# Modern Nonlinear Optimization Techniques for an Optimal Control of System Dynamics Models

I. Vierhaus[1], A. Fügenschuh[2], R. Gottwald[1], S. Grösser[3]

[1] Zuse Institute Berlin, Germany

[2] Helmut Schmidt University / University of the Federal Armed Forces Hamburg, Germany

[3] Bern University of Applied Sciences, Bern, Switzerland

**Abstract:**
We study System Dynamics models with several free parameters that can be altered by the user. We assume that the user's goal is to achieve a certain dynamic behavior of the model by varying these parameters. In order to the find best possible combination of parameter settings, several automatic parameter tuning methods are described in the literature and readily available within existing System Dynamic software packages. We give a survey on the available techniques in the market and describe their theoretical background. Some of these methods are already six decades old, and meanwhile newer and more powerful optimization methods have emerged in the mathematical literature. One major obstacle for their direct use are tabled data in System Dynamics models, which are usually interpreted as piecewise linear functions. However, modern optimization methods usually require smooth functions which are twice continuously differentiable. We overcome this problem by a smooth spline interpolation of the tabled data. We use a test set of three complex System Dynamic models from the literature, describe their individual transition into optimization problems, and demonstrate the applicability of modern optimization algorithms to these System Dynamics Optimization problems.

**Keywords:**

System Dynamics; Optimal Control; Nonlinear Optimization; Spline Interpolation.

# 1 Introduction

System dynamics models describe the behavior of dynamically complex social systems that consists of several interrelated stocks, flows, and feedback loops. System dynamics models are mainly developed, first, to understand existing dynamics in a system, and second, to improve current policies, e.g., policies to improve the impact of projects [1], the energy-efficiency in the residential built environment [2], city development [3], or in innovation management [4]. System dynamics models consist of ordinary differential equations, nonlinear functional relations, and tabular data. Even if each of the elements in such a system is individually well understood, the interplay of several of such elements shows a surprising, unexpected behavior over time [5]. Given that one objective of system dynamics model is to enhance current policies, we need to discuss how such an improved policy can be identified. Operationally it is simple to perform different policies: simulation runs with varying input parameters can be used to test and evaluate different policies. When two different runs lead to different dynamic behaviors of the system, one asks, which of the two simulations is more adequate or "better"? To answer this question properly, one needs to define an objective function. The objective function needs to be of such a kind that the higher the simulated value for the objective function, the better the policy [6]. In this way, two different simulation runs become comparable. Given that an objective function has been defined sufficiently, the next step is to identify the parameter configuration that results in the best-possible objective function, i.e., the best policy possible. This is what is called "policy optimization". To identify the optimal policy, an optimization algorithm has to do be deployed which can identify a local or even proven global optimal policy for a given situation.

In this article, we survey the existing approaches that are found in System Dynamics software packages on the market. Two of the vendors use methods from the mid-60s, and that are able to find local optimal solutions for optimization problems with just a very few number of free variables (or parameters). Recently, mathematical optimization algorithms were developed that go far beyond this limit, and that were already been tested on optimization problems having several thousand free variables and constraints. We demonstrate the applicability of these methods on three System Dynamics models as test cases. As a technical difficulty, tabled data (i.e., piecewise linear functions) that are used to describe parts of the models needs to be approximated by smooth curves that are twice continuously differentiable, in order to apply modern optimization techniques. Here we use spline interpolation. Finally, numerical results for these three test cases are presented and discussed.

## 2 Optimization Approaches in Professional Software Packages

We contacted a number of software providers for System Dynamics simulation software, and asked about further information on the optimization capabilities of their respective software packages. In the following, we present their replies.

**Vensim.** The Vensim software uses a derivative of the basic Powell algorithm [7]. This algorithm works on the basis of a power hill climb. It can be used for policy optimization or for model calibration. Also it can be complemented by a Kalman filter.

**Powersim.** Optimization is a built-in feature of Powersim Studio. The user has to define the model variable that represents the target key performance indicator, together with the variables which are the assumptions that can be varied and define the valid range they may assume. The optimization routine allows the user to fine-tune values before the analysis starts. After the analysis, Powersim Studio presents the combination which is the closest to the objective. No further information on the mathematical background of the used method is provided.

**isee systems: Stella.** At present, Stella does not have an integrated optimization capability. Currently, the software producer is working on the development of this function which should debut within the year 2014. One can perform calibration and some kinds of optimization using a subroutine called PEST (Parameter Estimation) from a third-party vendor (Echo Valley Graphics, Inc.).

**Goldsim.** GoldSim uses Box' Method [8], which can optimize a "moderate number" of decision variables in a dynamic simulation model. For this type of simulation, the user specifies an objective function, i.e., a specific result that shall be minimized or maximized, an optional constraint, and one or more optimization variables, i.e., variables in the model that one has control over. GoldSim then simulates the model multiple times, systematically selecting combinations of values for each of the optimization variables. By this, GoldSim can determine the values of the optimization variables that optimize (minimize or maximize) the objective function while meeting the specified constraint.

**Forio.** Forio does not support optimization processes. Forio rather focuses on the display of models and how they can be designed. It offers to upload models from Vensim, Powersim, Stella or Excel.

**Anylogic.** Anylogic offers an optimization capability. This capability is built on the top of what the vendor calls the "OptQuest Optimization Engine". The engine finds automatically the "best" parameters of a model, with respect to a certain constraints. The OptQuest Optimization Engine uses metaheuristic, mathematical optimization, and neural network components to guide the search for best solutions to decision and planning problems of all types. The user-written application tells OptQuest Engine the quality of each solution generated during the search, by calling an evaluator, such as an objective function, that can take any form. It also provides support for multi-objective optimization.

**Dynaplan Smia.** Smia supports optimization at two levels: the global level and the local level. Global optimization will adjust selected independent variables to minimize or maximize the value of an objective. Local optimization is used in combination with dynamic simulation to mimic the optimization processes performed by human or automatic decision makers when making decisions for the next period during a simulation. No further information on the mathematical background of the used method is provided.

**Insight Maker.** Insight Maker includes built-in optimization functionality to automatically adjust one or more parameter values to achieve a desired goal, where also Powell's method is used [7]. The search algorithm as described is purely deterministic. The algorithm will find a single minimum value that, if multiple minima exist, may or may not be the global minimum. Many optimization techniques address this issue by introducing stochasticity into the optimization process, e.g. simulated annealing or genetic algorithms. Given convex problems, however, such stochasticity may reduce efficiency in finding the minimum. As such, the simulator uses the deterministic algorithm as described, but deals with local minima by offering random starting locations for the search pattern.

## 3 Past and Present Nonlinear Optimization Techniques

Five decades of mathematical research in optimization methods have passed since the work of Powell [7] and Box [8]. State of the art optimization techniques base on a totally different methodology. From this the question arises, if those more recent methods can be used for parameter optimization in System Dynamics models. In the following, we start with an overview of the two classical methods of Powell and Box, since they are still used in recent SD software packages. Then we describe a modern algorithmic ideas of Wächter and Biegler [9] and Drud [10, 11] that we use for solving our models.

The method of Powell [7] finds a locally optimal solution (not a global one), but for unconstrained optimization problems only, which are of the following form:

$$\begin{aligned} \max \quad & f(x_1, \dots, x_n) && \text{(1a)} \\ \text{s.t.} \quad & x_1, \dots, x_n \in \mathbb{R}. && \text{(1b)} \end{aligned}$$

There are no further assumptions made on function $f$, in particular, it does not need to be a convex or a differentiable function. The method starts from an initial approximation of an optimal solution $x^0$. In each iteration, one tries to find an improving point by performing a line search along $n$ linearly independent search directions $\xi^1, \dots, \xi^n$. Initially, these search directions are the coordinate vectors, and thus in the beginning one modifies one of the variables in $x^0 = (x^0_1, \dots, x^0_n)$ at a time. In latter steps of the method the search directions are modified, such that the search directions are still linearly independent, but the search is not limited to coordinate vectors only, see Figure 1. Powell's method is usually applicable to small scale problems with up to 20-50 variables.

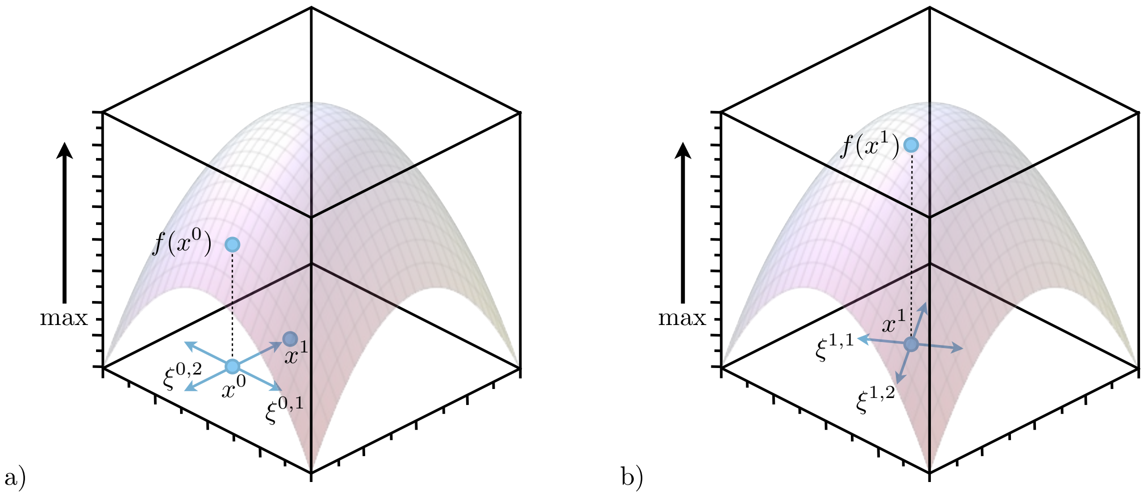


Figure 1: Search directions of Powell's method a) at the beginning (coordinate vectors), and b) at a later step.

The method of Box [8] is based on the notion of a simplex or complex. The basic iteration of Box' method is shown in Figure 2. Given is an optimization problem of the following form:

$$\begin{aligned} \max \quad & f(x_1, \dots, x_n) && \text{(2a)} \\ \text{s.t.} \quad & g_k(x_1, \dots, x_n) = 0, \quad \forall\, k = 1, \dots, m, && \text{(2b)} \\ & x_1, \dots, x_n \in \mathbb{R}. && \text{(2c)} \end{aligned}$$

In contrast to Powell's method, the method of Box is able to deal with constraints. Here $x_1, \dots, x_n$ are free variables, $f$ is the objective function, and $g_k$ are the constraint functions. No assumptions on the smoothness of $f$ and $g$ are made, hence they can be non-differentiable or non-continuous functions. It is only assume that the feasible region $\{(x_1, \ldots, x_k) \in \mathbb{R}^n : g_k(x_1, \ldots, x_n) \leq 0\}$ is a convex set. In Box' method, an initial number of at least $n+1$ points $x^1, \ldots, x^k \in \mathbb{R}^n, k \geq n+1$ is sought that all fulfill the constraints $g_k(x^i) \leq 0$ for all $i = 1, \ldots, k$. Initially, these points are found using a pseudo-random generator that adds some noise to one initial solution $x^0$. The objective function $f$ is evaluated for each point $x^1, \ldots, x^k$. The object $(x^1, \ldots, x^k)$ is called a complex. The point with the worst (smallest) objective function value is removed from the complex, and reflected at the centroid of the remaining points. The reflection is $\alpha \geq 1$ times as far from the centroid as the worst point, where $\alpha$ is a user-defined parameter. If the trial point is not feasible (i.e., violated one of the $g_k \leq 0$ conditions), then some repair mechanisms take place. The procedure terminates, if the new points do not improve the objective function value over a certain number of rounds. Note that the method does not require smoothness of the constraints and objective function, or derivates. Hence it can easily be applied to SD models with only discrete data (from tabled data). Further note that the optimal solution is not a global, but a local optimal solution only. There is no guarantee that this method fulfills optimality conditions, hence in a mathematical sense, it might not even by a locally optimal solution. The problem sizes that can be handled with this method are usually rather small, around 20-50 variables and about the same number of constraints.

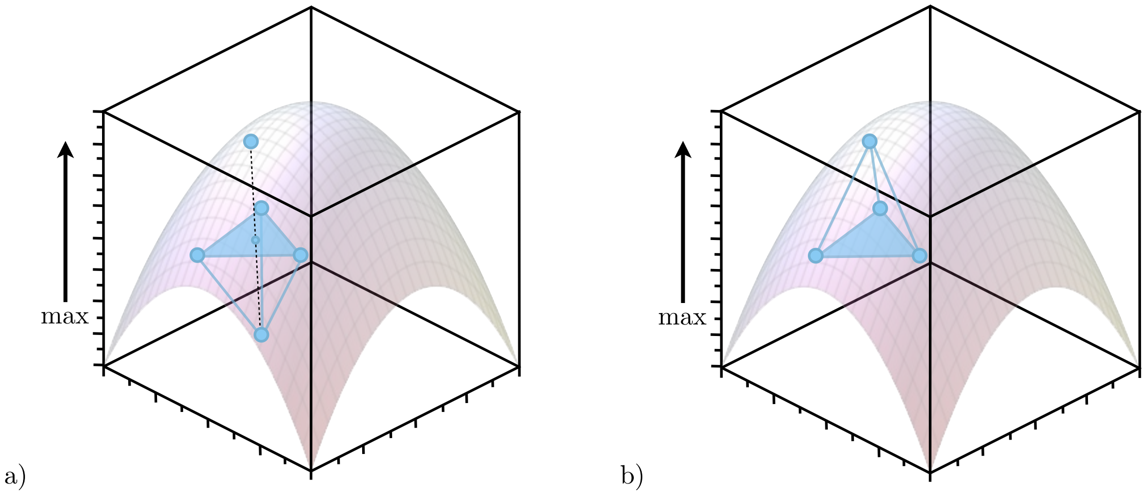


Figure 2: One iteration of Box' method; a) the point with the worst objective function value is reflected to define a new complex b).

In our work, and as an example for a modern optimization method, we make use of interior-point line-search algorithms, that were described by Wächter and Biegler [9] or Drud [10,11]. It also deals with constrained optimization problems of the form (2). The method is based on the Karush-Kuhn-Tucker (KKT) conditions [12,13], a necessary condition for local optimality of constrained optimization problems. These conditions yield a nonlinear equality system, which is solved multiple times during an optimization run for a parameter that converges to zero. When being (approximately) zero, a local optimal solution is found, see Figure 3. This method, however, makes use of first and second derivate informations, hence it must be assumed that all functions involved are at least twice continuously differentiable. As a positive effect of this, it can be applied to large problem sizes. Wächter and Biegler implemented their method in the publicly available software code IPOPT, and applied it to a test set of problems, where the largest has about 250,000 variables and constraints. A similar implementation based on a generalized reduced-gradient algorithm is available from Drud through the software code CONOPT, which we primarily use for our optimization computations, because our experiments revealed that CONOPT is more suitable for optimization problems with an inherent dynamic structure.

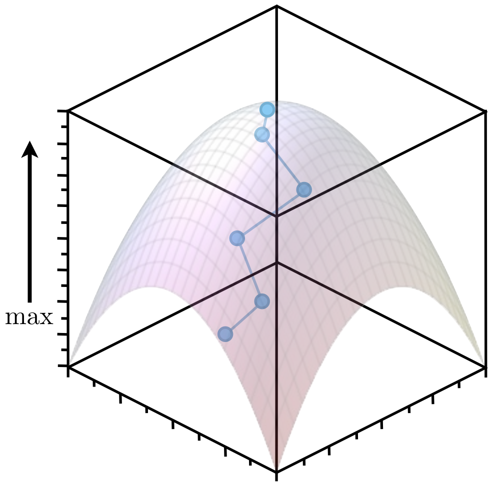


Figure 3: The central path converging to a (local) optimal solution in the method of Wächter and Biegler.

# 4 Test Instances

In the following, we describe three standard System Dynamics models which we use to demonstrate our optimization approach. Since the models are covered extensively in existing literature and are standards in the field of system dynamics, we introduce them only briefly in this section. For each of the models, we will also set up an optimization problem for each of the models.

## 4.1 Market Growth Model

The market growth model, one of the classical models in System Dynamics, has the objective to identify and to explain "sales stagnation, or even sales decline, to arise out of an overly cautious capital investment policy" [14]. The model structure (Figure 4) shows the interdependency of the company's backlog and the sales force on the one hand, and the capacity extension on the other hand. It is the different adjustment times of the stocks of sales force and production capacity which complicate managing the system of stocks. Figure 5 shows that the development of order booked is interrupted, and even reduced, around month 30, 62 and 100. This occurs since the delivery time for shipping this product increases so significantly that customers refrain from ordering the product. The growth prospects of this product are limited since expanding the production capacity cannot match the demand. What initiatives are necessary to improve the performance of business system?

### Optimization

To investigate the possibilities of the company to alter the systems behavior, we formulate an optimization problem with the goal to avoid the collapse of the companies business, that is shown in the base run. This is represented by the objective function $J_m$, which is defined as follows:

$$J_m = -\int_t \text{Production capacity}(t)\,\mathrm{d}t \tag{3}$$

We set up two optimization scenarios, as summarized in Table 1. In the first scenario, we conduct a simple parameter optimization. In the second scenario, we are using time dependent controls, i.e., the parameter that is optimized as a time dependent control, can assume a different value at each time step.

## 4.2 World2

Forrester [15] developed the World2 model to address the evolving challenges in the 1970s which the Club of Rome identified [16]. The model mapped important interrelationships between the stocks world

|  | Scenario 1 |  | Scenario 2 |  |
|---|---|---|---|---|
| Variable | Interval | Mode | Interval | Mode |
| Time to hire salesmen | $[10, 40]$ | Parameter | $[10, 40]$ | Control |
| Production delay time | $[6, 24]$ | Parameter | $[6, 24]$ | Control |
| Time to set delivery delay goal | $[6, 24]$ | Parameter | $[6, 24]$ | Control |

Table 1: Optimization scenarios for Market Growth.

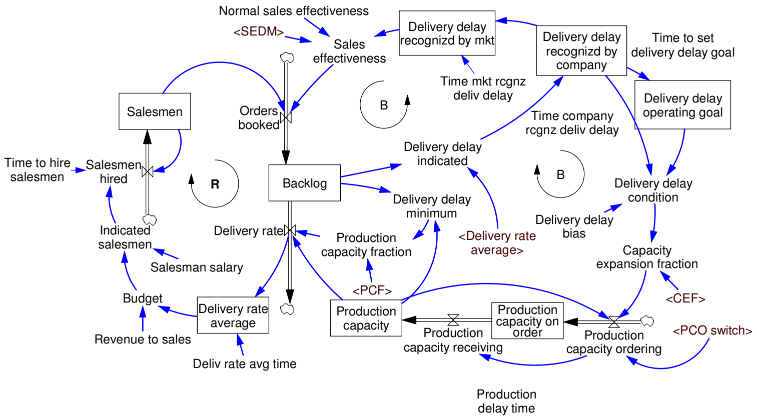


Figure 4: Market growth model [14].

population, food, resources, industrial production, and pollution (Figure 6). The simulation results of the model shows a collapse of the world socio-economic system sometime during the twenty-first century, if no interventions are undertaken to lessen the demands on the earth's carrying capacity (Figure 7). The figure shows that with an increasing population, the demand for food and resources continuously increase which, first, slowly depletes the available natural resources, and second, result in increasing pollution. Both due to a lack of food resources and due to hazardous pollution, the total population starts to decline in about 2020. At the end of the 21st century, the population level might reach a level which is twice the amount of the population at the beginning of the 19th century. At the same time, the available resources have been reduced significantly. The model has also been used to identify policy changes capable of moving the global system to a fairly high-quality state that is sustainable far into the future.

## Optimization

Again we formulate an optimization problem to analyze possible measures to change the systems behavior. Our goal is to maximize the integral over the quality of life for the considered time window of 200 years. This is represented by the objective function $J_m$, which is defined as follows:

$$J_w = \int_{\frac{3}{4}T}^{T} \text{Pollution}(t)\,\mathrm{d}t - \int_{\frac{3}{4}T}^{T} \text{quality of life}(t)\,\mathrm{d}t - \int_{\frac{3}{4}T}^{T} \text{Population}(t)\,\mathrm{d}t \qquad (4)$$

Again we set up one parameter optimization and one control scenario, as summarized in Table 2.

|  | Scenario 1 |  | Scenario 2 |  |
|---|---|---|---|---|
| Variable | Interval | Mode | Interval | Mode |
| Capital Investment Rate Norm | $[0.02, 0.1]$ | Parameter | $[0.02, 0.1]$ | Control |
| Birth rate normal | $[0.01, 0.08]$ | Parameter | $[0.01, 0.08]$ | Control |

Table 2: Optimization scenarios for World2.

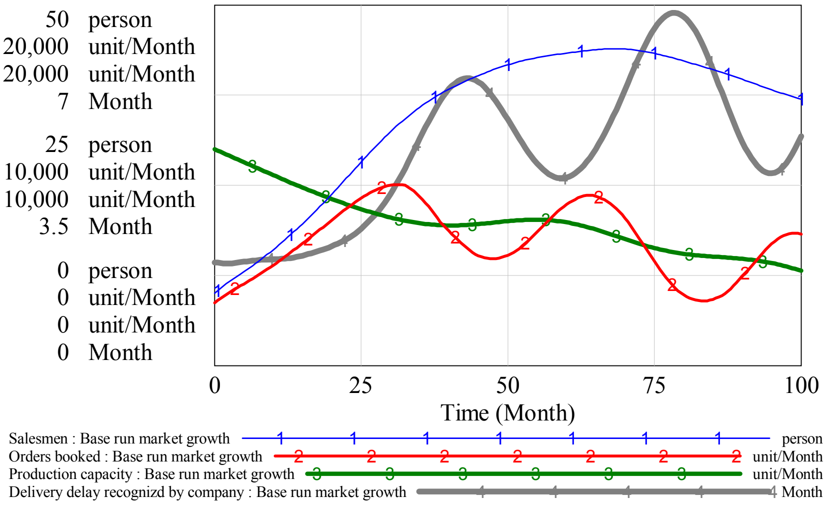
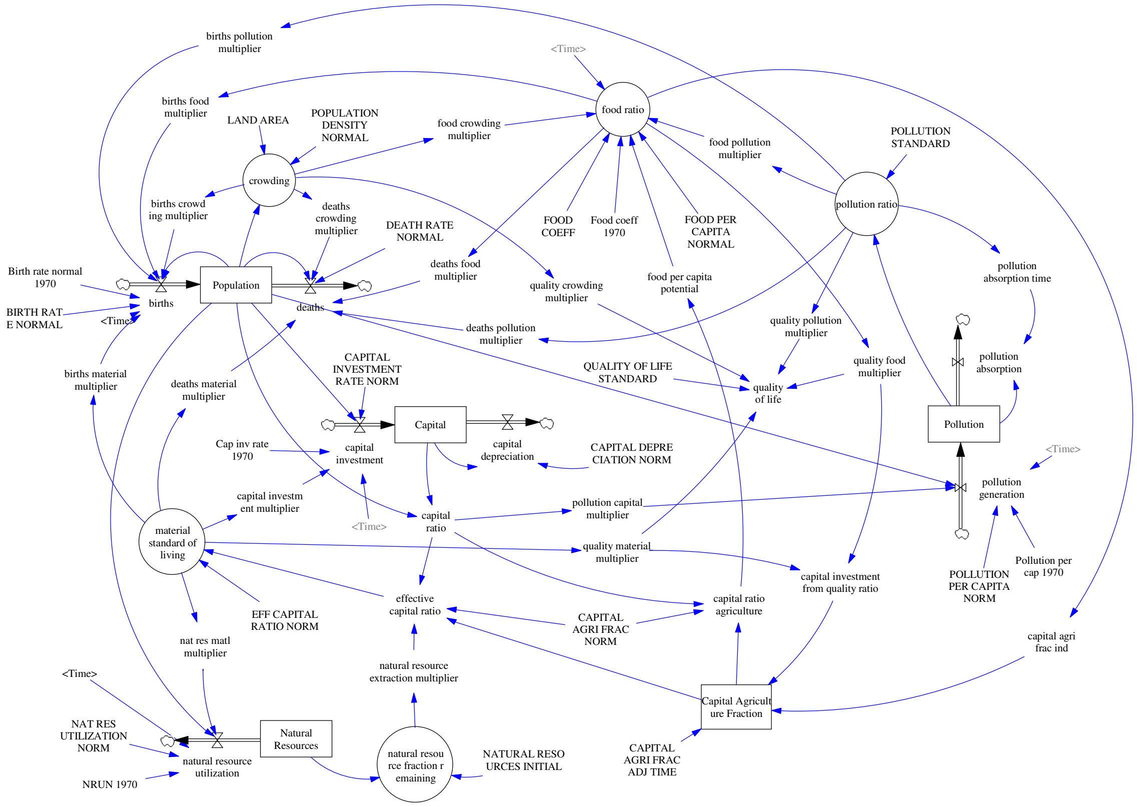


Figure 5: Base run simulation of the market growth model.



births pollution multiplier
<Time>
births food multiplier
LAND AREA
POPULATION DENSITY NORMAL
food crowding multiplier
food ratio
food pollution multiplier
POLLUTION STANDARD
crowding
births crowd ing multiplier
deaths crowding multiplier
DEATH RATE NORMAL
FOOD COEFF
Food coeff 1970
FOOD PER CAPITA NORMAL
pollution ratio
pollution absorption time
Birth rate normal 1970
births
Population
deaths
deaths food multiplier
quality crowding multiplier
food per capita potential
BIRTH RATE NORMAL
<Time>
deaths pollution multiplier
quality pollution multiplier
quality food multiplier
pollution absorption
births material multiplier
deaths material multiplier
CAPITAL INVESTMENT RATE NORM
QUALITY OF LIFE STANDARD
quality of life
Capital
Pollution
Cap inv rate 1970
capital investment
capital depreciation
CAPITAL DEPRE CIATION NORM
<Time>
capital investm ent multiplier
<Time>
capital ratio
pollution capital multiplier
pollution generation
material standard of living
quality material multiplier
capital investment from quality ratio
Pollution per cap 1970
POLLUTION PER CAPITA NORM
effective capital ratio
capital ratio agriculture
EFF CAPITAL RATIO NORM
CAPITAL AGRI FRAC NORM
nat res matl multiplier
capital agri frac ind
natural resource extraction multiplier
<Time>
Capital Agriculture Fraction
NAT RES UTILIZATION NORM
Natural Resources
natural resource utilization
natural resou rce fraction r emaining
NATURAL RESO URCES INITIAL
CAPITAL AGRI FRAC ADJ TIME
NRUN 1970

Figure 6: World2 model [17].

## 4.3 Project Dynamics Model

The third model is on project dynamics (Figure 8). Richardson and Pugh [18] develop an advanced model to understand the ripple effects occurring in large-scale projects (see Lyneis and Ford [19] for a later version). The essential aspect in the model is the differentiation of the completed tasks in real progress rate, i.e., that tasks are completed with high quality and not additional work is required, and the generation of undiscovered rework, i.e., completed of tasks which have flaws and have to be reworked, but

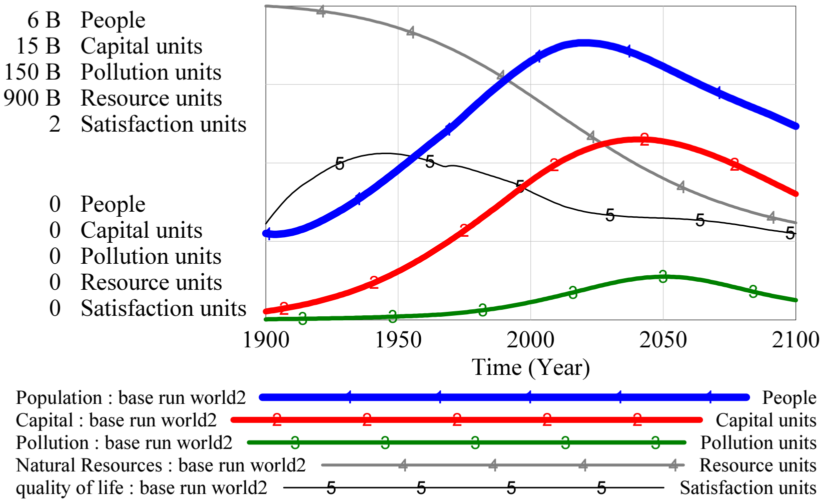


Figure 7: Base run simulation of the World2 model.

| Variable | Scenario 1<br>Interval | <br>Mode | Scenario 2<br>Interval | <br>Mode |
|---|---|---|---|---|
| Time to Perceive Productivity | $[1, 12]$ | Parameter | $[1, 12]$ | Control |
| Workforce Adjustment Time | $[1, 8]$ | Parameter | $[1, 8]$ | Control |

Table 3: Optimization scenarios for the Project Dynamics model.

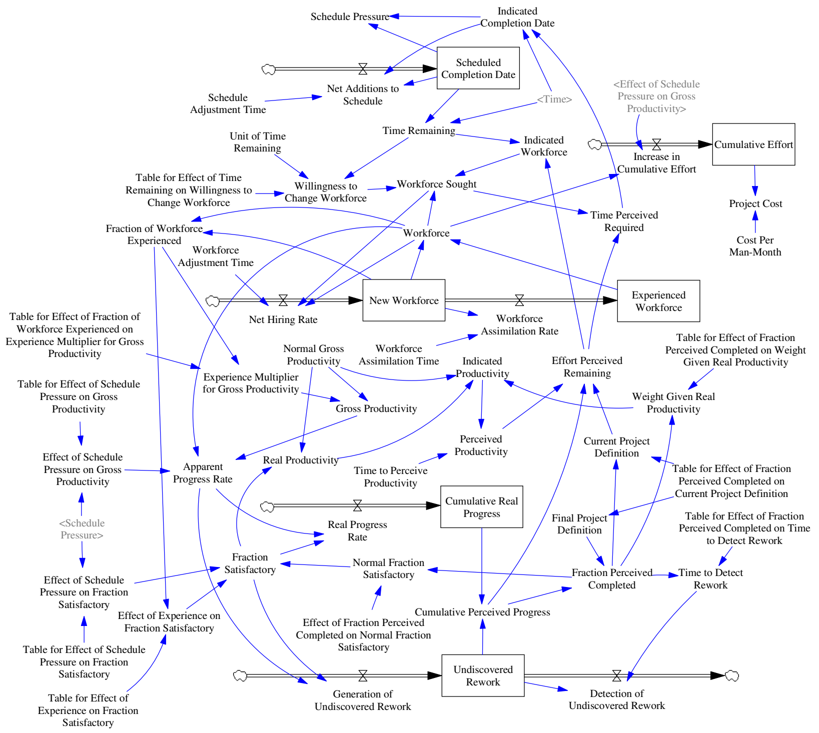
are yet not discovered as flawed. The project managers, however, only perceive the cumulated perceived progress which adds both stocks "undiscovered rework" and "cumulative real progress". Hence, a systematic misperception occurs which is the basis for all management interactions, e.g., change of the completion date, hiring of additional workforce. These management actions cause secondary effects, i.e., ripple effects. For instance, expanding the workforce leads to a dilution of the workforce's average experience resulting in a lower productivity and higher failure. The model's behavior (Figure 9) shows that productivity only increases slowly as the project advances over time. In addition, initially the undiscovered rework increases and peaks around month 38. Significant resources have to be used to perform the rework tasks and to finish the project. The result is that overall project costs increase significantly and that the estimated complete (month 30) date slips by an additional 30 months. What policies have to be changed to avoid such destructive dynamics?

### Optimization

As before, we formulate an optimization problem to analyze possible external measures to change the systems behavior. In this model, we have several goals in order to optimize the project management. We aim to keep the final project cost, the additions to schedule as well as the undiscovered rework low, while maximizing the productivity of the workforce. These goals can be written in terms of the following objective function:

$$J_p = \text{Project Cost}(T) + \int_t \text{net additions to schedule}(t)\,\mathrm{d}t + \int_t \text{undiscovered rework}(t)\,\mathrm{d}t - \int_t \text{Real Productivity}(t)\,\mathrm{d}t \tag{5}$$

As before we set up two optimization scenarios. The parameters are listed in Table 3.



Schedule Pressure
Indicated Completion Date
Scheduled Completion Date
Schedule Adjustment Time
Net Additions to Schedule
<Time>
<Effect of Schedule Pressure on Gross Productivity>
Time Remaining
Unit of Time Remaining
Indicated Workforce
Increase in Cumulative Effort
Cumulative Effort
Table for Effect of Time Remaining on Willingness to Change Workforce
Willingness to Change Workforce
Workforce Sought
Project Cost
Time Perceived Required
Fraction of Workforce Experienced
Workforce
Cost Per Man-Month
Workforce Adjustment Time
New Workforce
Experienced Workforce
Net Hiring Rate
Workforce Assimilation Rate
Table for Effect of Fraction of Workforce Experienced on Experience Multiplier for Gross Productivity
Normal Gross Productivity
Workforce Assimilation Time
Indicated Productivity
Effort Perceived Remaining
Table for Effect of Fraction Perceived Completed on Weight Given Real Productivity
Table for Effect of Schedule Pressure on Gross Productivity
Experience Multiplier for Gross Productivity
Gross Productivity
Weight Given Real Productivity
Perceived Productivity
Current Project Definition
Effect of Schedule Pressure on Gross Productivity
Apparent Progress Rate
Real Productivity
Time to Perceive Productivity
Table for Effect of Fraction Perceived Completed on Current Project Definition
Cumulative Real Progress
<Schedule Pressure>
Real Progress Rate
Final Project Definition
Table for Effect of Fraction Perceived Completed on Time to Detect Rework
Fraction Satisfactory
Normal Fraction Satisfactory
Fraction Perceived Completed
Time to Detect Rework
Effect of Schedule Pressure on Fraction Satisfactory
Effect of Experience on Fraction Satisfactory
Cumulative Perceived Progress
Effect of Fraction Perceived Completed on Normal Fraction Satisfactory
Table for Effect of Schedule Pressure on Fraction Satisfactory
Undiscovered Rework
Generation of Undiscovered Rework
Detection of Undiscovered Rework
Table for Effect of Experience on Fraction Satisfactory

Figure 8: Project dynamics model [18].

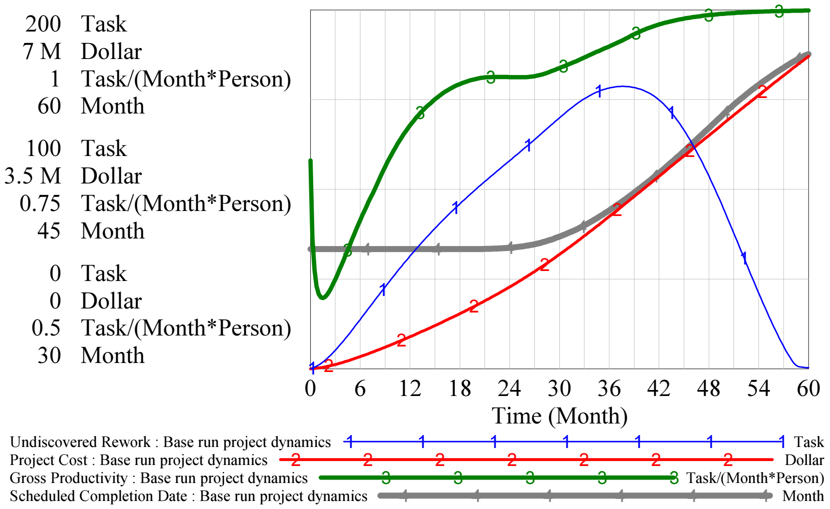


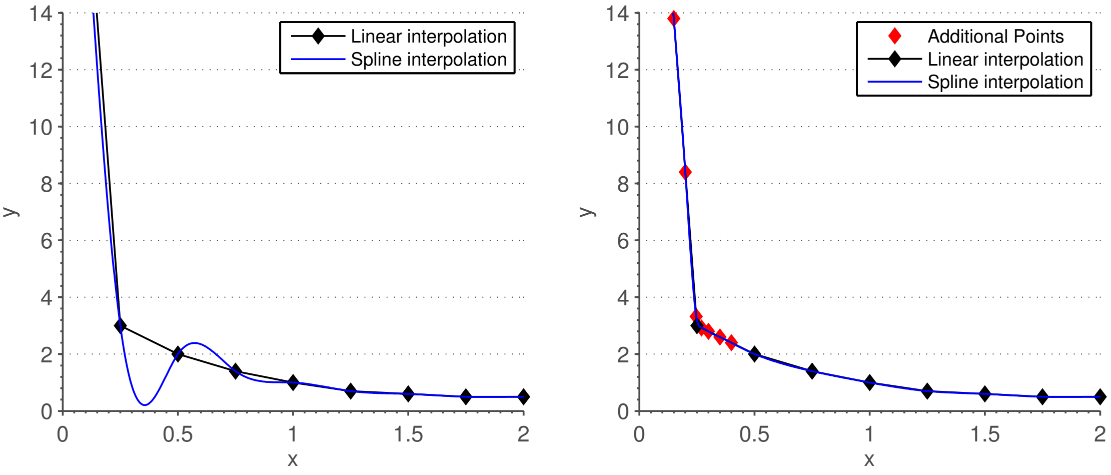
Figure 9: Base run simulation of the project dynamics model.

# 5 Interpolation of Tabled Data

Tabled data can be found in most system dynamics models, as a means to represent measured data, where it is not clear a priori, if the data can be described in terms of a mathematical formula. For use in simulations, the linear interpolation between discrete data points is an easy way to produce the expected behavior. However, as described in Section 3, for modern local optimization solvers to work efficiently, first and second derivatives of the model functions must be available. The derivatives of a piecewise linear function are undefined at the discrete data points, where two linear curves with different slopes meet, so that solvers like IPOPT or CONOPT cannot be used to solve system dynamics optimization problems as they are. However, it stands to reason that any set of discrete data points that is likely to appear in practice will stem from some continuous and in most cases smooth distribution of data. The non-smooth functions that are used to represent them, must then be considered an *artifact* of the representation, and not an inherent property. We argue, that data that appears in system dynamics models can almost always be represented by a differentiable mathematical function without influencing the qualitative behavior of the model. This can be done by hand, but can also be successfully automated. In this paper we follow a spline interpolation approach.

For our purposes only interpolation methods that yield functions with continuous first and second derivatives are eligible. One method which meets this requirement is polynomial interpolation. While some of the tabled data can be approximated very well by polynomials of a reasonable degree, this method is prone to oscillation. In order to obtain a good approximation in all cases we decided to interpolate the data by piecewise polynomial functions that are also referred to as splines. Since we require continuity up to the second derivative, a natural choice is to use cubic spline interpolation, i.e., to find a cubic polynomial for each interval such that the first and second derivatives match at the data points. The continuity condition and the condition to interpolate the data points do not uniquely determine the cubic spline and hence, additional conditions at the boundaries have to be specified. Possible choices are to require a zero second derivative which gives the *natural* cubic spline interpolant or to set explicit values for the first derivative which gives the *complete* cubic spline interpolant. These and other interpolation and approximation methods are described comprehensively by de Boor [20].

For the boundary conditions we chose to follow the latter approach, because it yields a closer approximation to the piecewise linear interpolation which is used, for instance, by Vensim for the tabulated data. Visual inspection of the resulting cubic splines validated that they fit most of the tabulated data very well. In a few cases adding additional data points significantly improved the approximation as illustrated by Figure 10.



a) b)

Figure 10: Tabled data from World 2 model a) with strong deviation b) with additional data points

However, in most cases the cubic spline interpolant was sufficiently close to the piecewise linear interpolant without any modifications. Thus, it is very unlikely that these approximations negatively affect the qualitative behavior of the models and if nevertheless this approach is suspected to be insufficient, better smooth approximations may be obtained by utilizing more sophisticated curve fitting algorithms.

## 5.1 Example market growth

To evaluate the effect of the smooth data interpolation on simulation runs, we compare the values of *Delivery delay minimum* and *Production capacity fraction* in the base run of the market growth model. In the original model, the the two variables are connected through the equation

$$\text{Production capacity fraction} = PCF(\text{Delivery delay minimum}) \tag{6}$$

where PCF is the lookup function that we replace. Figure 11 a) shows the smooth interpolation compared with the piecewise linear interpolation. At all data points, the two functions take the same value by construction. Between the data points, the deviation between linear and spline interpolation is very small. To visualize the effect of replacing the piecewise linear interpolation of data with the smooth representation,

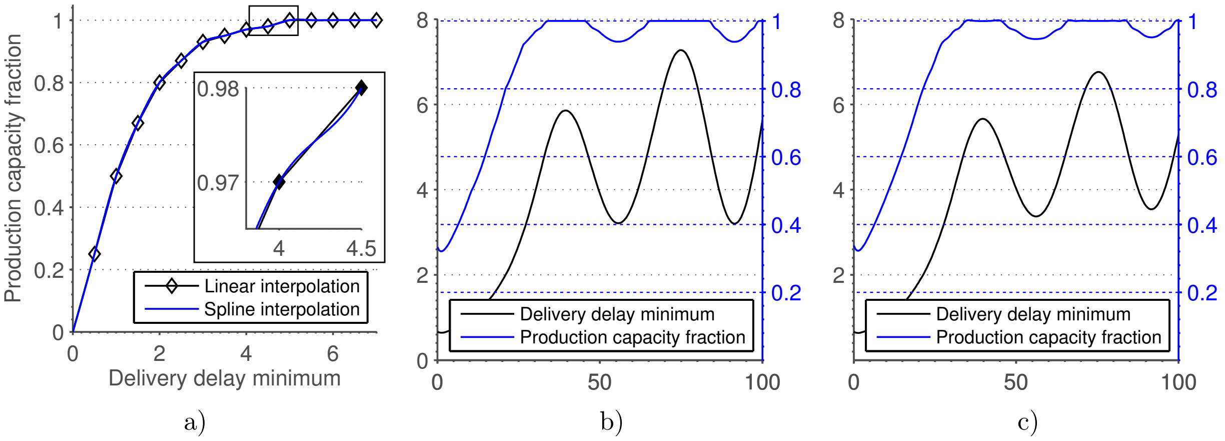


Figure 11: a) Linear and sigmoid interpolation of the data. b) Simulation with Linear Interpolation. c) Simulation with smooth interpolation of all data.

Figure 11 b) shows the value of the ”Production capacity fraction” together with its only cause ”Delivery delay minimum” for one simulation conducted with the original lookups, and Figure 11 c) shows one simulation conducted with the differentiable representation of the tabled data. We see identical qualitative behavior, and very similar quantitative behavior. In the simulation with smooth data, the *Delivery delay minimum* takes a slightly large value at the second maximum. We conclude from this comparison, that a smooth representation of data is equally suitable to create the desired qualitative and quantitative model behavior.

# 6 Computational Results

In this section, we shortly summarize our approach to optimization with a modern NLP solver. We then proceed to discuss solutions of the optimization scenarios introduced in Section 4.

## 6.1 Writing the Optimization Problems as Nonlinear Programs

In order to solve our optimization problems with a modern NLP solver, we need to write the models in the standard form of Nonlinear Programs (NLPs). A nonlinear program consists of a set of variables and a set of linear and nonlinear constraints. In order to define the variables and constraints, we must first decide on a discretization scheme, and a time frame for the simulation. We choose a simple Euler Scheme in all cases, and a step size and time horizon for the optimization as used in the simulation base run. A summary of the parameters for all models can be found in Table 4.

In the Project Dynamics Model, there is a special complication to be taken into account. The model contains variable *Time remaining*, which appears in the denominator of the definition of *Indicated Workforce*. If the chosen time windows exceeds the time it takes the project to be finished, the *Time remaining* reaches zero, leading to a division by zero. To avoid this, we choose a variable step size for our optmization, and added an additional constraint, making sure that the step size is such that the *Time remaining* is below 0.1 at the end of the time window, i.e. such that the project es just finished.

|  | Market Growth | World2 | Project Dynamics Model |
|---|---|---|---|
| Unit | Month | Years | Month |
| Step size ($\Delta$) | 1/2 | 1/4 | $[0.1, 0.4]$ |
| Time window | $0 - 100$ | $1900 - 2100$ | $[24, 96]$ |
| Final time ($T$) | 100 | 2100 | $[24, 96]$ |
| Time steps ($n_T$) | 200 | 800 | 240 |

Table 4: Optimization parameters for the test instances.

After deciding on discretization and the time frame, we need to transcribe the model into an NLP , see Füegenschuh and Vierhaus [21] for more details. We denote the vector of state variables by $x$ the vector of auxiliary or algebraic variables by $y$ and the vector of control variables by $z$. We consider the value of each variable at each of the discrete time steps as one discrete variable: $x(t) \to x_0, x_1, \ldots, x_n, y(t) \to y_0, y_1, \ldots, y_n$. Then we can express the differential and algebraic equations of the model in terms of these variables, resulting in one constraint for each model equation and each time step. For the chosen Euler discretization method, this yields an NLP of the form

$$\min J(x, y, z), \tag{7a}$$

$$\text{s.t. } x_{j+1} = f(x_j, y_j, z_j), \quad j \in \{0, 1, \ldots, n_T - 1\}, \tag{7b}$$

$$y_j \quad = g(x_j, y_j, z_j), \quad j \in \{0, 1, \ldots, n_T\}, \tag{7c}$$

$$\underline{x}_j \leq x_j \leq \overline{x}_j, \quad j \in \{0, 1, \ldots, T\}, \tag{7d}$$

$$\underline{y}_j \leq y_j \leq \overline{y}_j, \quad j \in \{0, 1, \ldots, T\}, \tag{7e}$$

$$\underline{z}_j \leq z_j \leq \overline{z}_j, \quad j \in \{0, 1, \ldots, T\}, \tag{7f}$$

which is a special case of the constrained optimization problem defined in equation (2).

In all of the optimization runs described in the following section, we initialized all variable values with the solutions of the base run simulation in order to provide a feasible starting point for the solver.

## 6.2 Market Growth Model

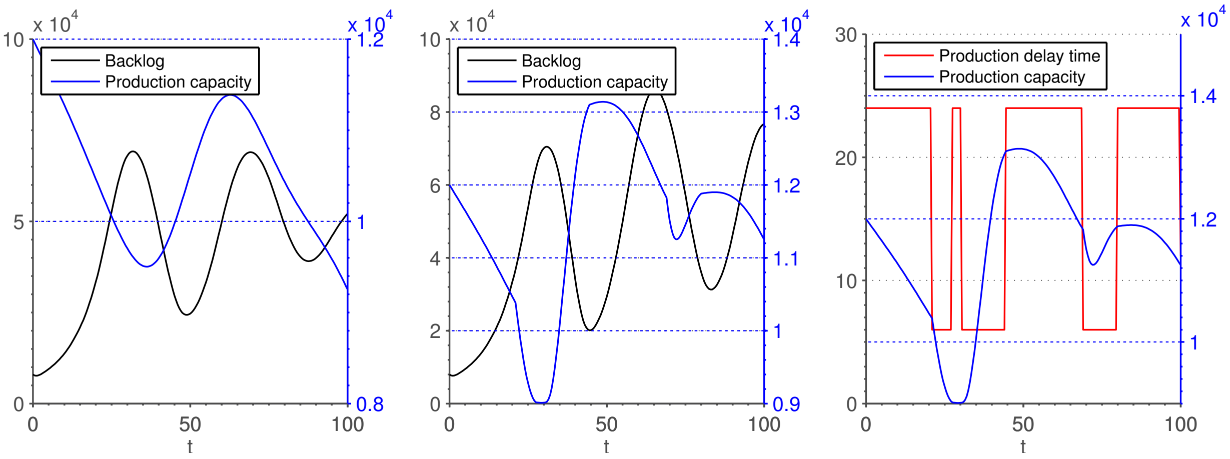
In the base run, the simulation shows that mounting backlogs and delays lead to a successive reduction of the companies production capacities. In the end of the base run, the production capacity has fallen to almost zero. Our optimization goal is to find a course of action, which allows to avoid this result.

The two optimization scenarios we will consider are summarized in Table 1. We begin with the pure parameter optimization in scenario 1. The free variables in the optimization are parameters to the delays in the model, i.e. we allow the company to vary the delays of hiring additional salesmen and adjusting to changing production demand as well as adjusting the production time.

The optimization problem is solved in under three seconds. The solution values for all three parameters are at the bounds: The ”Time to hire salesmen” takes the lower bound value of 10. The ”Production delay time” as well as the ”Time to set delivery delay goal” take the upper bound value of 24. The resulting development of *Backlog* and *Production capacity* is shown in Figure 12 a).

Even though we were able to achieve a significantly better objective value as listed in Table 5, there is no qualitative change of the behavior compared to the base run. Since two of the three delay parameters are increased, the period of oscillations in *Production capacity* and *Backlog* is also slightly increased. We still see however, that the production capacity decreases significantly from start to the end of our time window. The amplitude of the oscillations does not decrease over time, indicating that even with the optimized parameters, the reduction of the *Production capacity* to zero will eventually occur.

We now solve the second scenario, with three time dependent controls. Again, the solver converges in under three seconds. Again, we achieve a significant improvement of the objective value. The resulting development of *Backlog* and *Production capacity* for this solution are shown in Figure 12 b). The development of the *Backlog* is similar to the solution of scenario 1, showing an oscillation with a gradual increase



a) b) c)

Figure 12: Solutions for the market growth model from parameter and control optimization. a) Variable values for scenario 1. b), c) Variable and control values in scenario 2.

of the mean value. The development of the production capacity has however change qualitatively. The period of the oscillations is now close the that of the *Backlog*. The second minimum in production capacity is reached within the considered time window, and the decrease in production capacity is much smaller. At the point of the last maximum close to end of the time window, the *Production capacity* takes on a value close to the initial value.

Figure 12 c) shows the *Production capacity* together with one of the control variables. The control changes drastically over time, switching between the imposed upper and lower bound. The plot shows the correlation between the values of the control and the *Production capacity*. When reaching the maximum at $t \approx 40$, the control takes a high value, to slow down the decrease in production capacity. Close to the minimum in *Production capacity*, the delay is minimized, allowing the company to react more quickly to the increased backlog. The amplitude of the oscillation in the *Production capacity* decreases significantly indicating that after a few oscillations, a steady state will be reached. The solution can be considered an active countering of the oscillations imposed by the model structure.

## 6.3 Project Dynamics Model

We first applied the optimization approach described in the previous sections to solve the "Parameter Optimization" scenario for the Project Dynamics model (Table 3, scenario 1). The solution values for both parameters are at the bounds, the "Workforce Adjustment Time" takes the lowest possible value of 1, while the "Time to Perceive Productivity" assumes the upper bound of 12.

In the second scenario, each value was allowed to assume a different value inside the given interval at each discrete point in time. Figure 13 compares the resulting trajectories for two selected variables and shows the controls in scenario 2 over time.

Table 5 summarizes running times and objective values for the base run and all three scenarios. The parameter optimization in scenario 1 was able to reduce the objective value by of $1.5 \cdot 10^5$. By considering the optimization parameters as time dependent controls, a further improvement of $1.3 \cdot 10^5$ was possible. However, Figure 13 shows clearly, that in neither of the scenarios a qualitatively different behavior of the model could be achieved. The structure of the model governs the quick development of "Undiscovered Rework". At least within the parameter bounds chosen for the conducted optimization runs, this could not be avoided.

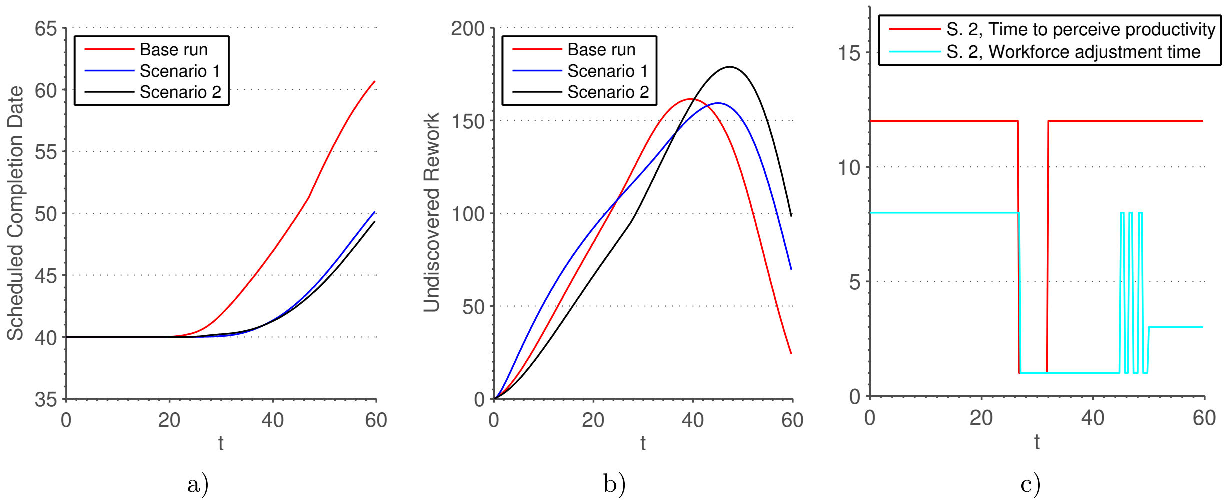


Figure 13: Comparison of the base run and both optimization scenarios a) Scheduled completion date, b) undiscovered rework, c) solution for the controls in scenario 2.

| | Base run | Scenario 1 | Scenario 2 |
|---|---|---|---|
| Market Growth | $-1.67 \cdot 10^6$ | $-2.11 \cdot 10^6$ | $-2.31 \cdot 10^6$ |
| Project Dynamics Model | $5.33 \cdot 10^6$ | $5.18 \cdot 10^6$ | $5.05 \cdot 10^6$ |
| World2 | $-1.19 \cdot 10^2$ | $-1.30 \cdot 10^6$ | $-2.83 \cdot 10^2$ |

Table 5: Summary of objective values

## 6.4 World2

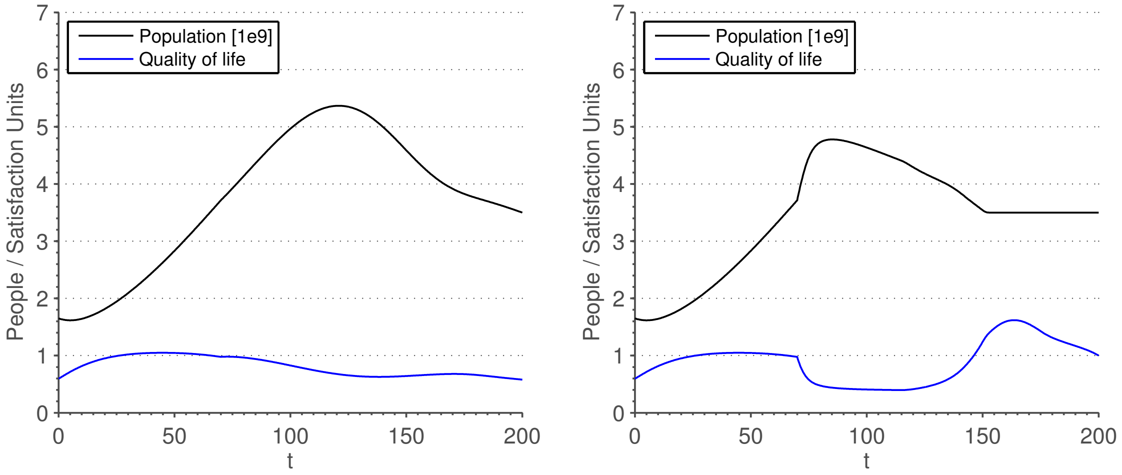
The dynamical behavior of the World2 model shows after the simulation run with default parameters a decline of the population due to an overuse of the natural resources and a growing pollution. The quality of life also reduces in later years, see Figure 7.

In the optimization model we selected two parameters as control variables, the Birthratenormal1970 and the Capinvrate1970. Both values are effectively changing the outcome of the simulation after the year 1970. The optimization goal is to maximize the quality of life and the size of the population, while minimizing the pollution. As a further constraint, the population should not fall below 3.5 in the last half of the time horizon.

In the parameter optimization problem (with constant control) the Birthratenormal value raises from 0.04 (before 1970) to 0.08 (after 1970), and the Capinvrate drops from 0.05 (before 1970) to 0.02 (after 1970). These two measures are sufficient to significantly reduce the pollution and lower the population size towards the desired target level of 3.5 at the end of the time horizon, see Figure 14 a).

In the control optimization problem (with parameters varying over time), the Birthratenormal and the Capinvrate almost show the same behavior. The difference to the previous optimization with constant controls is negligible. Hence there is no use of the additional degree of freedom in order to better control the behavior of the World2 model, see Figure 14 b).

As an interpretation of these results in the context of the application, one could say that the only way to prevent the population from a collapse offered by our modified World2 optimization model is to transform a highly developed economy and society (one with low birthrate and a high capital investment rate) into a lesser developed one (with high birthrates and low capital investment rates).



a)

b)

Figure 14: a) Optimization scenario 1, b) Optimization scenario 2

# 7 Conclusions

Mathematical optimization techniques for the numerical solution of nonlinear problems have advanced over the last years, so that they can be used to solve even more complex System Dynamics Optimization models. The only technical obstacle is to approximate tabled data by smooth functions, for which we suggest spline interpolation. After that, numerical codes such as CONOPT are able to solve SDO problem to (local) optimality in very short time, typically less than one minute. Besides parameter optimization, which is currently used in System Dynamics software packages, it is also possible to compute optimal control functions, that is, parameter functions that vary over time. Having such solutions at hand offers a more detailed view on the dynamical behavior of the model and its controllability.

The solutions found by the nonlinear optimization techniques are only locally optimal, and no clue is given how far they are away from a global optimal solution. To answer this question, it is necessary to embed the whole procedure into a branch-and-bound search that guarantees to find global optimal solutions in finite time. Our recent research (see also Fügenschuh and Vierhaus [21]) is heading in this direction.

# Acknowledgement

We gratefully acknowledge the funding of this work by the German Research Association (Deutsche Forschungsgemeinschaft, DFG), Collaborative Research Center CRC1026 (Sonderforschungsbereich SFB1026). We thank all software providers for sharing information and thus supporting our research.

# References

[1] Ford, D. N., Sterman, J. D., 1998. Dynamic modeling of product development processes. System Dynamics Review, 14(1):31–68.

[2] Groesser, S. N., 2012. Co-Evolution of Standards in Innovation Systems: The Dynamics of Voluntary and Legal Building Codes. Williams and Wilkins Company.

[3] Forrester, J., 1969. Urban Dynamics. Waltham, MA, Pegasus Communications.

[4] Repenning, N. P., Goncalves, P., Black, L. J., 2001. Past the Tipping Point: The Persistence of Firefighting in Product Development. Industrial Management Review, 43(4):44–54.

[5] Forrester, J., 1971. Counterintuitive Behavior of Social Systems. Industrial Management Review, 73(2):52–68.

[6] Dangerfield, B., Roberts, C., 1996. An Overview of Strategy and Tactics in System Dynamics Optimization. Journal of the Operational Research Society, 47:405–423.

[7] Powell, M. J. D., 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. Computer Journal, 7(2):155 – 162.

[8] Box, M. J., 1965. A New Method of Constrained Optimization and a Comparison With Other Methods. The Computer Journal, 8(1):42 – 52.

[9] Wächter, A., Biegler, L. T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming, 106:25 – 57.

[10] Drud, A. S., 1985. CONOPT: A GRG Code for large sparse dynamic nonlinear optimization problems. Mathematical Programming, 31(2):153 – 191.

[11] Drud, A. S., 1994. A Large Scale GRG Code. ORSA Journal on Computing, 6(2):207 – 216.

[12] Karush, W., 1939. Minima of Functions of Several Variables with Inequalities as Side Constraints. Master's thesis, Dept. of Mathematics, Univ. of Chicago, Chicago, Illinois.

[13] Kuhn, H., Tucker, W., 1951. Nonlinear Programming. In Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press, 481 – 492.

[14] Forrester, J., 1968. Market Growth as Influenced by Capital Investment. Industrial Management Review, 9(2):83–105.

[15] Forrester, J. W., 1971. World Dynamics. Wright-Allen Press, Boston, MA.

[16] Ozbekhan, H., 1970. The Predicament of Mankind: A Quest for Structured Responses to Growing World-Wide Complexities and Uncertainties. Original Proposal to the Club of Rome.

[17] Forrester, J., 1971. Counterintuitive Behavior of Social Systems. Industrial Management Review, 73(2):52–68.

[18] Richardson, G. P., Pugh, A. L., 1981. Introduction to System Dynamics Modeling with Dynamo. MIT Press, Cambridge, MA, USA.

[19] Lyneis, J. M., Ford, D. N., 2007. System Dynamics Applied to Project Management: A Survey, Assessment, and Directions for Future Research. System Dynamics Review, 23(2-3):157 – 189.

[20] De Boor, C., 1978. A Practical Guide to Splines. Number 27 in Applied Mathematical Sciences. Springer Verlag, New York.

[21] Fügenschuh, A., Vierhaus, I., 2013. A Global Approach to the Optimal Control of System Dynamics Models. Technical Report, ZIB Technical Report ZR-13-28.