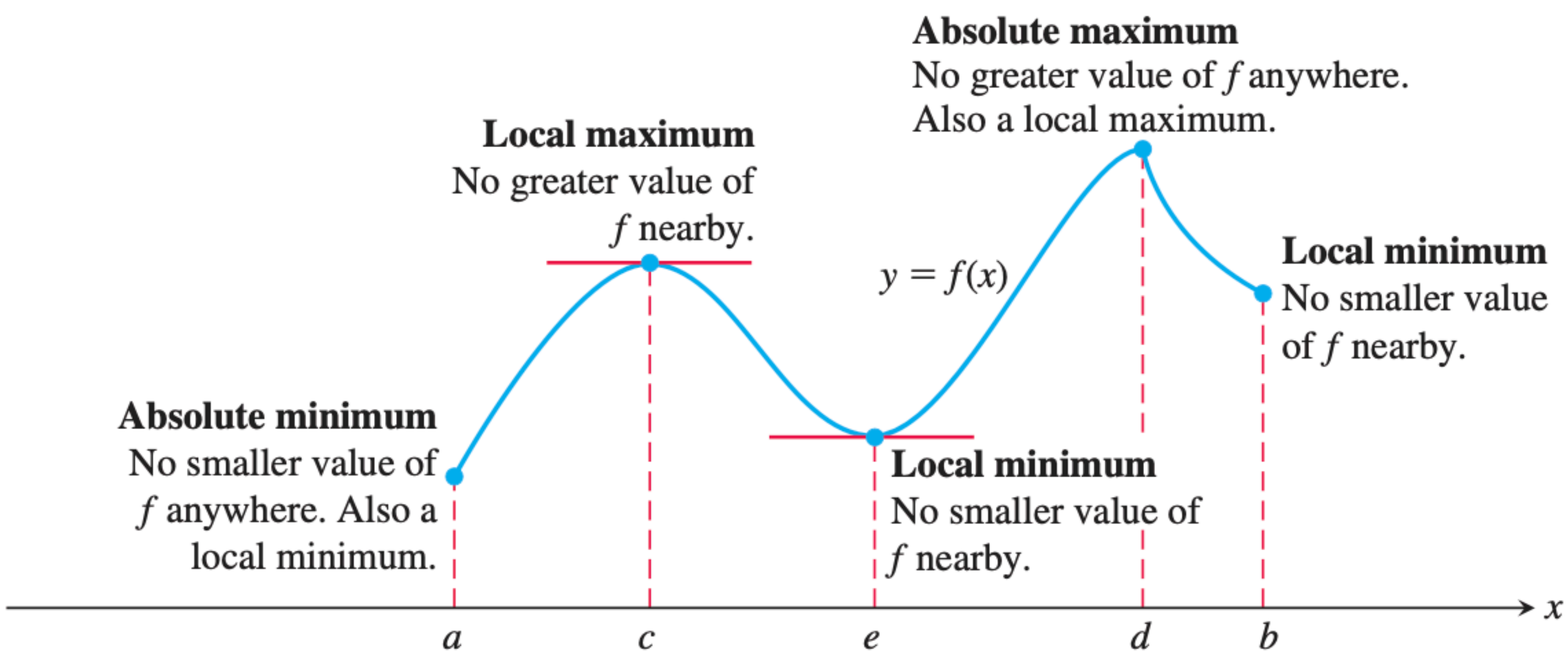
In mathematics, engineering, computer science and economics an **optimization problem** is the problem of finding the *best* solution from all feasible solutions.

$$
\begin{aligned}
&\textit{minimize} && f(x) && \\
&\textit{subject to} && g_i(x) \leq 0, && i = 1,\ldots,m \\
& && h_j(x) = 0, && j = 1,\ldots,p
\end{aligned}
$$

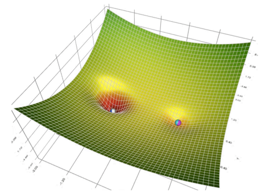


Muscle Force Optimization

Maximum and minimum values are called extreme values of the function $f(x)$.

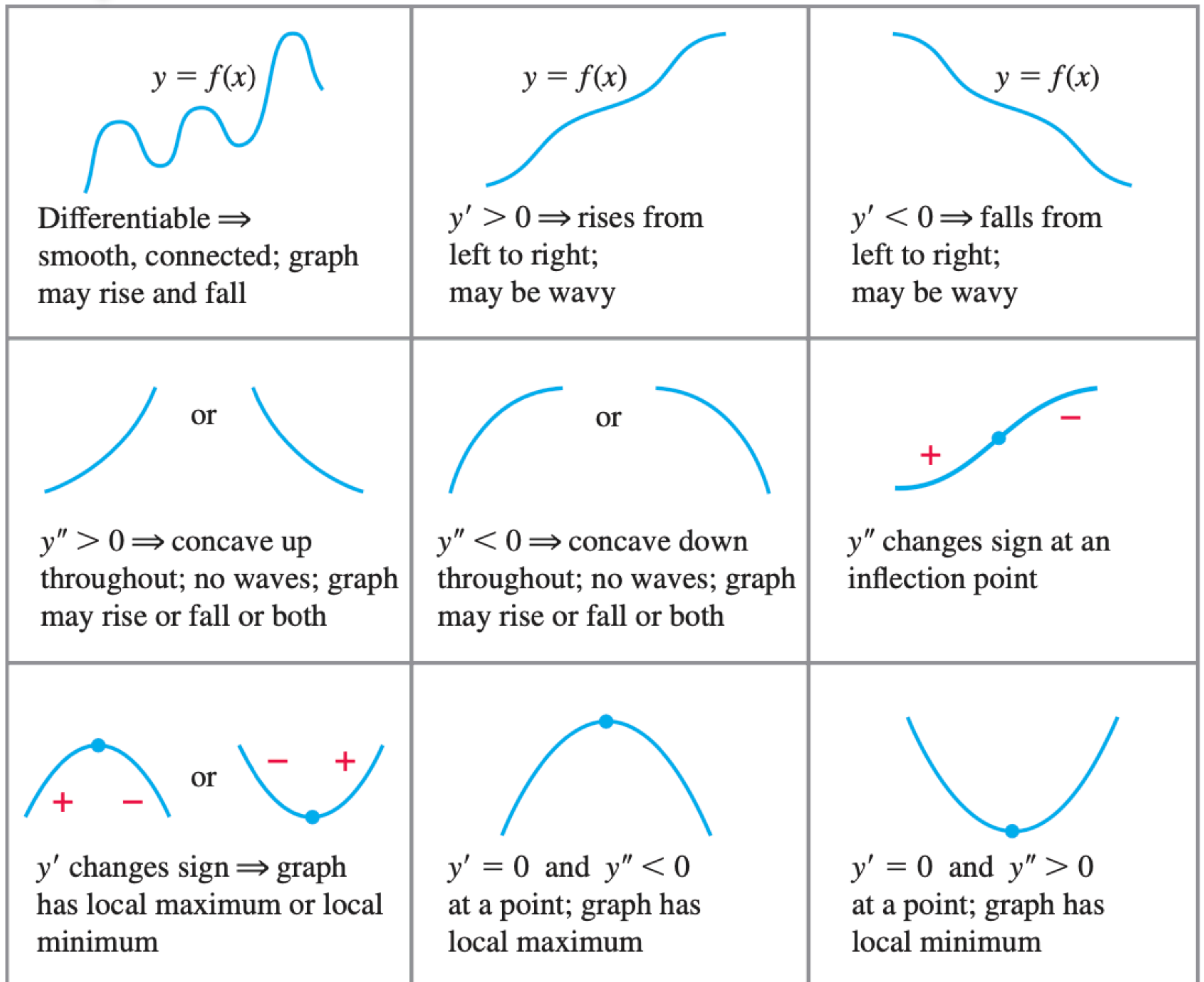


HAB718 Spor Biyomekaniğinde Hareket Analizi

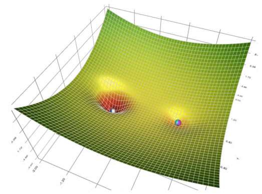


Muscle Force Optimization

Graphical Behavior of Functions from Derivatives

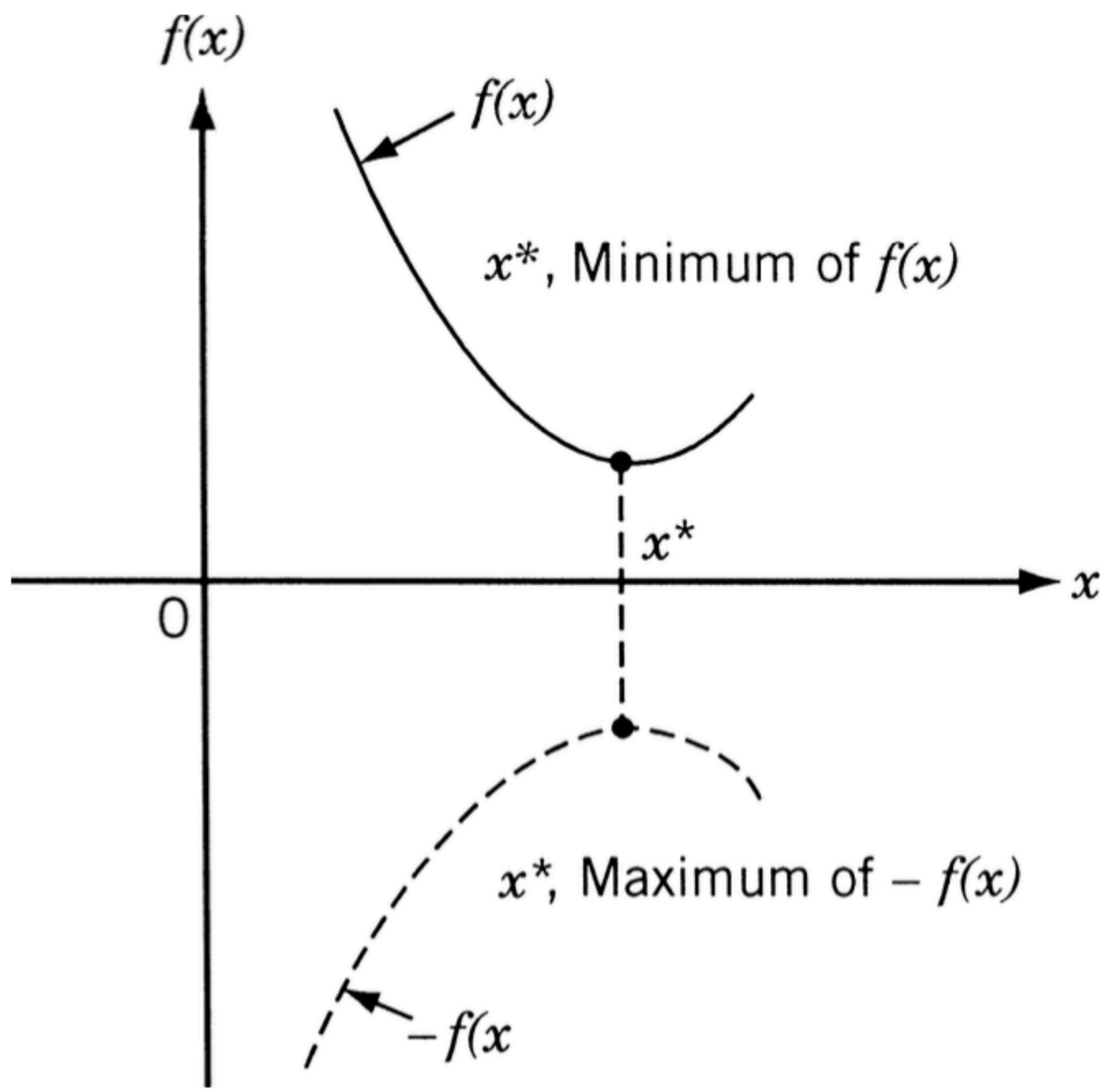


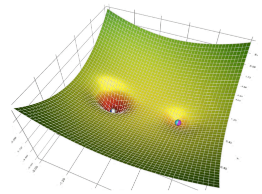
HAB718 Spor Biyomekaniğinde Hareket Analizi



Muscle Force Optimization

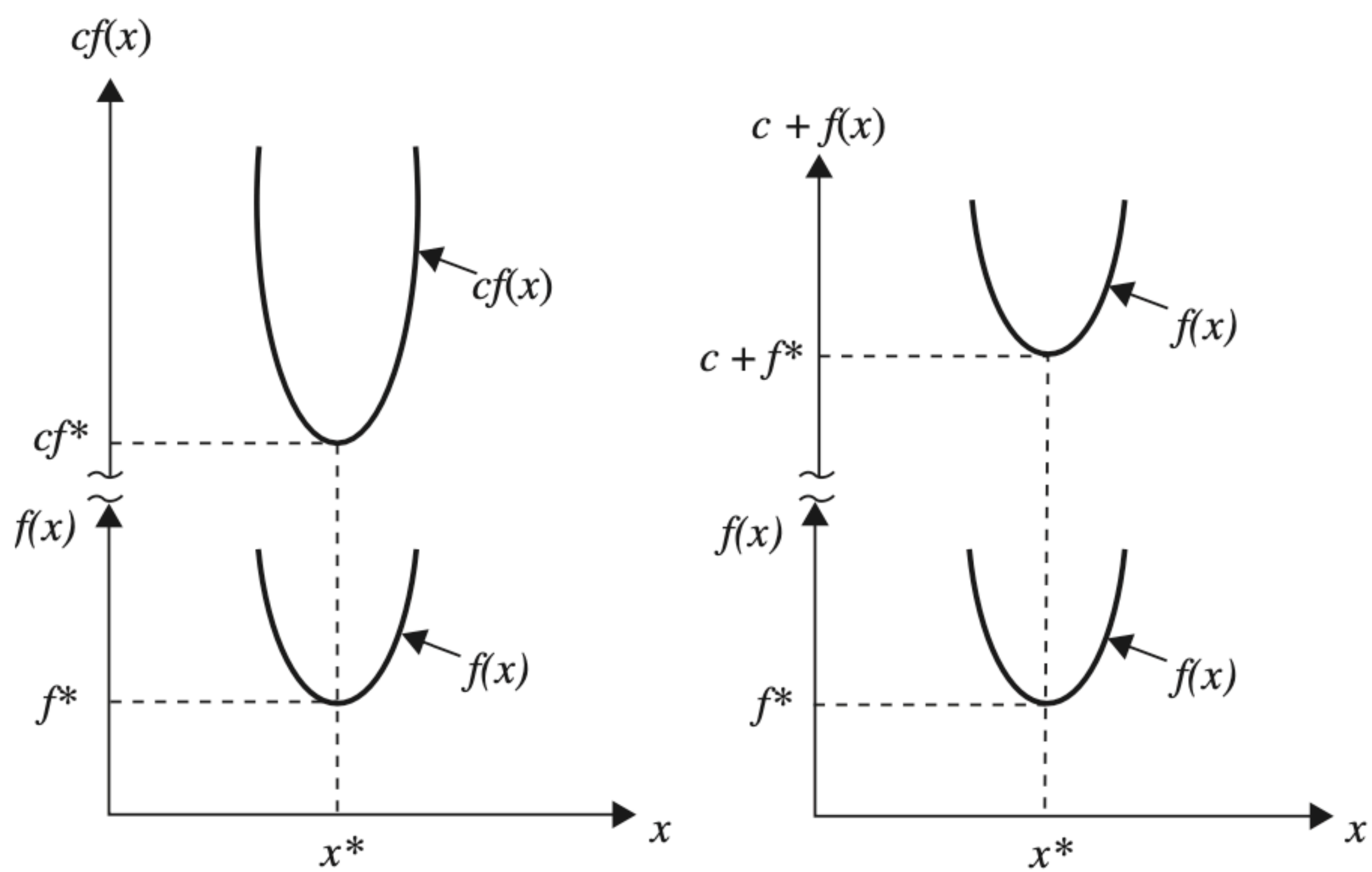
Minimum of $f(x)$ is same as maximum of $-f(x)$

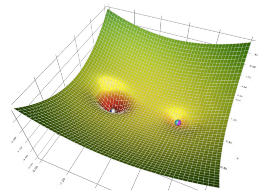




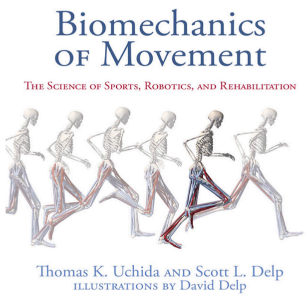
Muscle Force Optimization

Optimum solution of $cf(x)$ or $c + f(x)$ same as that of $f(x)$





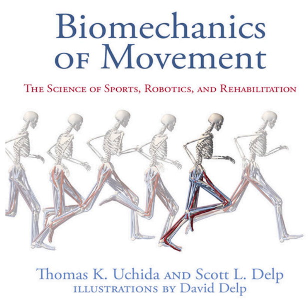
Muscle Force Optimization

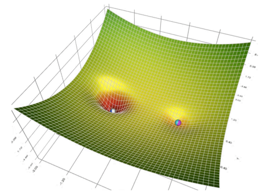


Methods of Operations Research

Mathematical programming or optimization techniques	Stochastic process techniques	Statistical methods
Calculus methods	Statistical decision theory	Regression analysis
Calculus of variations	Markov processes	Cluster analysis, pattern recognition
Nonlinear programming	Queueing theory	
Geometric programming	Renewal theory	Design of experiments
Quadratic programming	Simulation methods	Discriminate analysis (factor analysis)
Linear programming	Reliability theory	
Dynamic programming		
Integer programming		
Stochastic programming		
Separable programming		
Multiobjective programming		
Network methods: CPM and PERT		
Game theory		
Modern or nontraditional optimization techniques		
Genetic algorithms		
Simulated annealing		
Ant colony optimization		
Particle swarm optimization		
Neural networks		
Fuzzy optimization		

HAB718 Spor Biyomekaniğinde Hareket Analizi





Muscle Force Optimization

Solving Optimization Problems

https://docs.octave.org/latest/Minimizers.html

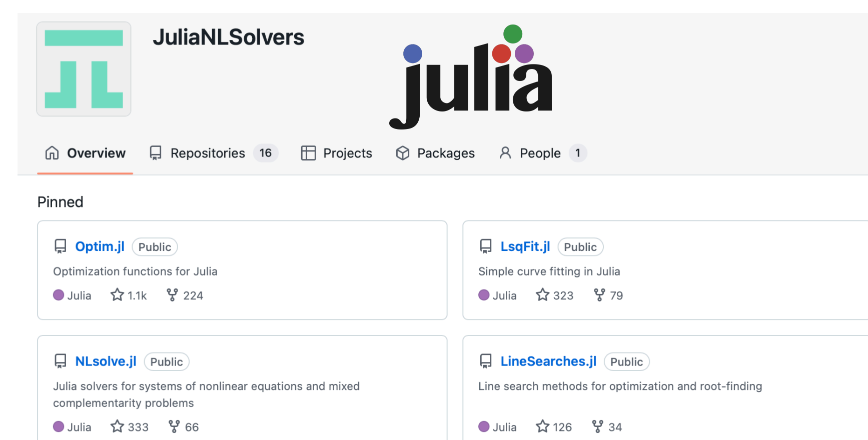
Type of optimization problem	Standard form for solution by MATLAB	Name of MATLAB program or function to solve the problem
Function of one variable or scalar minimization	Find x to minimize $f(x)$ with $x_1 < x < x_2$	`fminbnd`
Unconstrained minimization of function of several variables	Find $\mathbf{x}$ to minimize $f(\mathbf{x})$	`fminunc` or `fminsearch`
Linear programming problem	Find $\mathbf{x}$ to minimize $\mathbf{f}^T\mathbf{x}$ subject to $[A]\mathbf{x} \le \mathbf{b}$, $[A_{eq}]\mathbf{x} = \mathbf{b}_{eq}$, $\mathbf{l} \le \mathbf{x} \le \mathbf{u}$	`linprog`
Quadratic programming problem	Find $\mathbf{x}$ to minimize $\frac{1}{2}\mathbf{x}^T[H]\mathbf{x} + \mathbf{f}^T\mathbf{x}$ subject to $[A]\mathbf{x} \le \mathbf{b}$, $[A_{eq}]\mathbf{x} = \mathbf{b}_{eq}$, $\mathbf{l} \le \mathbf{x} \le \mathbf{u}$	`quadprog`
Minimization of function of several variables subject to constraints	Find $\mathbf{x}$ to minimize $f(\mathbf{x})$ subject to $\mathbf{c}(\mathbf{x}) \le \mathbf{0}$, $\mathbf{c}_{eq} = \mathbf{0}$ $[A]\mathbf{x} \le \mathbf{b}$, $[A_{eq}]\mathbf{x} = \mathbf{b}_{eq}$, $\mathbf{l} \le \mathbf{x} \le \mathbf{u}$	`fmincon`
Goal attainment problem	Find $\mathbf{x}$ and γ to minimize γ such that $F(\mathbf{x}) - \mathbf{w}\gamma \le \mathbf{goal}$, $\mathbf{c}(\mathbf{x}) \le \mathbf{0}$, $\mathbf{c}_{eq} = \mathbf{0}$ $[A]\mathbf{x} \le \mathbf{b}$, $[A_{eq}]\mathbf{x} = \mathbf{b}_{eq}$, $\mathbf{l} \le \mathbf{x} \le \mathbf{u}$	`fgoalattain`
Minimax problem	$\underset{\mathbf{x}}{\text{Minimize}}\ \underset{[F_i]}{\text{Max}}\ \{F_i(\mathbf{x})\}$ such that $\mathbf{c}(\mathbf{x}) \le \mathbf{0}$, $\mathbf{c}_{eq} = \mathbf{0}$ $[A]\mathbf{x} \le \mathbf{b}$, $[A_{eq}]\mathbf{x} = \mathbf{b}_{eq}$, $\mathbf{l} \le \mathbf{x} \le \mathbf{u}$	`fminimax`
Binary integer programming problem	Find $\mathbf{x}$ to minimize $\mathbf{f}^T\mathbf{x}$ subject to $[A]\mathbf{x} \le \mathbf{b}$, $[A_{eq}]\mathbf{x} = \mathbf{b}_{eq}$, each component of $\mathbf{x}$ is binary	`bintprog`

scipy.optimize.
fminbound

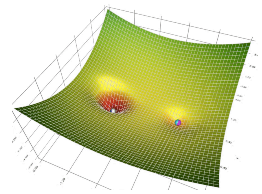
```
fminbound(func, x1, x2, args=(), xtol=1e-05, maxfun=500, full_output=0, disp=1)
```
[source]

Bounded minimization for scalar functions.

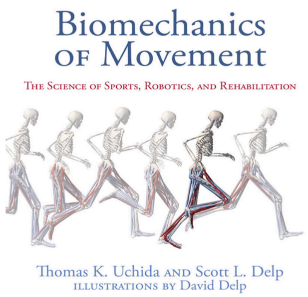
https://docs.scipy.org/doc/scipy/reference/optimize.html



https://github.com/JuliaNLSolvers



Muscle Force Optimization



$$\textit{minimize} \quad f(F)$$

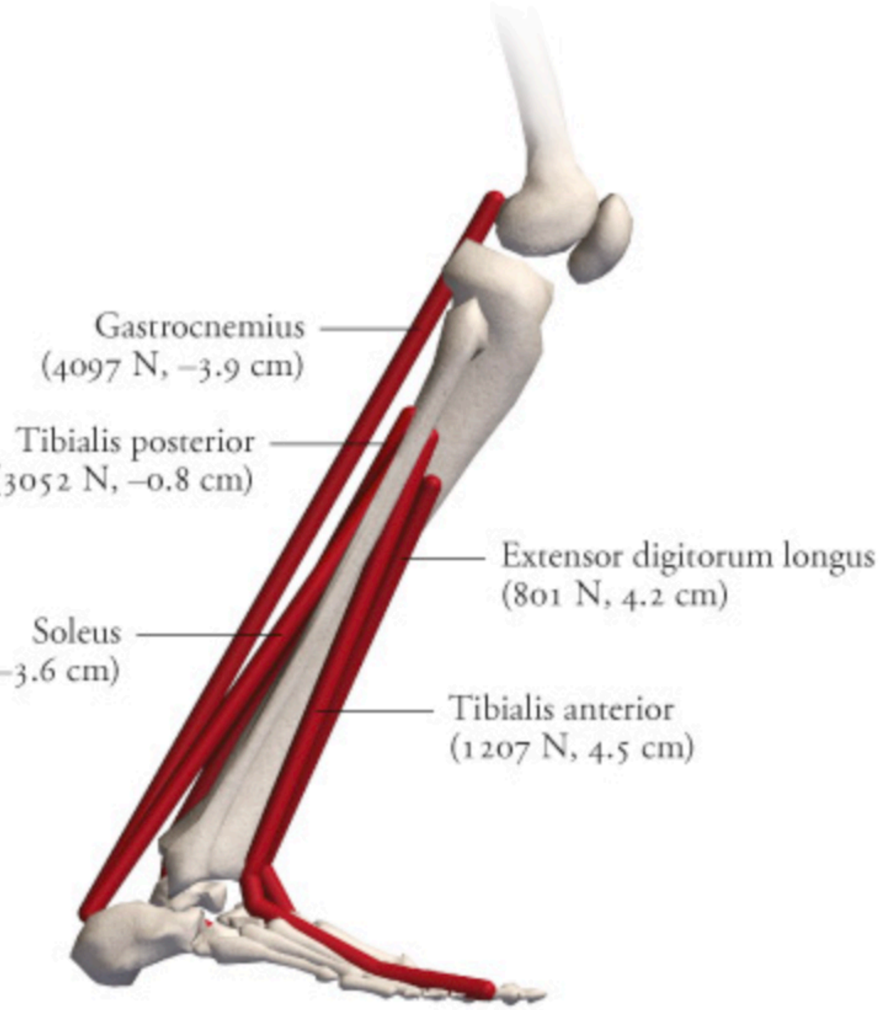
$$\textit{subject to} \quad 0.039F^{GAS} + 0.036F^{SOL} + 0.008F^{TP} = 100$$

$$0 \leq F^{GAS} \leq 4097$$
$$0 \leq F^{SOL} \leq 6435$$
$$0 \leq F^{TP} \leq 3052$$

Optimization Problem 1: Find the ankle plantarflexor muscle forces that produce a desired joint moment

minimize	$f(F)$	Objective function where smaller values are favored
subject to	$0.039F^{GAS} + 0.036F^{SOL} + 0.008F^{TP} = 100$	Muscles must produce the desired net ankle moment
	$0 \leq F^{GAS} \leq 4097$ $0 \leq F^{SOL} \leq 6435$ $0 \leq F^{TP} \leq 3052$	Muscle forces must lie within physiological ranges

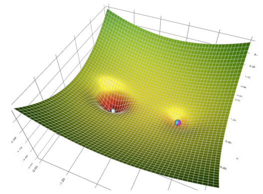
One approach to solving an underdetermined problem is to modify the problem so that there are as many equations as unknowns. In our example, **we will assume for simplicity that the dorsiflexors are inactive and generate zero force.** This assumption reduces the number of unknowns from five to three: the force generated by the gastrocnemius (which we will call F^{GAS}), soleus (F^{SOL}), and tibialis posterior (F^{TP}).



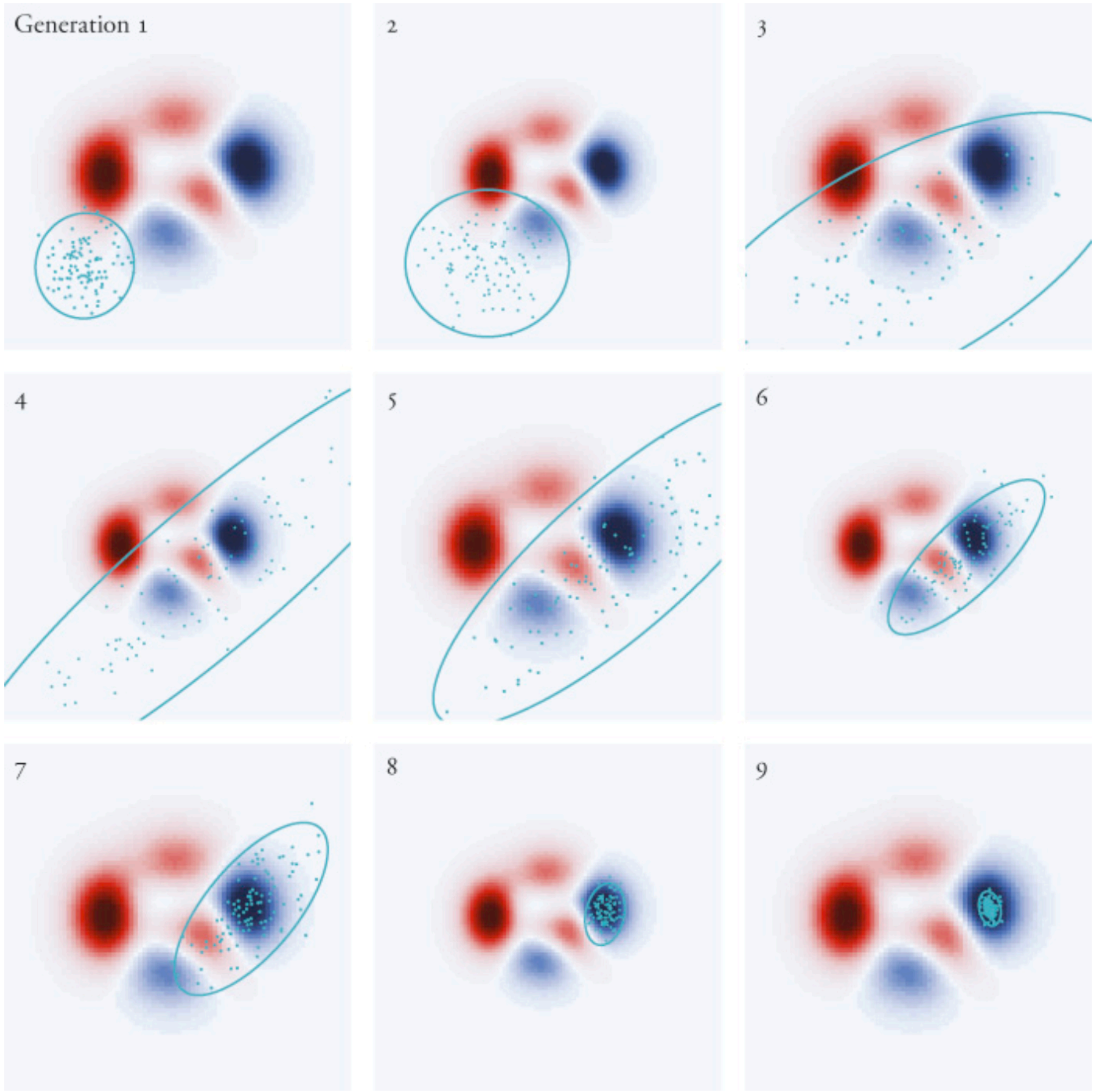
The solution in this case is F^{GAS} = ***2564***, F^{SOL} = ***0***, and F^{TP} = ***0***, because the gastrocnemius has the largest moment arm and therefore can generate the desired ankle plantarflexion moment using the least amount of force.

Figure 9.2 A musculoskeletal model of the shank and foot with the key plantarflexor and dorsiflexor muscles. A measured ankle moment could have been generated by many combinations of muscle forces. Values in parentheses are the instantaneous maximum force (assuming zero velocity and a rigid tendon) and moment arm at the ankle corresponding to the pose shown.

HAB718 Spor Biyomekaniğinde Hareket Analizi

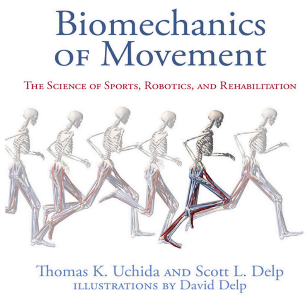


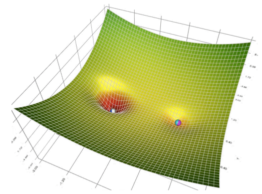
Muscle Force Optimization



Covariance matrix adaptation evolution strategy (CMA-ES) is a particular kind of strategy for numerical optimization.

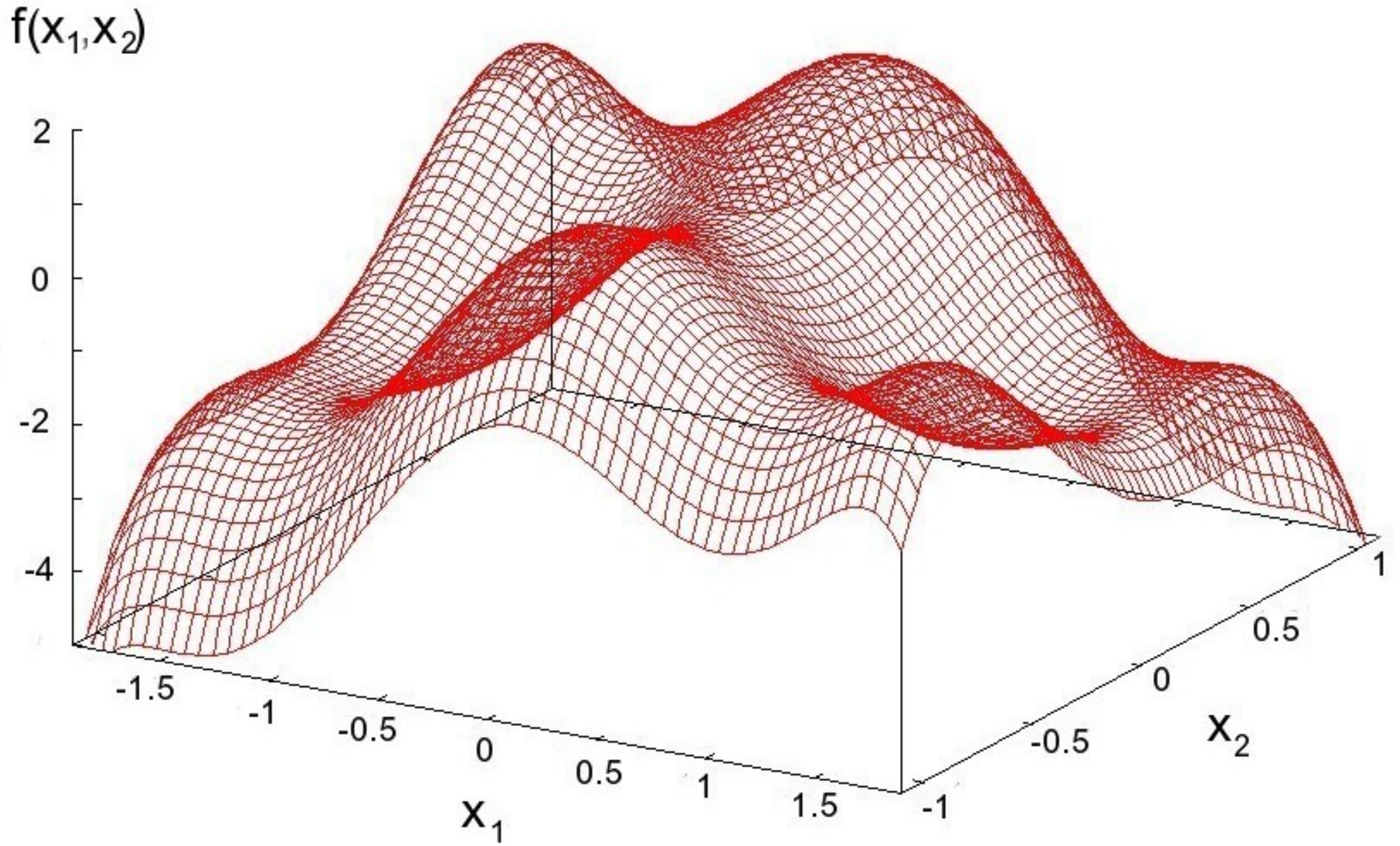
Figure 9.7 Global optimization methods like the covariance matrix adaptation evolution strategy (CMA-ES) avoid converging to local minima. Shown here are nine generations of the CMA-ES algorithm as it minimizes the function shown in Figure 9.4. Each cyan dot represents a candidate solution; the ellipse in each panel indicates the distribution of the population in that generation.

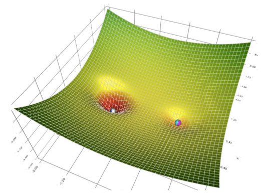




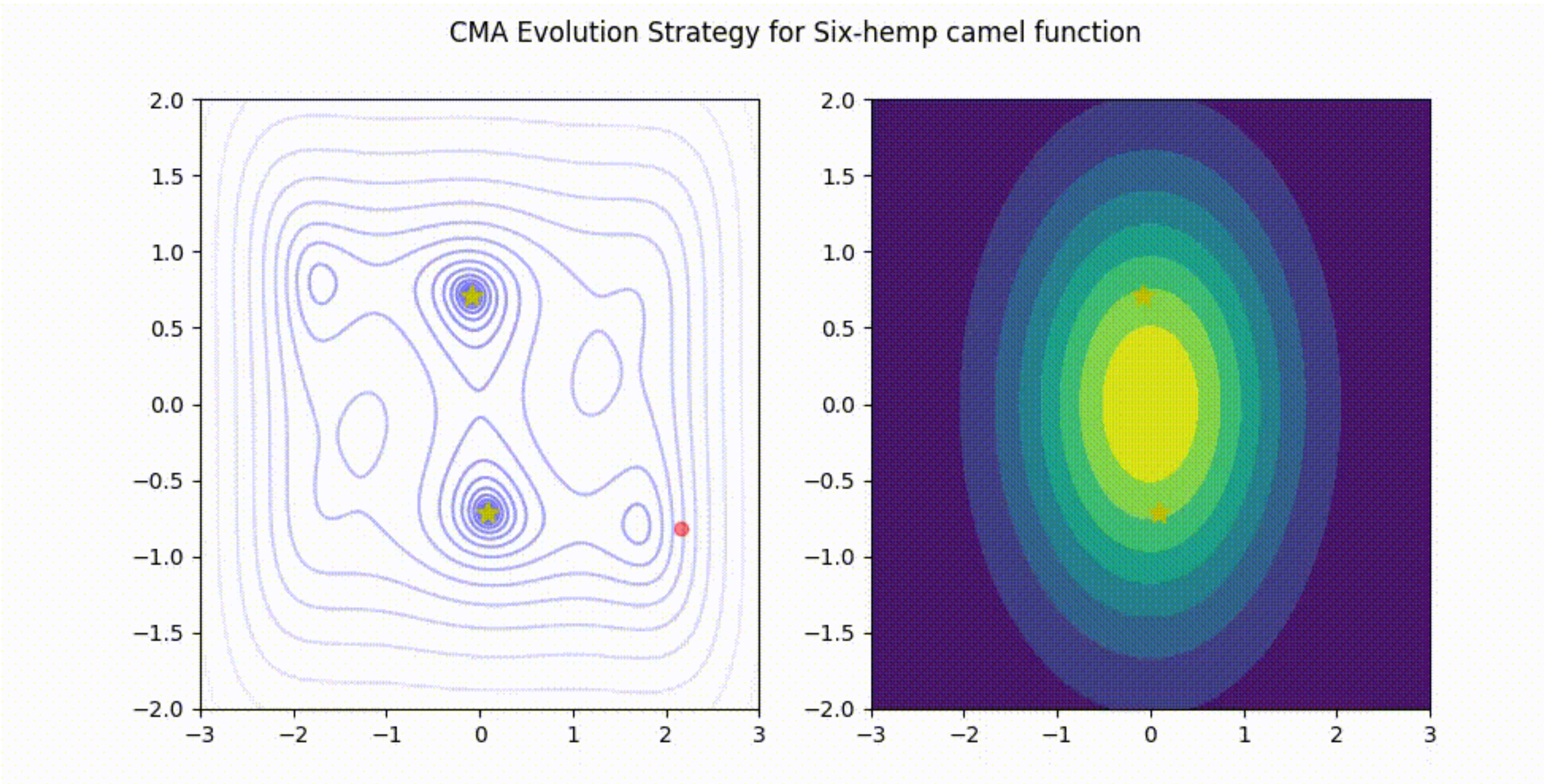
Muscle Force Optimization

The Six Hump Camel Function

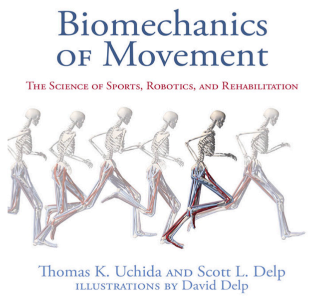


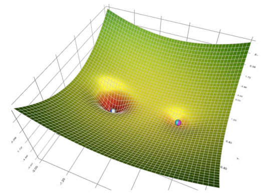


Muscle Force Optimization



HAB718 Spor Biyomekaniğinde Hareket Analizi





Muscle Force Optimization

Optimization Problem 2: Find muscle activations at the instant of peak vertical ground reaction force during running

$$\text{minimize} \quad J(\underline{a}) = \sum_{i=1}^{9} a_i^2$$

$$\text{subject to} \quad \sum_{i=1}^{9} a_i \left(r_i^{\text{hip}} F_i^{\max}\right) = -67.3$$

$$\sum_{i=1}^{9} a_i \left(r_i^{\text{knee}} F_i^{\max}\right) = -139$$

$$\sum_{i=1}^{9} a_i \left(r_i^{\text{ankle}} F_i^{\max}\right) = -206$$

$$0 \le a_i \le 1 \quad \text{for } i = 1, \ldots, 9$$

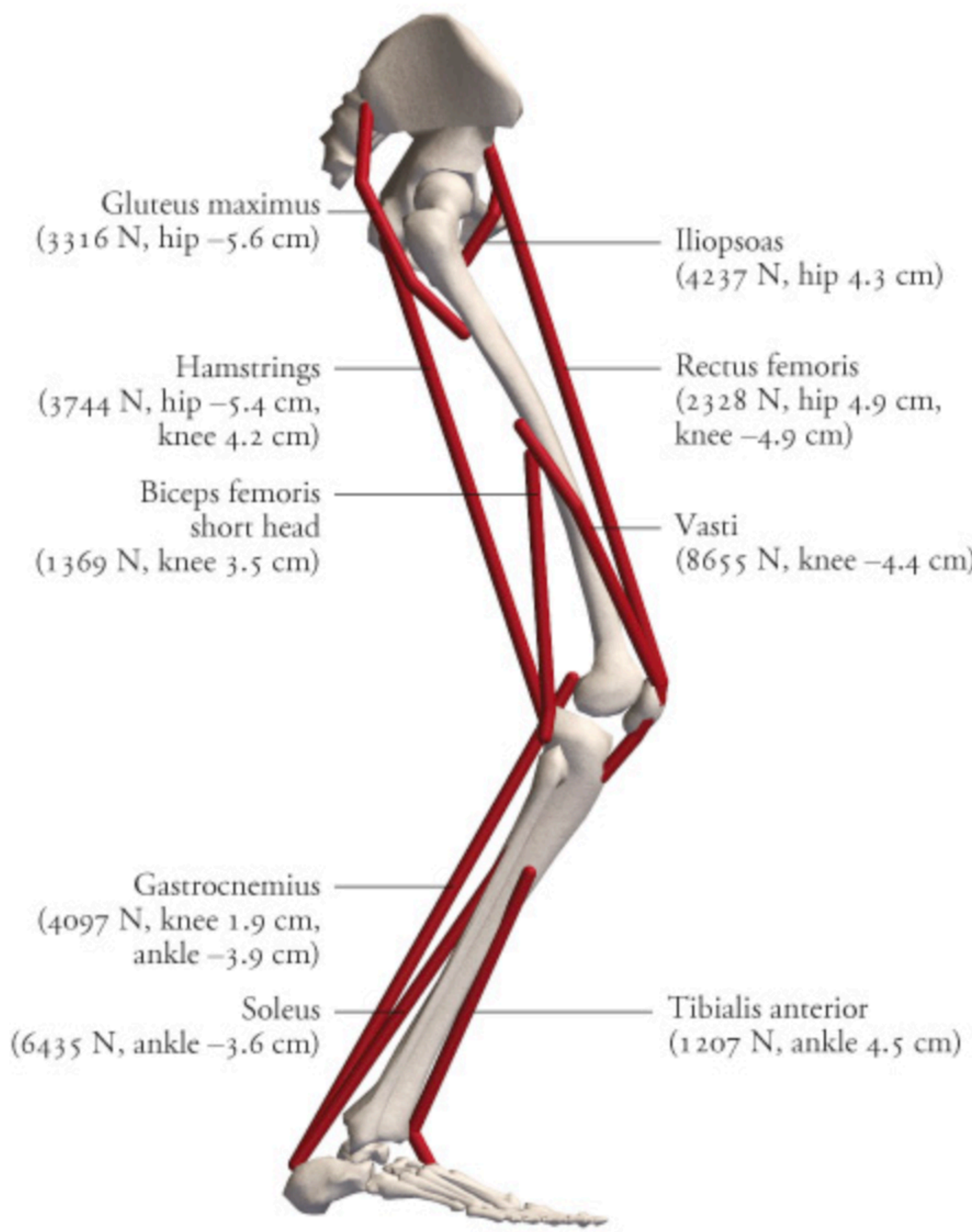
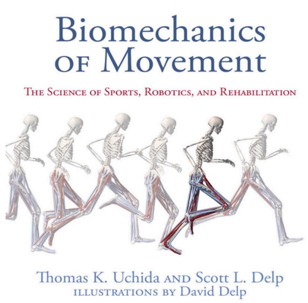


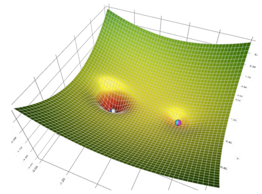
Figure 9.9 A simple musculoskeletal model of the leg can be used to study muscle coordination and joint loads during the stance phase of running. Key muscles involved in generating sagittal-plane movement have been grouped into nine representative muscle paths (Hamner and Delp, 2013). Values in parentheses are the instantaneous maximum force (assuming zero velocity and a rigid tendon) and moment arms corresponding to the pose shown.

We can estimate the forces generated by each muscle during walking and running by repeating this analysis at evenly

Muscle or group	**Force, F_i (N)**
Gluteus maximus	875
Iliopsoas	0
Hamstrings	340
Rectus femoris	0
Biceps femoris short head	0
Vasti	4134
Gastrocnemius	1396
Soleus	4167
Tibialis anterior	0

HAB718 Spor Biyomekaniğinde Hareket Analizi





Muscle Force Optimization

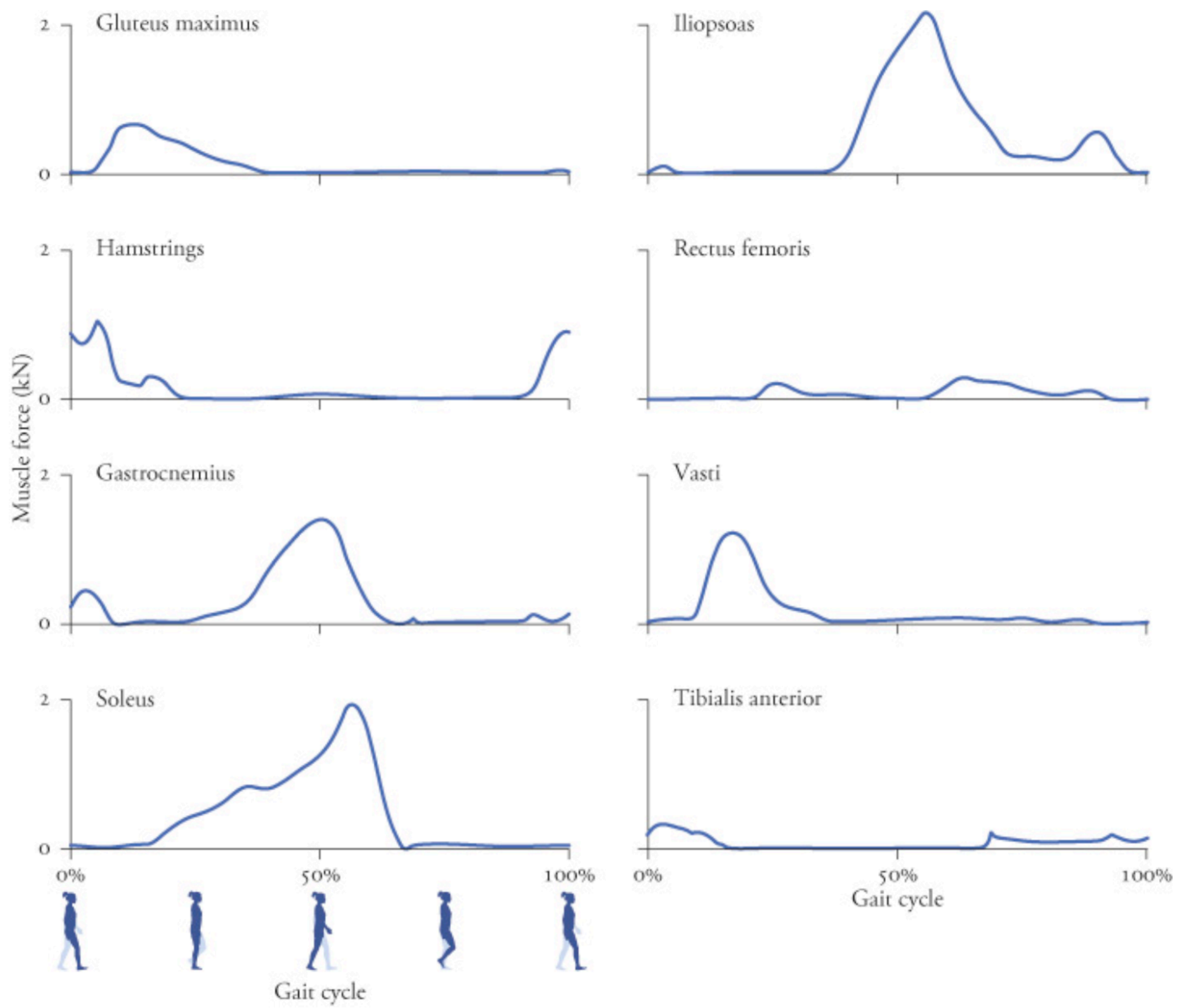
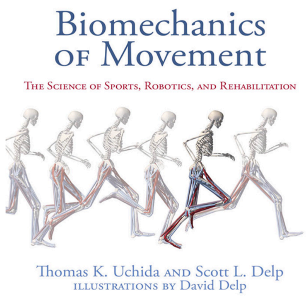
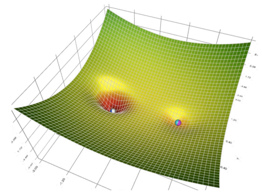


Figure 9.10 Muscle forces for one subject (male, 67.1 kg) walking at freely selected speed (1.67 m/s), computed using Static Optimization in OpenSim. Data from Dembia et al. (2017).

HAB718 Spor Biyomekaniğinde Hareket Analizi





Muscle Force Optimization

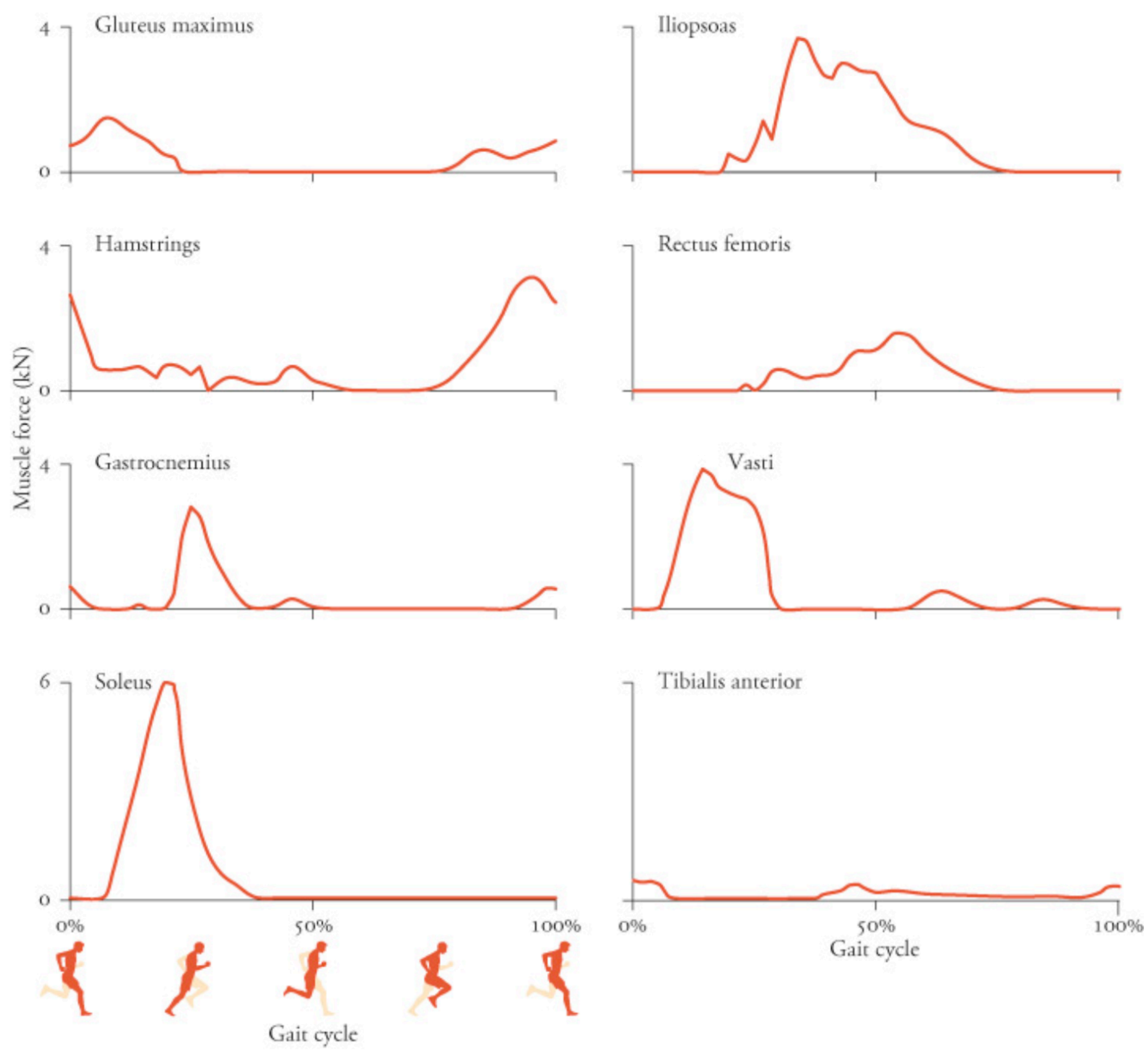
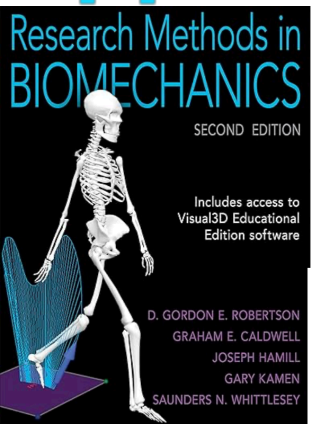


Figure 9.12 Muscle forces for one subject (male, 69.4 kg) running at 5 m/s, computed using Static Optimization in OpenSim. Data from Hamner and Delp (2013).



Musculoskeletal Modeling

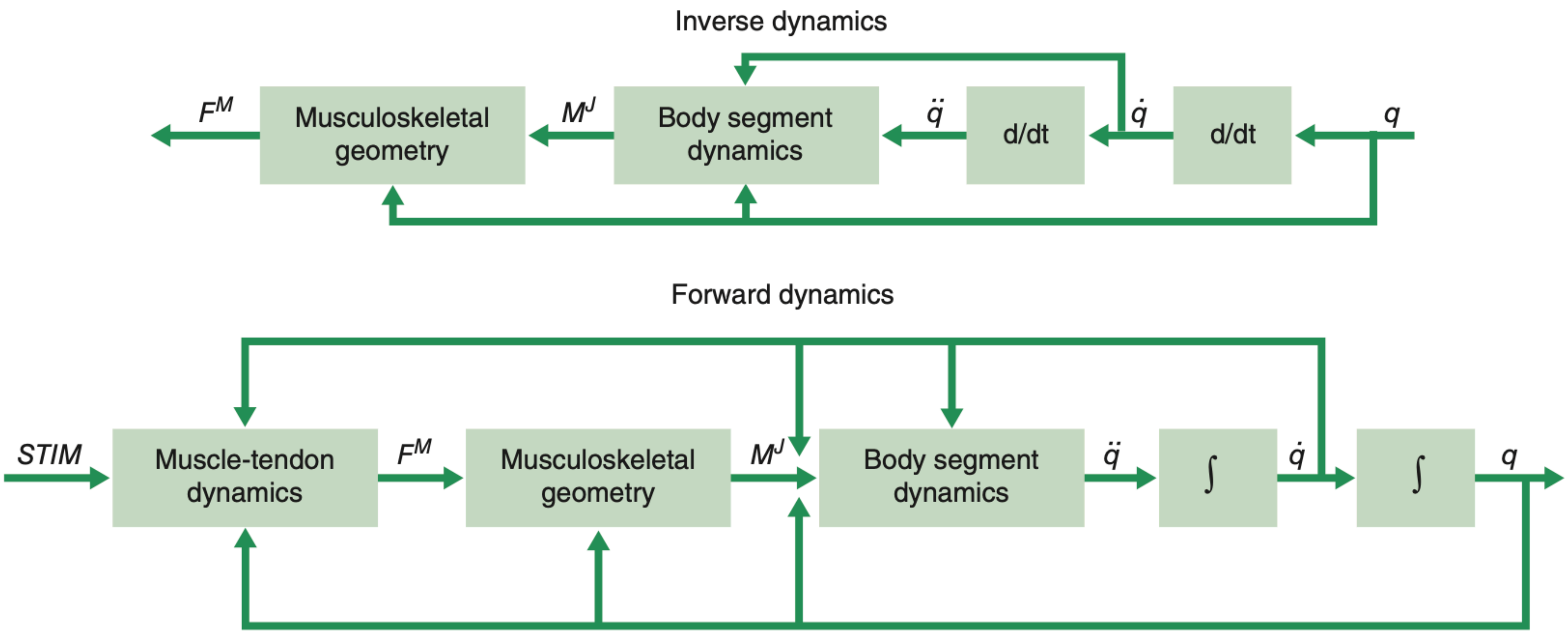
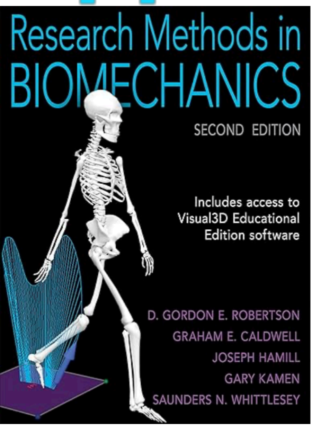


Figure 11.1 Overview of musculoskeletal model-based inverse dynamics and forward dynamics analyses. In inverse dynamics, the inputs are body segment positions *(q)* and the outputs are muscle forces *(F^M)*. In forward dynamics, the inputs are muscle stimulation patterns *(STIM)* and the outputs are body segment positions *(q)*. In both cases, musculoskeletal geometry defines the transformations between muscle forces *(F^M)* and joint moments *(M^J)*.

Modified from Pandy 2001.

HAB718 Spor Biyomekaniğinde Hareket Analizi

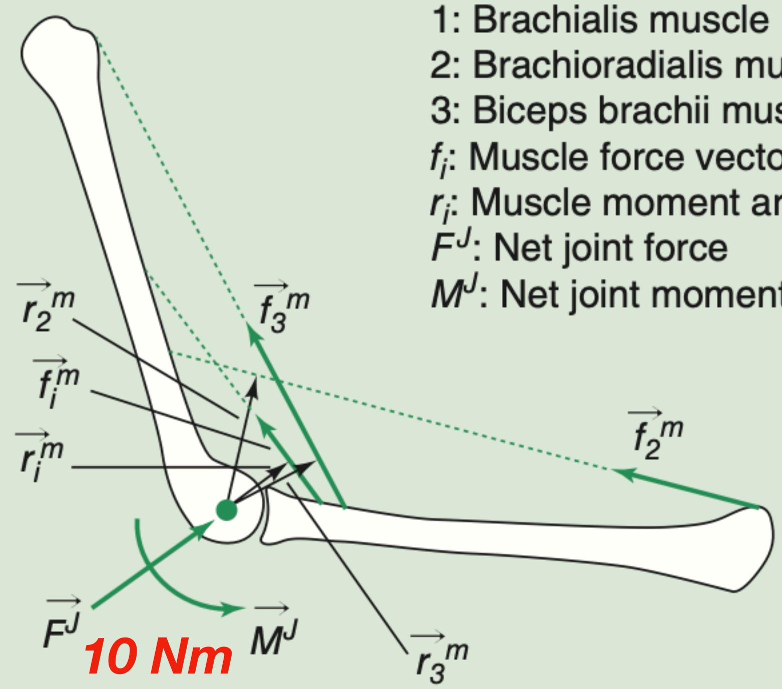


Musculoskeletal Modeling

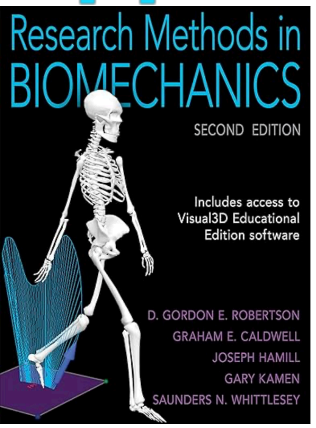
EXAMPLE 11.1

The human upper limb is depicted in figure 11.10, with the arm and forearm segments articulating at a one DOF elbow joint, which is crossed by three muscles representing the brachialis, brachioradialis, and biceps brachii. Numerical values for individual muscle parameters used in this example are provided in table 11.1. We will assume that an inverse dynamics analysis revealed a net elbow flexor moment ($\vec{M}^{j}$) of 10 N·m. Static optimization will be used to solve for the forces in the three muscles that gave rise to this measured joint moment. To solve this otherwise indeterminate problem, we will seek the combination of muscle forces that minimize the sum of the cubed muscle stresses.

Key
1: Brachialis muscle
2: Brachioradialis muscle
3: Biceps brachii muscle
f_i: Muscle force vectors
r_i: Muscle moment arm vectors
F^J: Net joint force
M^J: Net joint moment



▲ **Figure 11.10** A simplified one DOF model of the human elbow joint with three flexor muscles. See text and table 11.1 for additional details. (Adapted from Crowninshield and Brand 1981b.)



Musculoskeletal Modeling

Table 11.1

	Muscle 1 Brachialis	Muscle 2 Brachioradialis	Muscle 3 Biceps brachii
f^0 (N)	1000	250	700
A^m (m^2)	0.0033	0.0008	0.0023
r^m (m)	0.02	0.05	0.04

f^0 = maximum muscle force; A^m = physiological cross-sectional area; r^m = muscle moment arm.

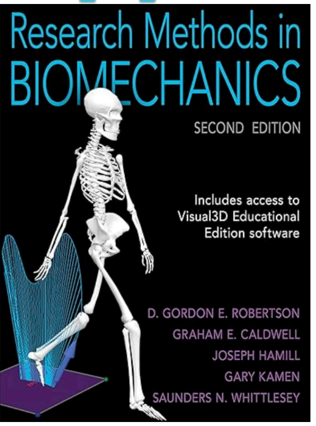
In this musculoskeletal system, the net joint *(J)* forces and moments determined using inverse dynamics are related to the internal muscle *(m)*, ligament *(l)*, and articular contact *(c)* forces *(f)* by the following equations:

$$\vec{F}^J = \sum_{i=1}^{m} \vec{f}_i^m + \sum_{i=1}^{l} \vec{f}_i^l + \sum_{i=1}^{c} \vec{f}_i^c \qquad (11.7)$$

$$\vec{M}^J = \sum_{i=1}^{m} \left(\vec{r}_i^m \times \vec{f}_i^m\right) + \sum_{i=1}^{l} \left(\vec{r}_i^l \times \vec{f}_i^l\right) + \sum_{i=1}^{c} \left(\vec{r}_i^c \times \vec{f}_i^c\right) \qquad (11.8)$$

which indicate that the net joint force $(\vec{F}^J)$ from inverse dynamics is equal to the vector sum of the muscle, ligament, and articular contact forces (equation 11.7), whereas the net joint moment $(\vec{M}^J)$ is equal to the vector sum of the moments generated by muscle, ligament, and articular contact forces (equation 11.8). If we make the common simplifying assumptions that (1) ligament forces are small enough to be ignored when the joint is in the middle of its range of motion and (2) the articular contact forces pass through the joint center of rotation, then equations 11.7 and 11.8 reduce to

HAB718 Spor Biyomekaniğinde Hareket Analizi



Musculoskeletal Modeling

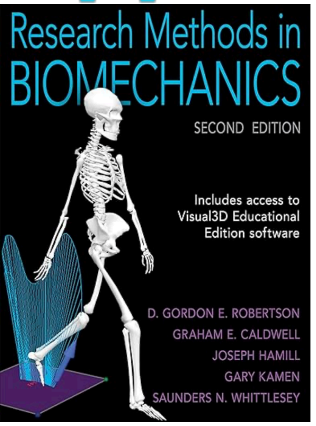
$$\vec{F}^J = \sum_{i=1}^{m} \vec{f}_i^m + \sum_{i=1}^{c} \vec{f}_i^c \tag{11.9}$$

$$\vec{M}^J = \sum_{i=1}^{m} \left(\vec{r}_i^m \times \vec{f}_i^m \right) \tag{11.10}$$

Equation 11.9 provides a basis for determining joint contact loads, if they are of interest in a particular application, but we will not discuss this further here. In solving the static optimization problem, we will use equation 11.10 as a constraint to ensure that the muscle forces we determine via static optimization reproduce the measured joint moment. In this context, equation 11.10 is known as an *equality constraint* and must be satisfied during the solution process. It is also common to place *boundary constraints* on the muscle forces, such that muscles only generate tensile forces and do not exceed a designated maximum value (f_i^0). The upper bound for each muscle is usually determined by multiplying muscle physiological cross-sectional area (A^m) by an assumed value for muscle-specific tension. The boundary constraints can be expressed as

$$0 \le f_i^m \le f_i^0 \tag{11.11}$$

Note that there are an infinite number of combinations of muscle forces $\vec{f}_i^m$ that will satisfy equation 11.10 for a particular net joint moment. For example, given the moment arm (r^m) values listed in table 11.1, the following three potential candidate solutions ($f_1^m = 500$ N, $f_2^m = 0$ N, $f_3^m = 0$ N), ($f_1^m = 0$ N, $f_2^m = 200$ N, $f_3^m = 0$ N), and ($f_1^m = 0$ N, $f_2^m = 0$ N, $f_3^m = 250$ N) all balance the net moment of 10 N·m. However, none of these three candidate solutions would appear to be physiologically reasonable, as only one of three synergistic muscles is selected. To achieve a distribution of forces among the three muscles, we will seek the solution that reproduces the



Musculoskeletal Modeling

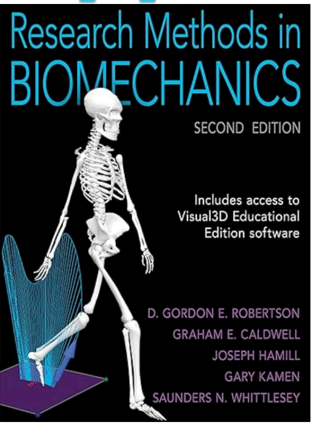
10 N·m joint moment while simultaneously minimizing the nonlinear function U, which can be expressed as

$$\text{Minimize } U = \sum_{i=1}^{3} \left(\frac{f_i^m}{A_i^m} \right)^3 \tag{11.12}$$

Equation 11.12 will serve as the cost function in the optimization problem. The quotient inside of the parentheses represents the individual muscle stresses, which are raised to the third power. If we assume that muscle forces are nonnegative, then the solution to equation 11.12 alone is (f_1^m = 0 N, f_2^m = 0 N, f_3^m = 0 N). However, if we require that the solution simultaneously satisfy equations 11.10, 11.11, and 11.12, then muscle forces greater than zero will be predicted in all muscles when the joint moment is non-zero.

Despite the relative simplicity of this example, obtaining an analytical solution by hand is quite challenging. Fortunately, any number of general purpose optimization algorithms can be used to obtain a numerical solution to equation 11.12, subject to the constraints expressed in equations 11.10 and 11.11. For this example, a commonly used technique known as sequential quadratic programming was used. The solution that was obtained was

$$f_1^m = 160.38 \text{ N}$$
$$f_2^m = 30.27 \text{ N}$$
$$f_3^m = 131.97 \text{ N}$$



Musculoskeletal Modeling

The reader can easily confirm that this solution balances the measured joint moment and none of the individual muscle forces fall outside of the specified bounds. The reader can also verify, if not exhaustively, that other solutions that satisfy the constraints, such as the three potential candidate solutions mentioned previously, result in greater values of the cost function *U*.

A common variant of the approach presented here has been to minimize the sum of squared, rather than cubed, muscle stresses. Solving this quadratic, rather than cubic, problem results in a qualitatively similar, but numerically different solution, which for the present example is (f_1^m = 151.04 N, f_2^m = 22.19 N, f_3^m = 146.74 N). If, however, the exponent in equation 11.12 is set to 1 (i.e., a linear cost function), then the solution is (f_1^m = 0 N, f_2^m = 0 N, f_3^m = 250 N), which is one of the unrealistic candidate solutions mentioned previously. In the quadratic case, the muscle forces are distributed across all three muscles but in a different manner than with the cubic cost function. In the linear case, the moment is borne entirely by the biceps brachii, which has the most favorable combination of moment arm and A^m. Recall that the rationale for using a cubic cost function was that it was supposed to result in a distribution of muscle forces that maximize endurance (Crowninshield and Brand 1981a). Although the quadratic and cubic solutions may appear to be reasonable, it is important to recall that the opportunities to validate muscle forces predicted via static optimization, or any other technique, have thus far been quite limited (e.g., Prilutsky et al. 1997).