6 Engineering Computing for Design [Incomplete]

Now that we have a solid foundation in programming and the manipulation of data, we can appreciate how computing can serve as an essential engineering tool. The primary engineering use of computing is to assist in the design of products. Design is a different paradigm than analysis. Analysis breaks a complex problem into manageable parts. Design, also called synthesis, constructs something new from several elements. However, the engineering design process includes analysis as a guide to optimizing its products.

How can we represent and solve design problems computationally? This chapter presents an engineering computing paradigm that answers this question. This will guide the design of our programs in the following chapters.

6.1 Design as Optimization

The early stages of the engineering design process involves the specification of quantitative **requirements** that must be satisfied by the product of the design. Once requirements have been specified, the design process can be considered as the selection of an element from a design space $\mathcal{D}$, comprised of mathematical models of various designs. Design requirements do not fully specify the design, however. Requirements constrain specific quantities of interest, which we call **output variables**; we call the space of all possible output variable quantities $\mathcal{O}$. A set of **governing equations** will determine how each design affects the output variables.

Governing equations relate **input variables**, known varying quantities like force and voltage, to output variables and include **parameters**, quantities like length, mass, and electrical resistance. Several designs will have the same set of governing equations E_i, differing only in their inputs and parameters. Let the corresponding set of designs be a **design set** and be denoted D_i, which is a subset of the design space $\mathcal{D}$. Let the solution of E_i for the output variables be denoted f_i and the output

image (i.e., all possible outputs) of f_i be called the **output set** and denoted O_i. This is represented in figure 6.1.

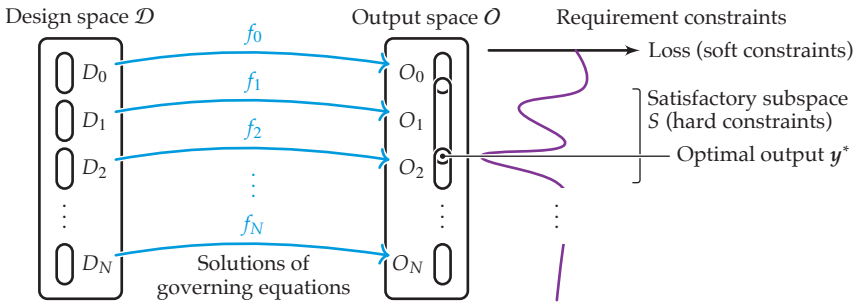

Figure 6.1. A visualization of the design process as the selection of an element from the design space $\mathcal{D}$ that is in a region of the output space O that meets the requirements.

Requirements are mathematical constraints, and there are three primary types:

- **Equality constraints** are requirements that specify that a given function of output variables must equal some number (e.g., the velocity must 4 m/s)
- **Inequality constraints** are requirements that specify that a given function of output variables must be greater than ($>$), lesser than ($<$), greater than or equal to ($\geq$), or lesser than or equal to ($\leq$) some number (e.g., the current must be positive)
- **Extremal constraints** are requirements that specify that a given function of output variables should be maximized or minimized (e.g., the length should be minimized)

Equality and inequality constraints are sometimes called **hard constraints** because they must be satisfied. The subspace of the output space O that contains outputs that satisfy the hard constraints we call the **satisfactory subspace** and denote S. Extremal constraints are sometimes called **soft constraints** because any value can satisfy the constraint, it's just that greater or lesser values are desirable. Extremal constraints are typically combined into a single scalar function called a **loss function** L. The **minimization** of the loss function L, subject to the hard constraints, results in a set of **optimal outputs** O^*. Let a vector[1] of output variables be denoted $\boldsymbol{y}$ and an optimal output vector $\boldsymbol{y}^*$.

1. If the output variables are themselves functions (e.g., of time), the "output vector" is in fact a vector-valued function.

Requirement constraints are illustrated in figure 6.1. The output sets O_1 and O_2 of two different design sets D_1 and D_2 overlap at the optimal output. Recalling that different design sets correspond to different sets of governing equations, we recognize that multiple types of design might be optimal. Suppose the illustrated optimal output $\boldsymbol{y}^*$ is known. Letting d_1 and d_2 represent designs in O_1 and O_2, and letting a design d_i^* be one corresponding to optimal output $\boldsymbol{y}^*$, there are two **inverse solutions** for optimal designs corresponding to $\boldsymbol{y}^*$,

$$d_1^* = f_1^{-1}(\boldsymbol{y}^*) \qquad \text{and} \qquad d_2^* = f_2^{-1}(\boldsymbol{y}^*).$$

This result implies several things about the design process:

- The design process includes an optimization over the requirement constraints
- The output space $\mathcal{O}$ is where the optimization is most clearly defined, but what the design process seeks an optimal design, which occurs in the design space $\mathcal{D}$
- Governing equations should be as general as possible (i.e., design sets should be as inclusive as possible)
- There is no guarantee that one or any design set D_i will include the most optimal output, so more design sets improve the chances that the corresponding output set O_i will include an optimal value
- Finding optima in the output space without consideration for the coverage of output sets in the output space can help guide design by providing target outputs, but it cannot tell us how to create a design that can achieve an optimal output
- If an optimum $\boldsymbol{y}^*$ occurs in output set O_i, the inverse map $f_i^{-1}(\boldsymbol{y}^*)$ gives a corresponding optimal design

However, these observations entail several complications. First of all, determining optimal outputs under constraints is no trivial task. The field of **constrained optimization** has developed algorithms for solving such problems, but these only guarantee solutions in cases under specific types of constraints. We will explore the constrained optimization problem in detail in chapter 5.

The second complication is that for a given design set D_i, it can be difficult to determine its corresponding output set O_i (i.e., the image of D_i through f_i). It is common for the solution f_i to the governing equations to be involved, so establishing its output set O_i is far from trivial. However, if the output set O_i is determined, a **local optimization** can be performed over O_i (instead of over $\mathcal{O}$) by expressing O_i as a collection of additional constraints. The results answer the question, "What are the optimal outputs for the design set D_i?" The answer helps us determine the best design for a given design set.

The third complication we call the **inverse problem**: the inverse f_i^{-1} of the solution f_i to the governing equations may not exist, and when it does it can be difficult

or intractable to compute. For a solution f_i with a known inverse f_i^{-1}, the design corresponding to any output $\boldsymbol{y}$ in O_i can be computed with $f_i^{-1}(\boldsymbol{y})$.

A design set D_i with determined output set O_i, a solved optimization over O_i yielding a local optimum output $\boldsymbol{y}_i^*$, and a determined inverse f_i^{-1} has a **local optimal design** d_i^* given by the expression

$$d_i^* = f_i^{-1}(\boldsymbol{y}_i^*).$$

6.1.1 The Structure of Design Sets

A design set D (we suppress the subscript here) is a mathematical model of a design, comprised of combinations of the following elements:

- A set E of governing equations
- A set U of all possible input variable vector $\boldsymbol{u}$ values
- A set P of all possible parameter vector $\boldsymbol{p}$ values

A design set D is the set of all pairs $(\boldsymbol{u}, \boldsymbol{p})$, where $\boldsymbol{u}, \boldsymbol{p}$ are vectors of input variable values from U and parameter values from P (i.e., $D = \{(\boldsymbol{u}, \boldsymbol{p}) \mid \boldsymbol{u} \in U \text{ and } \boldsymbol{p} \in P\}$) associated with a set of governing equations E. For a given input vector $\boldsymbol{u}$ and parameter vector $\boldsymbol{p}$, the set E of governing equations can perhaps be solved for the output variables in the vector $\boldsymbol{y}$, yielding a solution. There is no guarantee that such a solution is possible, but in many cases such a solution exists. If the equations of E are solved without specifying the input values of $\boldsymbol{u}$ or the parameter values of $\boldsymbol{p}$, the solution is f. Otherwise, only part of f is found. Note the value of leaving $\boldsymbol{u}$ and $\boldsymbol{p}$ unspecified: the solution for all inputs and all parameters is found.

6.1.1.1 Governing Equations A set of governing equations E relates input variables, parameters, and output variables. The equations are called "governing" because they express how these quantities relate. Many governing equations express physical theories (i.e., "laws") and phenomenologically established relations among the variables and parameters. Examples of governing equations include the following:

- Geometric relations (e.g., the area of a circle is π times the square of its radius)
- The conservation of energy (e.g., the energy in minus the energy lost is equal to the remaining energy)
- Newton's second law of motion, force is equal to the product of an object's mass and acceleration
- Ohm's law, that for a resistor the ratio of voltage across and current through it is constant

Governing equations often include variables that are not inputs, parameters, or outputs. Such variables are, like output variables, unknown, but are not themselves

output variables (i.e., the requirement constraints do not specify what they should be).

6.1.1.2 Input Variables Input variables are variables are quantities that vary with some independent variable, usually time and sometimes space, and enter the set of governing equations E. What distinguishes them from other variables in E is that they are considered to be known. For instance, input variables can be:

- The voltage supplied by a battery
- A loading loading force on a structure
- The number of units of a part supplied per minute to a manufacturing machine

In some cases, input variables can be chosen by the designer; we call these **adjustable input variables**. We treat the selection of adjustable input variables as part of the design process. Adjustable input variables often have practical hard constraints (e.g., they can be power limited), called **input variable constraints**. These will be taken into account during the optimization step.

6.1.1.3 Parameters A parameter is a constant quantity (i.e., it is not a dependent variable) in the set of governing equations E. However, a parameter potentially has different (constant) values in the context of a design set D. For instance, for a resistor with voltage v and current i, Ohm's law gives

$$v = iR,$$

where R is a parameter, specifically, the resistance. The other quantities v and i are usually taken to be (unknown) time-varying variables.

Some parameters cannot be changed by the designer; those that can be changed are called **adjustable parameters**. We treat the selection of adjustable parameters as part of the design process. Adjustable parameters often have practical hard constraints, called **parameter constraints**. These will be taken into account during the optimization step.

6.1.2 Solving a Set of Governing Equations

In theory, the solution f of the governing equations requires that the output variables be determined. In practice, it is usually just as easy to solve for *all unknown variables* in E, and then select the subset of these that are outputs. This implies two key features of E:

- The number of independent equations in E should be greater than or equal to the number of unknowns (i.e., there should exist at least one solution)
- The equations in E should include all output variables

A solution f maps any design set D element $d = (\boldsymbol{u}, \boldsymbol{p})$, for $\boldsymbol{u}$ in U and $\boldsymbol{p}$ in P, to a corresponding output $\boldsymbol{y}$ in O. This is visualized in figure 6.2.

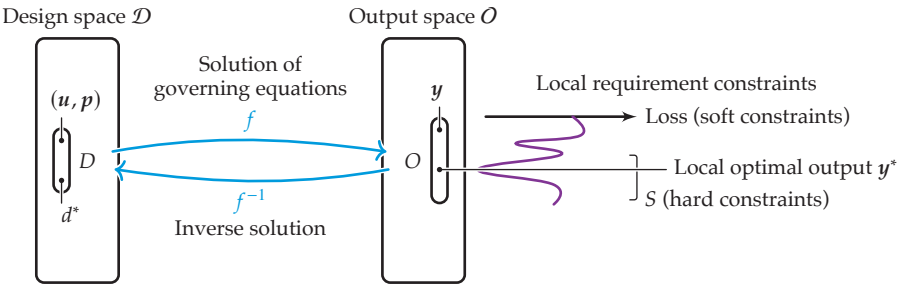

Figure 6.2. A single design set D, the solution f of its set E of governing equations, and the inverse solution f^{-1}. The local optimal design $d^* = f^{-1}(\boldsymbol{y}^*)$.

For some types of sets of governing equations (e.g., linear), f can be found in a general way, meaning for all inputs $\boldsymbol{u}$ and/or parameters $\boldsymbol{p}$. Such solutions are carried out **symbolically**. Computing tools for this are described in chapter 4.

For other types of sets of governing equations, a solution for f is difficult or intractable. For such sets, several approximate **specific solutions** $\tilde{f}$ for the outputs given specific inputs and parameters are found **numerically**. Computing tools for numerical solutions are described in chapter 5. Note, however, that even with a set $\widetilde{F}$ of specific solutions $\tilde{f}$, a full solution f is lacking. This implies we can know neither the extent of the output image O, nor all the possible loss values in O. Therefore, it is always preferable to have a symbolic solution for f than to have a set of specific solutions $\tilde{f}$. However, one nice feature of $\tilde{f}$ is that elements of its inverse[2] are known for free because if some design d maps to output $\tilde{f}(d)$, then we already know $d = \tilde{f}^{-1}(\tilde{f}(d))$.

6.1.3 Solving for the Inverse Solution of a Set of Governing Equations

An inverse solution f^{-1} of the governing equations maps an output $\boldsymbol{y}$ to a design $d = (\boldsymbol{u}, \boldsymbol{p})$. That is, f^{-1} maps an output vector $\boldsymbol{y}$ to the input vector $\boldsymbol{u}$ and parameter vector $\boldsymbol{p}$ that yielded $\boldsymbol{y}$. An inverse solution may not exist or the reverse mapping is not unique (i.e., it is not one-to-one), in which case multiple designs yield the same output. In some such cases, a pseudoinverse solution exists that can map to a single design that yields the output.

For some forward maps f, the corresponding inverse f^{-1} can be found in general, for all $\boldsymbol{y}$ in O the corresponding design $d = (\boldsymbol{u}, \boldsymbol{p})$ is determined. Such inverse solutions are carried out symbolically. Computing tools for this are described in chapter 4.

2. Here we ignore the possibility that $\tilde{f}$ may have no inverse. However, if we allow $\tilde{f}^{-1}$ to be a pseudoinverse (i.e., as a nonunique reverse mapping), a design d is still found.

As with a forward solution f, it may be difficult or intractable to find an inverse solution f^{-1}. Several approximate **specific inverse solutions** $\tilde{f}^{-1}$ for the design given a specific output are found **numerically**. Computing tools for numerical solutions are described in chapter 5. These are the same tools that are used to numerically solve for forward solutions.

6.1.4 Local Constrained Optimization in Terms of Adjustables

The loss function $L(\boldsymbol{y})$ is constructed from the outputs with soft constraints. The form of L is chosen according to the following considerations:

- An output variable that is to be minimized should reduce L for smaller values; conversely, one that is to be maximized should reduce L for larger values
- The relative scaling of each output variable should reflect its relative numerical magnitude (e.g., if one is numerically large compared with another, the latter might be scaled up in L)
- The relative scaling of each output variable should reflect its design importance (i.e., an important output variable should have greater scaling)
- Simpler forms of L (e.g., linear or quadratic) tend to be easier to solve for optimal output values

Optimizing L in terms of outputs yield globally optimal outputs. These optima are good targets for the design process, and are in that sense can be instructive. However, global optima often cannot be achieved by the design sets developed. Therefore, we also perform local optimization for each design set by minimizing the loss L over the adjustable input variables and adjustable parameters, which we call **adjustables**. Local optimization in terms of adjustables is subject to the following constraints:

- Soft requirement (output variable) constraints (via L)
- Hard requirement (output variable) constraints
- Hard input variable constraints
- Hard parameter constraints

The input variable and parameter constraints are already expressed in terms of adjustables. However, the requirement constraints must be transformed into adjustables. Transform requirement constraints into adjustables by substituting the solution f into L (soft constraints) and into the hard constraints.

Minimize L in terms of adjustables. A resulting optimum in terms of adjustables corresponds to a local optimal design d^*. By transforming the requirement constraints to the adjustables, we have effectively moved the local optimization problem to the design set D. This is ideal because a result of the optimization yields an optimal design d^* instead of an optimal output $\boldsymbol{y}^*$, which would require the

inverse mapping $d^* = f^{-1}(\boldsymbol{y}^*)$ be computed. Therefore, transforming the local constrained optimization to the adjustables allows us to find optimal designs without a solution f^{-1} to the inverse problem, which may not exist or may be intractable to compute.

Example 6.1

Suppose we are designing a 3D printer extruder that melts a plastic filament and deposits it on the print. Based on the customer's demands, the following output variables have been identified:

- T (K), the temperature at the hot end of the extruder (greater than room temperature)
- r (m/s), the filament feed rate
- t_C (s), the time to cool the hot end to near ambient temperature

The following design requirements have been identified in terms of the output variables:

- $T = 200$ K
- r should be maximized (for faster prints)
- t_C should be minimized (for faster cooldown)

Based on the requirements, an initial design has been developed with a resistive heater. A sketch of the design is shown in figure 6.3.

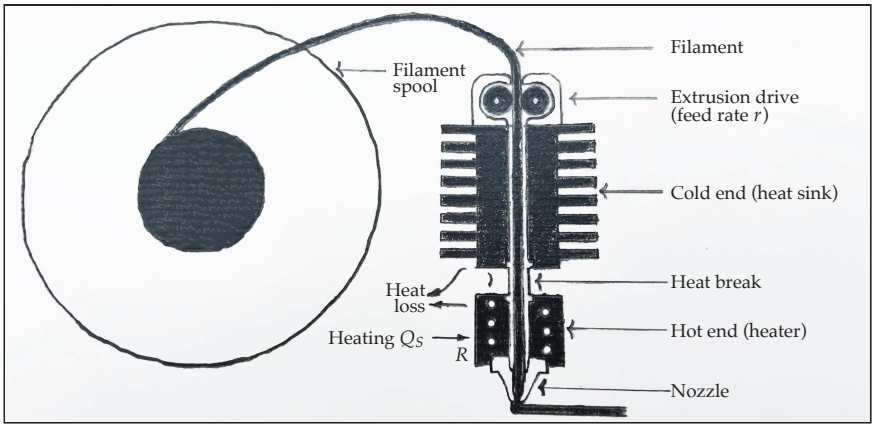

Figure 6.3. Sketch of the extruder design.

The corresponding design set has been developed as follows. The input variables are as follows:

- r, the filament feed rate (also an output), adjustable
- $v = 12$ V, the voltage supplied to the heater at the hot end of the extruder, nonadjustable

The following parameters have been identified:

- R_T (K/W), the thermal resistance to heat dissipation, adjustable, with lower bound $R_T \geq 1$ K/W
- $C_T = 0.5$ W·s/K, the thermal capacitance of the hot end, nonadjustable
- $R = 20\ \Omega$, the electrical resistance of the heater, nonadjustable
- $c_p = 1590$ J/(kg·K), the specific heat capacity of the filament
- $\rho = 1240$ kg/m^3, the density of the filament
- $\delta = 1.75 \cdot 10^{-3}$ m, the diameter of the filament

The following set E of equations can be developed to describe the thermal and electrical system interactions in terms of input and output variables:

$$E = \left\{ \begin{array}{ll} T = \dfrac{1}{1/R_T + 1/R_f} Q_S & \text{(steady-state temperature)} \\ R_f = \dfrac{4}{\pi c_p \rho \delta^2 r} & \text{(filament heating)} \\ t_c = 5\tau & \text{(cooldown time, sufficient)} \\ \tau = R_T C_T & \text{(cooldown time, time constant)} \\ Q_S = vi & \text{(resistive heater, power)} \\ v = iR & \text{(resistive heater, Ohm's law)} \end{array} \right\}$$

where the following intermediate variables have been introduced:

- Q_S (W), the resistive heating
- R_f (K/W), the thermal resistance to heat transfer from the hot end to the filament
- τ (s), the thermal time constant for the hot end
- i (A) the electrical current supplied to the resistive heater

Solve for an optimal design d^*.

We will assume that the filament is fed at a constant rate r.

Defining the Design Set The output variable vector $\boldsymbol{u}$ can be defined in terms of the output variables as

$$\boldsymbol{y} = \begin{bmatrix} T & R & t_C \end{bmatrix}^\top.$$

Similarly, the input variable vector $\boldsymbol{u}$ can be defined in terms of the input variables:

$$\boldsymbol{u} = \begin{bmatrix} r & v \end{bmatrix}^\top.$$

Finally, we can define the parameter vector $\boldsymbol{p}$ in terms of the parameters:

$$\boldsymbol{p} = \begin{bmatrix} R_T & C_T & R & c_p & \rho & \delta \end{bmatrix}^\top.$$

The design set D is the set of all pairs $(\boldsymbol{u}, \boldsymbol{p})$.

Solving the Set of Governing Equations The solution f of the set E of governing equations for the output variables can be found algebraically by eliminating the intermediate variables. This yields the following solution:

$$\boldsymbol{y} = \begin{bmatrix} T \\ r \\ t_C \end{bmatrix} = f(d) = \begin{bmatrix} \dfrac{1}{1/R_T + \pi\delta^2 c_p \rho r/4} \dfrac{v^2}{R} \\ r \\ 5R_T C_T \end{bmatrix}.$$

Note that the solution for output r is just r because it is also an input. The solution is expressed entirely in terms of input variables from $\boldsymbol{u}$ and parameters from $\boldsymbol{p}$.

Local Constraint Optimization We must now construct a loss function $\boldsymbol{L}(\boldsymbol{y})$ in terms of output variables. The variables for the two soft constraints, maximizing r and minimizing t_C, will be in L. We begin by constructing a simple linear combination of r and t_C,

$$L(\boldsymbol{y}) = a_r r + a_{t_C} t_C,$$

where a_r and a_{t_C} are real constant coefficients we will now choose. Because we are maximizing r, we want great values of r to correspond to lesser values of L (i.e., less loss); therefore, we should choose $a_r < 0$. Conversely, because we are minimizing t_C, we should choose $a_{t_C} > 0$.

Now consider the magnitudes of a_r and a_{t_C}. First, we select a middling value for each of r and t_C by suggesting that a feed rate of $r = 1 \cdot 10^{-2}$ m/s and a cooldown time of $t_C = 10$ s are reasonable. Note that this r is much smaller than this t_C, so if $|a_r|/|a_{t_C}| \approx 1$ then the loss will be dominated by t_C. This suggests an initial ratio $|a_r|/|a_{t_C}| = 10^3$. That's to even the playing field, if you will. Now consider the relative importance of the feed rate and cooldown constraints. Suppose the feed rate r is twice as important as the cooldown time t_C. We can express this in the loss function by increasing the ratio to $|a_r|/|a_{t_C}| = 2 \cdot 10^3$. The total magnitude of the loss is unimportant, so we can set either coefficient arbitrarily. Let our final choice of coefficients be

$$a_r = -2 \cdot 10^3 \text{ s/m} \quad \text{and} \quad a_{t_C} = 1 \text{ s}^{-1}.$$

Therefore, the loss function is

$$L(\boldsymbol{y}) = -2 \cdot 10^3 \, r + t_C.$$

Note that we have made the loss function dimensionless.

To optimize $L(\boldsymbol{y})$ in terms of output $\boldsymbol{y}$ is simple. Clearly, the greater r is and the lesser t_C is, the lesser loss L is, which doesn't add much to our understanding. With what we know, the hard constraint $T = 200$ K doesn't help us restrict the other outputs.

To perform a local optimization, we use the solution f to transform the requirement (output) hard constraint and soft constraints (loss function) such that they are expressed in terms of the adjustables r and R_T. The hard requirement constraint can be transformed as follows:

$$T = 200 \quad \xrightarrow{f} \quad \frac{1}{1/R_T + \pi\delta^2 c_p \rho r/4}\frac{v^2}{R} = 200 \quad \Rightarrow \quad r = \frac{4}{\pi\delta^2 c_p \rho}\left(\frac{v^2}{200R} - \frac{1}{R_T}\right).$$

This equality relates two output variables, r and R_T, effectively constraining them to each other. Transforming the soft requirement constraints via applying f to $L(\boldsymbol{y})$,

$$L(d) = -2 \cdot 10^3\, r + 5R_T C_T.$$

This version of the loss function can be solved for the optimal adjustables r and R_T, subject to the hard constraints

$$r = \frac{4}{\pi\delta^2 c_p \rho}\left(\frac{v^2}{200R} - \frac{1}{R_T}\right) \quad \text{and}$$

$$R_T \geq 1\ ,$$

where the latter is a parameter constraint. The equality constraint can be applied directly to $L(d)$, yielding

$$L(d) = \frac{8 \cdot 10^3}{\pi\delta^2 c_p \rho}\left(\frac{1}{R_T} - \frac{v^2}{200R}\right) + 5R_T C_T.$$

There is only a single adjustable remaining, R_T. To solve for the minimizing thermal resistance R_T^*, differentiate L with respect to R_T and set it equal to zero, as follows:

$$\frac{dL}{dR_T} = 0 = \frac{-8 \cdot 10^3}{\pi\delta^2 c_p \rho R_T^*} + 5C_T.$$

Solving for R_T^*,

$$R_T^* = \frac{40}{\delta\sqrt{\pi c_p \rho C_T}} \approx 13.0\ \text{K/W}.$$

To check that this is a minimum, we plot $L(d)$ in figure 6.4.

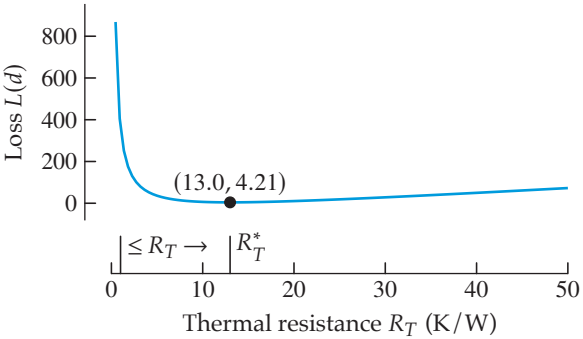

Figure 6.4. The constrained loss function $L(d)$ versus a single adjustable, the thermal resistance R_T.

Note that the hard parameter constraint $R_T \geq 1$ K/W is satisfied by R_T^*. From the equality constraint, this corresponds to an optimal feed rate of $r^* = 14.1 \cdot 10^{-3}$ m/s. In summary, the optimal design d^* is

$$d^* = (\boldsymbol{u}^*, \boldsymbol{p}^*) = \left(\begin{bmatrix} r^* \\ v^* \end{bmatrix}, \begin{bmatrix} R_T^* \\ C_T^* \\ R^* \\ c_p^* \\ \rho^* \\ \delta^* \end{bmatrix} \right) \approx \left(\begin{bmatrix} 14.1 \cdot 10^{-3}\ \text{m/s} \\ 24\ \text{V} \end{bmatrix}, \begin{bmatrix} 13.0\ \text{K/W} \\ 0.5\ \text{W} \cdot \text{s/K} \\ 20\ \Omega \\ 1590\ \text{J/(kg·K)} \\ 1240\ \text{kg/m}^3 \\ 1.75 \cdot 10^{-3}\ \text{m} \end{bmatrix} \right).$$

This corresponds to a locally optimal output of

$$\boldsymbol{y}^* = f(d^*) = \begin{bmatrix} T^* \\ r^* \\ t_C^* \end{bmatrix} \approx \begin{bmatrix} 200\ \text{K} \\ 14.1 \cdot 10^{-3}\ \text{m/s} \\ 32.5\ \text{s} \end{bmatrix}.$$

If, in the loss function L, we had weighted the cooldown time t_C more, then t_C^* would have been shorter and r^* would have been slower. This type of tradeoff among soft constraints is common.

It is possible that another design with another set of input variables and parameters could achieve a lower loss than $L(d^*) = 4.21$. However, we have achieved an optimal design d^* given the single design set given here.

6.2 Representing Design Constraints