Aachen Institute for Advanced Study in Computational Engineering Science

Preprint: AICES-2009-8

28/January/2009

Incremental Single Shooting - A Robust Method for the Estimation of Parameters in Dynamical Systems

C. Michalik, R. Hannemann and W. Marquardt

Financial support from the Deutsche Forschungsgemeinschaft (German Research Association) through grant GSC 111 is gratefully acknowledged.

List of AICES technical reports: http://www.aices.rwth-aachen.de/preprints

Incremental Single Shooting - A Robust Method for the Estimation of Parameters in Dynamical Systems

Claas Michalik, Ralf Hannemann, Wolfgang Marquardt *

AVT - Process Systems Engineering, RWTH Aachen University, Turmstraße 46, D-52064 Aachen, Germany.

Abstract

Estimating the parameters of a dynamical system based on measurement data is an important but complex task in industrial and scientific practice (Schittkowski, 2002). Due to its importance, different approaches to solve this kind of problem have been developed. The most established ones are single shooting (Bard, 1974), multiple shooting (Bock, 1983) and full discretization techniques (Biegler, 1984). Single shooting is the most natural and simple approach to the problem, directly combining numerical integration and optimization techniques. However, for unstable or singular systems the numerical integration of the dynamic model may fail. This problem is especially severe in the framework of parameter estimation, since the dynamic model has to be integrated many times with different parameter estimates. Multiple shooting and full discretization aim at overcoming these deficiencies but suffer from other drawbacks. Therefore single shooting is still widely applied in industry and academia. In this work we present a novel method called Incremental Single Shooting or ISS for short, which aims at overcoming the deficiencies of the classical single shooting approach while keeping its advantages.

Key words: single shooting, parameter estimation, numerical optimization, least-squares, dynamic parameter estimation, multiple shooting

1 Introduction

Estimating parameters in dynamical systems is an important task in industry and academia. Different methods are known to solve this kind of problem, the three most established ones are *single shooting* (Bard, 1974), *multiple shooting* (Bock, 1983) and *full discretization* (Tsang et al., 1975; Biegler, 1984). All these methods possess advantages and suffer from drawbacks. Hence, no single method turned

* Corresponding author. Address: Wolfgang.Marquardt@avt.rwth-aachen.de

out to be the general '*weapon of choice*' in dynamic parameter estimation. Before presenting our novel method (ISS) in detail, we will give a short review on classical parameter estimation techniques and briefly discuss their advantages and disadvantages.

2 Established Methods for the Estimation of Parameters in Dynamic Models

We consider a class of dynamical systems described by means of differential-algebraic equations. Let $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ be the vector of differential variables at time $t \in [t_0, t_f]$, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ be the vector of algebraic variables and $\mathbf{z}(t) = (\mathbf{x}(t)^T, \mathbf{y}(t)^T)^T \in \mathbb{R}^{n_z}$, $n_z = n_x + n_y$, be the vector of differential and algebraic variables. Let $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ be the vector of inputs und $\mathbf{p} \in P$ be a time-invariant parameter vector which has to be estimated. $P \subset \mathbb{R}^{n_p}$ is assumed to be a compact set. We assume that consistent initial conditions $\mathbf{z_0}$ and the inputs $\mathbf{u}(t)$ are known. Then the dynamics of the system are given by

$$F(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) = 0, \tag{1}$$

$$\mathbf{z}(t_0) = \mathbf{z_0}. \tag{2}$$

We assume that the index of the differential-algebraic system is less than or equal to one. Let $t_0 < t_1 < \cdots < t_l = t_f$ represent a grid of points in time. Let $\tilde{\mathbf{z}}_\mathbf{i}$, $i = 1, \ldots, l$, be measurements of the variables at the times t_i, $i = 1, \ldots, l$. For simplicity of presentation we assume that all variables are measured. The measurements are arranged in the column vector

$$\tilde{\mathbf{Z}} := \begin{pmatrix} \tilde{\mathbf{z}}_\mathbf{1} \\ \vdots \\ \tilde{\mathbf{z}}_\mathbf{l} \end{pmatrix} \in \mathbb{R}^{l \cdot n_z}, \quad \text{and} \quad \mathbf{Z}(\mathbf{p}) := \mathbf{Z}(t_1, t_2, \ldots, t_l, \mathbf{z_0}, \mathbf{u}; \mathbf{p}) := \begin{pmatrix} \mathbf{z}(t_1) \\ \vdots \\ \mathbf{z}(t_l) \end{pmatrix} \in \mathbb{R}^{l \cdot n_z}, \tag{3}$$

the vector of the corresponding model predicitions, computed by the solution of the initial value problem (1),(2).

Let $\mathbf{V_M} \in \mathbb{R}^{(l \cdot n_z) \times (l \cdot n_z)}$ be the covariance matrix of the measurements. We consider a weighted least-squares parameter estimation problem of the form

$$\min_{\mathbf{p} \in P} \quad (\tilde{\mathbf{Z}} - \mathbf{Z}(\mathbf{p}))^T \, \mathbf{V_M^{-1}} \, (\tilde{\mathbf{Z}} - \mathbf{Z}(\mathbf{p})). \tag{4}$$

2.1 Single Shooting

Single shooting also known as initial value approach has been introduced for optimal control of ODEs by Kraft (1985) and of DAEs by Cuthrell and Biegler (1987). Schittkowski (2002) employs the single shooting approach for the estimation of parameters in dynamical systems. In single shooting the dynamical system is solved by a numerical integrator. $\mathbf{Z}(\mathbf{p})$ in (4) is directly computed by numerically solving an initial value problem and the vector $\mathbf{p} \in P$ is the only degree of freedom for the nonlinear optimizer. The advantage of single shooting is that standard DAE solvers with sensitivity analysis capabilities and standard NLP solvers can be applied. Due to the use of standard DAE solvers the grid of the state discretization can be adapted automatically such that the error of the states is below a prescribed error tolerance. The disadvantage is that unstable systems are difficult or even impossible to converge even if a good initial guess for the optimization variables is available. Also, single shooting can converge to several local minima due to the usually high nonlinearity of the resulting NLP. For the treatment of unstable systems, multiple shooting or full discretization seem to be more favorable (Peifer and Timmer, 2007).

2.2 Direct Multiple Shooting

Multiple shooting for the direct optimization of optimal control problems [1] has been introduced by Bock and Plitt (1984). Roughly spoken, multiple shooting for direct optimization is an adaption of the multiple shooting method for the solution of multipoint boundary value problems to optimization. Multiple shooting for the solution of boundary value problems has firstly been investigated by Keller (1968), Osborne (1969) and Bulirsch (1971). A state-of-the-art direct multiple shooting implementation is MUSCOD of Leineweber et al. (2003a,b). The basic idea of multiple shooting is to divide the time horizon into a number of intervals. For simplicity we assume that the edges of the intervals coincide with the grid $t_0 < t_1 < \cdots < t_l$. If the dynamical system is described by a set of ordinary differential equations just the initial values of the state variables $\hat{\mathbf{x}}_\mathbf{i}$, $i = 1 \ldots, l$, are adjoint as additional degrees of freedom to the optimizer. To ensure the continuity of the trajectories, junction conditions are added as equality constraints to the overall nonlinear pro-

[1] Since this method is used for the direct optimization of optimal control problems it is sometimes referred as *direct multiple shooting* to distinguish it from the multiple shooting method for the solution of multipoint boundary value problems.

gram. We use the denotation

$$\hat{\mathbf{Z}} := \begin{pmatrix} \hat{\mathbf{x}}_\mathbf{1} \\ \vdots \\ \hat{\mathbf{x}}_\mathbf{l} \end{pmatrix} \in \mathbb{R}^{l \cdot n_z},$$

and let $\mathbf{x_i}(t; p)$, $i = 1, \ldots, l$, be the solution of the initial value problem

$$\dot{\mathbf{x}}_\mathbf{i}(t) = f(\mathbf{x_i}(t), \mathbf{u}(t), \mathbf{p}), \; t \in [t_{i-1}, t_i], \tag{5}$$

$$\mathbf{x_i}(t_{i-1}) = \hat{\mathbf{x}}_\mathbf{i-1}, \tag{6}$$

where for notational convenience we set $\hat{\mathbf{x}}_\mathbf{0} := \mathbf{x_0}$. Then, the nonlinear program in multiple shooting is given by

$$\min_{\mathbf{p} \in P, \hat{\mathbf{x}}_\mathbf{i}, i=1,\ldots,l} \quad (\tilde{\mathbf{Z}} - \hat{\mathbf{Z}})^T \, \mathbf{V_M^{-1}} \, (\tilde{\mathbf{Z}} - \hat{\mathbf{Z}}) \tag{7}$$

$$\text{s.t.} \quad \mathbf{x_i}(t_i; p) - \hat{\mathbf{x}}_\mathbf{i} = 0. \tag{8}$$

In the beginning of the optimization, the junction conditions (8) do not have to be satisfied, allowing for discontinuous trajectories to avoid instabilities. At the optimal solution, the junction conditions are satisfied yielding a continuous trajectory (see also Figure 1).

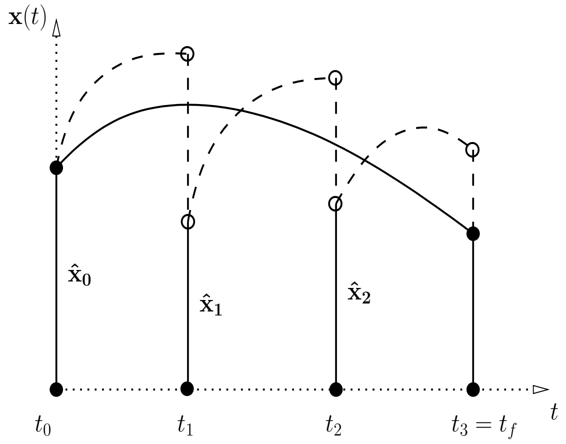

Fig. 1. Illustration of multiple shooting, the dashed lines show the initial trajectory, the solid line shows the trajectory if the junction conditions (8) are satisfied.

If DAEs instead of ODEs describe the dynamical system, several strategies can be applied. For example, the initial values of the algebraic variables at each shooting interval can be computed by a consistent initialization. Or, the initial values of the DAEs can be added as additional degrees of freedom if for each free initial value the algebraic equations are adjoint as equality constraints to the nonlinear program. Then, relaxation techniques to conserve consistency of the algebraic equations have to be applied (Leineweber et al., 2003a). The advantage of the multiple shooting method is that a standard DAE solver for stiff systems with stepsize control can be employed and the computer code can be parallelized in a natural way. As in single shooting the use of stepsize control guarantees that the error of the state variables is less than a prescribed error tolerance. On the other hand, due to the introduction of the additional degrees of freedom the size of the NLP is enlarged, especially if the dimension of the parameter vector is rather small compared to the dimension of the differential states.

2.3 Full Discretization

The full discretization method or simultaneous approach was developed by Tsang et al. (1975). In the simultaneous approach the numerical integration scheme for the differential-algebraic equations is added as equality constraints to the nonlinear program. Orthogonal collocation is a popular choice for the discretization of the differential-algebraic equations (Biegler, 1984). The choice of the discretization scheme is crucial for the success of the method. If the ”wrong” discretization scheme is used, the nonlinear program may diverge if the discretization is refined (Kameswaran and Biegler, 2006). Usually, the full discretization leads to large-scale nonlinear programs with a sparse constraint Jacobian and a sparse Hessian of the Lagrangian. Therefore, special optimization strategies have to be applied (Cervantes et al., 2000). On the other hand, it is well known that the simultaneous approach can successfully deal with instabilities of the dynamical model (Kameswaran and Biegler, 2006).

3 Incremental Single Shooting (ISS)

As discussed above, all methods presented here have some advantages and some disadvantages. The single shooting method is widely applied since it is comparably simple to implement. In addition, the method is available in many modeling environments and parameter estimation tools, as for instance gPROMS (ProcessSystemsEnterprise, 2006), Aspen Custom Modeller (Aspentech, 2007), JACOBIAN (Numerica Technology LLC, 2008) and Easy-Fit (Schittkowski, 2002). On the other hand, the method lacks numerical robustness (Schittkowski, 2002). The method presented here (ISS) is based on the single shooting approach and aims at overcoming its deficiencies while keeping its advantages. As discussed above, the main drawback of the single shooting method is the potential infeasibility of the numerical integration, which might break down in the course of the integration. If the model structure under investigation is correct, this can happen due to a very stiff set of differential equations or – even more severe – in case of a set of differential equations that is unstable or has singularities for certain parameter values.

Often, these problems are also caused by wrong estimates of the model parameters or initial values. However, the numerical integration typically breaks down in the course of the integration and not immediately at its beginning. Hence, the basic idea of ISS is to avoid the numerical integration of the whole period of time, as long as the parameter values are uncertain, by solving the parameter estimation on a reduced and successively enlarged time and data horizon. Accordingly, the period of integration is successively increased along with the accuracy of the estimated parameters until the integration is performed over the whole period of time, which is defined by the latest available measurement information.

3.1 Algorithm of the ISS Method

The suggested algorithm is depicted in Figure 2. The first step is mainly identical to any other parameter estimation method. The user has to provide initial guesses for the parameters to be estimated (`P_Start`) along with lower (`LB`) and upper (`UB`) bounds for the parameter values[2]. In addition ISS allows the user to provide a vector of times $(t_1, t_2, ..., t_{ISS\_End})$ entitled `ISS_Intervals` corresponding to the integration intervals solved successively. If no `ISS_Intervals` are provided, the intervals are determined automatically according to a heuristic described later. Hence, during the $i'th$ ISS step, a time horizon from 0 to the $i'th$ element in `ISS_Intervals` is used for the parameter estimation. It is obvious that `ISS_Intervals` should contain increasing values ending with the time of the latest available measurement information.

Having all these user provided information at hand, the next step of the algorithm is to perform some initializations. First, the counter variable (`ISS_No`) for the ISS step is assigned to 1 and a variable indicating the number of ISS steps (`ISS_End`) that have to be performed is set to the number of `ISS_Intervals`. The vector of parameter estimates (`P_Est`) is initialized to the user provided initial guess. The lower bounds (`ISS_LB`) and upper bounds (`ISS_UB`) used during the ISS steps are initialized to the user provided values for the parameter bounds. In addition, the standard deviations of the estimated parameters (`P_STD`) are initialized to infinity.

After the initialization, the first ISS problem can be solved. Here, any available single shooting algorithm can be used to solve the problem with the given initial parameter guesses and parameter bounds. However, not all available measurement information are used. For the $i'th$ ISS interval the set of measurement data is reduced to those stemming from measurements taken in between $t = 0$ and $t = t_i$. Hence, the numerical integration, which is at the core of any single shooting algorithm, also only has to be performed from 0 to t_i.

For ISS interval i the algorithm thus starts with solving the parameter estimation

[2] If no reasonable bounds are known, bounds of $\pm\infty$ can be provided.

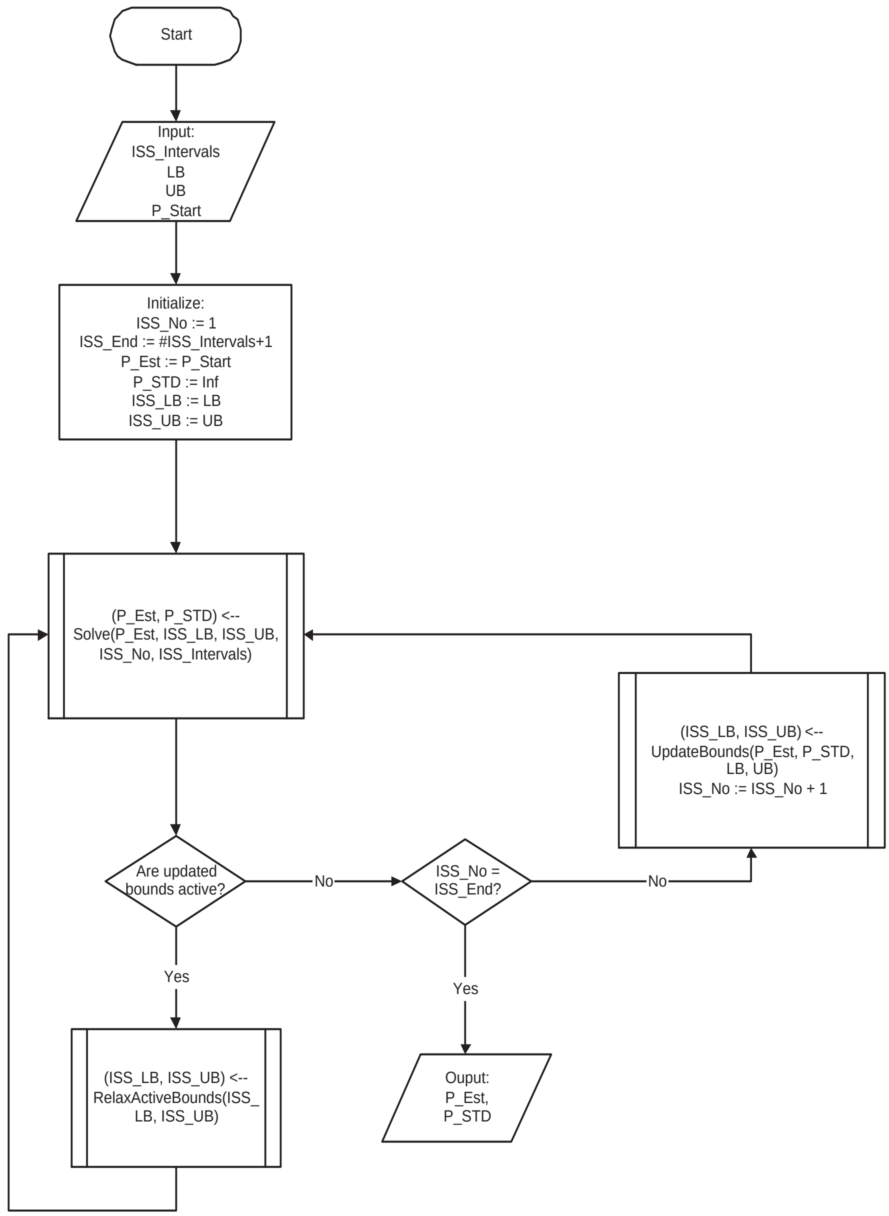

Fig. 2. Flowchart for ISS

problem on the time horizon $0 < t < t_i$ based on the available parameter estimates and bounds. The result is an updated vector `P_Est` of estimated parameter values. In addition, the vector `P_STD` needs to be updated for the calculation of new parameter bounds as described below. In our implementation we estimate `P_STD` based on the covariance matrix of the parameters (V_P) using

$$V_P = (S^T \cdot V_M^{-1} \cdot S)^{-1} \quad (9)$$

where S is the sensitivity matrix of the model predictions with respect to the model parameters and V_M is the covariance matrix of the measurement data.

The next step in the algorithm is to check whether or not so-called *updated bounds* are active. Any bound which differs from the original value provided by the user (stored in `LB` and `UB`) is an updated bound. If any updated bounds are active, these bounds are relaxed according to

$$ISS\_UB_{new} = min\Big(ISS\_UB_{old} + BRF \cdot (ISS\_UB_{old} - ISS\_LB_{old}), UB\Big) \tag{10}$$

$$ISS\_LB_{new} = max\Big(ISS\_LB_{old} - BRF \cdot (ISS\_UB_{old} - ISS\_LB_{old}), LB\Big) \tag{11}$$

and the estimation problem is solved again on the same time horizon. Here BRF is the *bound relaxation factor* which can be assigned any positive value. In our implementation we use $BRF = 2$. A higher value can reduce the necessary number of bound relaxations and thus the computational cost but at the same time decreases the robustness of the algorithm.

If `ISS_No` < `ISS_End` the algorithm continues with the regular update of the bounds. Here `ISS_LB` and `ISS_UB` are set to the 99% confidence values of the estimated parameters[3] if these values do not violate the corresponding original lower (`LB`) or upper bounds (`UB`), respectively. If the new bounds violate any original bound, the value of the original bound is used instead. In addition `ISS_No` is increased by one and the next ISS problem is solved. The algorithm terminates if the last ISS interval has been solved. In this case the parameter estimates and their standard deviations are returned. Please note that the number of degrees of freedom for the NLP arising from each ISS sub-problem is identical to the single shooting case. Hence, no additional degrees of freedom (in opposite to multiple shooting and full discretization) are added to the NLP.

3.2 Determining the ISS intervals

The choice of ISS intervals has a high impact on the robustness and performance of the algorithm. Various options are possible, as for instance to determine the ISS intervals based on an analysis of the system. If the numerical integration proceeds to the final time with the initially provided parameter estimates, classical SS can be applied. If the integration breaks down at t_{break}, the first ISS interval can be chosen to be $t_1 = \alpha \cdot t_{break}$, where α is some value between 0 and 1 used to tune the robustness of the method. Our experience is that such methods do not work well, since the instabilities typically occur in the course of the estimation procedure. Frequently it is no problem to integrate the system with the initially provided parameter estimates, but the integration breaks down during the estimation procedure after some iterations. This is, for example, true for the second exemplary system considered (see Section 5). Here the integration is possible for all provided initial parameter guesses, but single shooting fails in 84% of the test cases during the estimation procedure due to integration errors. Therefore, we use another strategy as default

[3] Any other confidence level, as for instance the 95% confidence level can be used as well.

option and for the test cases described below. We choose the ISS intervals such that the i'th interval contains 2^{i-1} data points for each measured quantity. For equally spaced measurement data this means that the first ISS interval goes from 0 to the time of the first measurement and that subsequent ISS intervals are always twice as long as the previous one.

4 Illustrative Parameter Estimation Example

To illustrate the advantages of our method as compared to classical single shooting, we apply it to two exemplary parameter estimation problems. It is important to mention that – for the sake of simplicity – two ordinary differential equation (ODE) examples are considered. However, the algorithm can, without any changes, be applied to differential-algebraic systems (DAEs) as well.

4.1 Exemplary Systems I

The first exemplary system is a well known ill-posed parameter estimation example, originally presented by Bulirsch (1971). It is given by the second order differential equation

$$\ddot{y} = \mu^2 \cdot y - (\mu^2 + p^2) \cdot sin(p \cdot t) \tag{12}$$

with initial conditions $\dot{y}(0) = 0$ and $y(0) = \pi$, and can be represented by the equivalent first order system

$$\dot{y}_1 = y_2, \quad y_1(0) = \pi \tag{13}$$

$$\dot{y}_2 = \mu^2 \cdot y_1 - (\mu^2 + p^2) \cdot sin(p \cdot t), \quad y_2(0) = 0. \tag{14}$$

Here $\dot{\bullet}$ represents the first and $\ddot{\bullet}$ the second derivative w.r.t. time. For $p = \pi$ the exact solution of Eq. (12) is

$$y(t) = y_1(t) = sin(\pi \cdot t) \tag{15}$$

$$y_2(t) = \pi \cdot cos(\pi \cdot t). \tag{16}$$

The numerical integration of Eqs. (13)-(14) from 0 to 1 fails for values of μ above approximately 50, depending on the applied integration tolerances and method, even if p approximates π to machine precision. Eqs. (13)-(14) therefore have been used in different publications (see for instance Bock (1983) and Schittkowski (2002)) to test parameter estimation and integration methods. Schittkowski (2002) explains this instability considering the general solution

$$y_1(t) = sin(p \cdot t) + \epsilon \cdot sinh(\mu \cdot t) \tag{17}$$

$$y_2(t) = p \cdot cos(p \cdot t) + \epsilon \cdot \mu \cdot cosh(\mu \cdot t), \tag{18}$$

with $\epsilon = \frac{(\pi - p)}{\mu}$. In Eqs. (17)-(18) it is obvious that even slight numerical deviations of p from π lead to an exponential increase of the computed solution. Thus, if p is the parameter to be estimated based on measurement information, this example is perfectly suited to test the robustness of a parameter estimation method.

4.2 Exemplary Systems II

The second example considered is a classical Lotka-Volterra ODE system given by

$$\dot{y}_1 = -k_1 \cdot y_1 + k_2 \cdot y_1 \cdot y_2 \tag{19}$$

$$\dot{y}_2 = k_3 \cdot y_2 - k_4 \cdot y_1 \cdot y_2 \tag{20}$$

with initial conditions $y_1(0) = 0.4$, $y_2(0) = 1$ and parameter values $k_1 = k_2 = k_3 = 1$, $k_4 = 0.1$. This system has been used by different authors as a test example for parameter estimation algorithms (see for instance: (Clark, 1976; Edsberg and Wedin, 1995; Schittkowski, 2002; Schwatz and Bremermann, 1975; Varah, 1982)). The system is singular for various combinations of parameter values as for instance $k_1 = k_2 = k_3 = 0.5, k_4 = -0.2$, which results in a pole near t = 3.3, where any numerical integration algorithm will break down (Schittkowski, 2002).

4.3 Estimation Tasks

To compare the novel method ISS with classical single shooting, we solve different estimation tasks based on the two exemplary systems given above.

4.3.1 Exemplary System I

For system I, we assume that the initial conditions are known and that 10 measurement data for y_1 and y_2, equally distributed between t = 0.1 and t = 1 are available. The measurements are based on the exact analytical solution of Eq. (12) and disturbed by uncorrelated Gaussian noise with a standard deviation of 0.05. The estimation task is to obtain a value of p, with the correct solution $p = \pi$. The value of μ is varied from 5 to 40 to adjust the ill-posedness of the problem. To allow for a sound comparison of both methods we perform 100 parameter estimations using different, randomly chosen initial guesses for p, stemming from a uniform distribution between -500 and 500. We repeat this procedure for $\mu = 5, \mu = 10, \mu = 20$ and $\mu = 40$ and solve each estimation task using the novel ISS and the classical single shooting approach. For the integration of the ODE-system along with its sensitivity equations we use the `MATLAB` DAE-integrator `ODE15s` and for the optimization we use `SNOPT`. We also apply a reformulation of the least-squares problem proposed by Schittkowski (1988) to enhance the robustness and efficiency

Table 1
Results for single shooting (SS) and ISS for exemplary system I and different values of μ

μ	Method	Correct Solution	Wrong Solution	Int. Error	Opt. Error
5	ISS	18	82	0	0
5	SS	16	84	0	0
10	ISS	100	0	0	0
10	SS	75	0	0	25
20	ISS	99	0	0	1
20	SS	20	0	0	80
40	ISS	100	0	0	0
40	SS	0	0	0	100

of the solution of least-squares problems. Therefore, the original least-squares type problem of the form

$$min \quad \sum_{i}^{l} f_i(x)^2, \quad x \in \mathbb{R}^n \tag{21}$$

with $f_i(x)$ being the (potentially weighted) residual for the i'th measurement value is transformed to the equivalent problem

$$min \quad z^T z \tag{22}$$
$$s.t. \quad F(x) - z = 0, \quad x, z \in \mathbb{R}^{n+l}, \tag{23}$$

where n is the dimension of the parameters to be estimated, l is the number of measurement data, $F(x) = (f_1(x), ..., f_l(x))^T$ and $z = (z_1, ..., z_l)^T$. Schittkowski (1988) shows that this reformulation of the least-squares optimization problem leads to an increased robustness and efficiency, if a sequential quadratic programming type of optimization algorithms is applied. For the solution of the exemplary systems we apply the same solvers, settings and initial values for single shooting and ISS and categorize the results as follows:

- correct solution: The algorithm converges and the maximum difference between the globally optimal parameter value and the estimated parameter value is less than 5% for all parameters.
- wrong solution: The algorithm converges and the difference between the globally optimal parameter value and the estimated parameter value is greater or equal to 5% for at least one parameter.
- integration error: The algorithm does not converge due to an error during the numerical integration.
- optimization error: The algorithm does not converge due to an optimization error.

The results can be found in Table 1 and are discussed in more detail in Section 5.

Table 2
Results for single shooting (SS) and ISS for exemplary system II

	Correct Solution	Wrong Solution	Integration Error	Optimization Error
ISS	69	2	29	0
SS	1	15	84	0

4.3.2 Exemplary System II

For the second exemplary system, we assume that the initial conditions are known and that 20 measurement data for y_1 and y_2, equally distributed from t = 0.5 to t = 10 are available. The data are based on the numerical integration of Eqs. (19)-(20) and disturbed by uncorrelated Gaussian noise with a relative standard deviation of 5% of the simulation value. The task is to obtain estimates for the 4 parameters $k_1 - k_4$. Again, we solve the estimation task 100 times using randomly chosen initial guesses (equally distributed between 0 and 10) for the parameter values. We solve each of these 100 estimation tasks using the novel ISS and the classical single shooting approach using the same settings and algorithms as described above. Table 2 allows to compare the robustness of single shooting and ISS and is discussed in more detail in the following section.

5 Results and Discussion

5.1 Exemplary System I

The results presented above show that ISS is more robust than classical single shooting. For system I and $\mu = 5$, both methods ISS and single shooting converge to a solution in 100% of the considered examples. This is not surprising, since the problem is well-posed and the necessary numerical integration of the system is unproblematic for such a low value of μ. However, the test problem seems to possess many local optima, since both methods, single shooting and ISS only converge to the correct solution in 16% and 18% of the considered test cases, respectively. In the remaining cases the algorithms converge to a local optimum. Hence, for the well-posed example with $\mu = 5$ both methods, ISS and single shooting perform almost equally well, with minor advantages for ISS.

Totally different results are obtained for $\mu = 10$. Here ISS clearly outperforms the classical single shooting approach. It converges to the global optimum in 100% of the test cases compared to 75% for single shooting, which terminates with the optimization error ’`no minimizer, the current point cannot be improved`’ in the remaining 25% of test cases.

Similar but even more extreme results are obtained for $\mu = 20$. Here ISS converges to the global optimum in 99% and fails due to an optimization error in 1% of the test cases. Single shooting on the other hand converges to the global optimum only in 20% of the test cases and fails due to an optimization error in the remaining 80% of the test cases. However, the optimization error differs from the on before. Here the optimization algorithm terminates with the error: '`General constraints cannot be satisfied accurately`'. The error message is surprising at the first glance, since the problem considered is an unconstrained least-squares problem. The constraints, however, appear due to the reformulation of the problem described above and obviously `SNOPT` is not able to find a feasible point if single shooting is used.

For $\mu = 40$ the trend seen thus far holds and ISS converges to the correct solution while single shooting fails to find the global optimum anytime. Here, single shooting fails due to the same optimization error as before, namely: '`General constraints cannot be satisfied accurately`'.

These results clearly indicate that ISS is more robust than single shooting, which seems reasonable. However, the results are surprising, since ISS converges to the global optimum in 100% of the considered test cases for $\mu = 40$ compared to only 16% for $\mu = 5$. Hence, the ill-posedness seems to be advantageous instead of problematic in this framework. This behavior can easily be explained based on an analysis of the objective function. Using Eqs. (17) and (18) the objective function can be calculated as a function of p and μ. Figure 3 shows the results for $-500 \leq p \leq 500$ and $\mu = 5$ and $\mu = 10$, respectively. It can be seen that the objective function has many local minima for $\mu = 5$. With increasing values for μ the ill-posedness is increasing, however, the local minima are more and more suppressed. Hence, the optimization task is – in a way – easier to solve for high values of μ except for the numerical difficulties, which are efficiently suppressed by ISS.

For both methods the quality of the final estimate is increasing with increasing values of μ although the same, noise corrupted measurements are used for all estimation tasks. This is due to the sensitivity of the model prediction, which is low for small values of μ. For a low sensitivity the measurement errors lead to comparably large errors in the estimated parameters, whereas, for a high sensitivity they lead to comparably low errors in the estimated parameters. Therefore, optimal experimental design strategies tend to design experiments where the model prediction is most sensitive to the parameter values to allow an exact estimation of the parameters (Espie and Macchietto, 1989). However, as seen above, a high sensitivity can cause numerical problems if classical single shooting is applied. This emphasizes the importance of the robustness for a parameter estimation method.

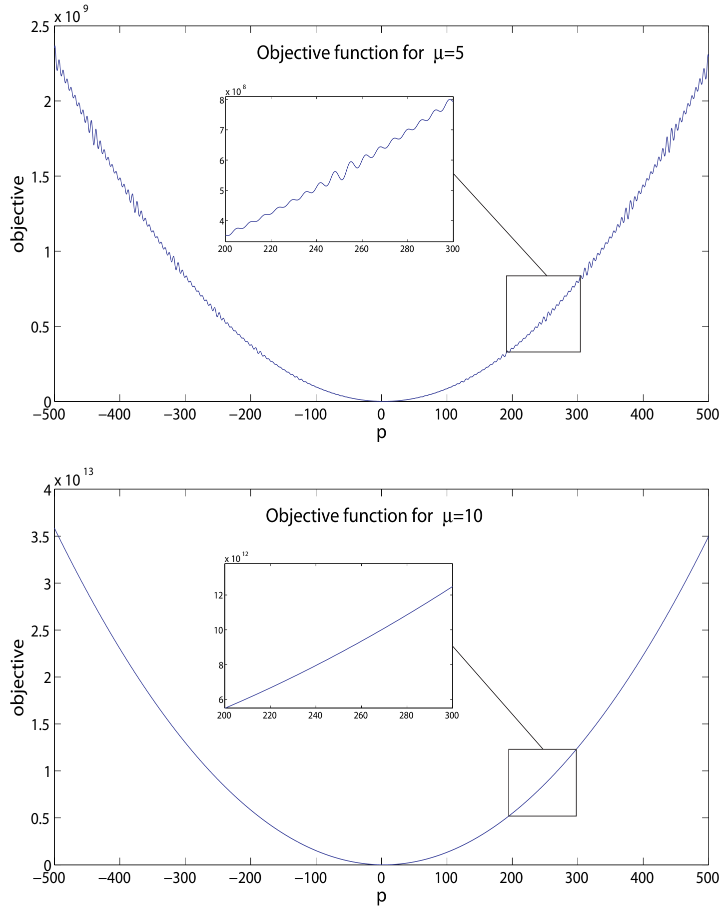

Fig. 3. Objective function for $\mu = 5$ and $\mu = 10$ and $-500 \leq p \leq 500$

5.2 Exemplary System II

The second example is a classical Lotka-Volterra system. It is not surprising that the single shooting approach does not perform well on this example in terms of robustness, due to the potential singularity of the system [4]. However, it is surprising that single shooting only converges to the correct solution in a single of the 100 test cases, whereas ISS converges to the correct solution in 69% of the examples considered. Table 2 also shows that for this example ISS reduces the risk of getting stuck in a local minimum. ISS converges to a local minimum only for 2 test cases compared to 15 for single shooting. This indicates that ISS not only offers advantages in terms of numerical robustness, but also tends to find the global optimum more reliably. Local optima are a well known problem for parameter estimation problems and might lead to completely wrong conclusions. Singer et al. (2006) present some examples from the literature, where local optima lead to wrong conclusions. Therefore they determine the global optimum using global parameter estimation techniques and compare the results to those reported in the literature obtained using local parameter estimation methods. However, such techniques are computationally very expensive and can only be applied to examples of low complexity comprising

[4] The system is not singular at the provided initial parameter guesses.

only few parameters (Papamichail and Adjiman, 2004). Therefore local methods are applied almost exclusively in practice and any local method that increases the chances of finding the global optimum is advantageous. Bock (1983) reports that multiple shooting tends to find the global optimum more reliably than single shooting. ISS is similar to multiple shooting in the sense that it does not use the whole set of available data immediately but solves multiple problems on reduced data sets. Hence, it seems reasonable that ISS can enhance the probability of finding the global optimum, as well. However, further investigations with different test cases are necessary to ascertain this characteristic of ISS.

The results discussed so far show that ISS is more robust than single shooting and can enhance the probability of finding the global optimum. However, no information on the computational efficiency is given thus far. Let us therefore consider the example as described in Section 4.3 for the exemplary system II and the initial parameter guess $k_1 = 4.2176, k_2 = 9.1574, k_3 = 7.9221, k_4 = 9.5949$. Using the classical single shooting approach, the system becomes singular during the second major iteration of `SNOPT` and the calculation is aborted. Using ISS with the same initial guess the correct solution is obtained after 55 iterations. The iterations along with the optimal estimates after each ISS interval are given in Table 3. It is important to mention that the number of iterations is relatively high in the first two ISS intervals. This is where 70% of all iterations are performed. However, the estimates for 3 of the 4 parameters are almost optimal after the second ISS interval. Consequently, only a small number of iterations is necessary in the remaining ISS intervals on longer time horizons. Within a total of 55 iterations the algorithm reaches the optimal estimates [5] for all parameters and terminates.

The number of iterations seems to be rather high; however, considering that most of the iterations are performed on a drastically reduced time horizon, the overall computational cost are quite low. Thus, in order to get an idea of the actual computational cost we calculate the *equivalent iteration number (I_{equiv})*. This value represents the equivalent number of iterations (∼integrations) on the full time horizon as it is carried out in single shooting iterations. For dynamic parameter estimation problems the computational cost are usually dominated by the numerical integration of the dynamic model. Therefore, the *equivalent iteration number (I_{equiv})* defined as

$$I_{equiv} = \frac{1}{t_{ISS\_End}} \sum_{i=1}^{ISS\_End} n_{Iterations,i} \cdot t_i$$

can be used as a measure of the computational cost. Here $n_{Iterations,i}$ is the number of iteration performed in the $i'th$ ISS step. For this example I_{equiv} calculates to $(11 \cdot 0.5 + 28 \cdot 1 + 5 \cdot 2 + 4 \cdot 4 + 3 \cdot 8 + 4 \cdot 10) \div 10 = 12.35$, which is comparably low. Comparing the equivalent iteration number using ISS and the actual number of

[5] Due to the noisy measurements some of the optimal parameter values differ from the true parameter values.

Table 3
Iterations and parameter estimates for the 6 ISS intervals of exemplary system II (optimal solution: $k_1 = 0.9824$, $k_2 = 0.9312$, $k_3 = 1.0521$, $k_4 = 0.1046$)

ISS Interval	$t \leq 0.5$	$t \leq 1$	$t \leq 2$	$t \leq 4$	$t \leq 8$	$t \leq 10$
Iterations	11	28	5	4	3	4
k_1	4.4363	1.0868	0.9172	1.0089	0.9933	0.9824
k_2	3.6457	1.0676	0.9362	0.9371	0.9928	0.9312
k_3	3.2760	0.9482	1.0187	1.0496	1.0588	1.0521
k_4	6.2779	0.0523	0.0977	0.1039	0.1050	0.1046

iterations using single shooting for various problems leads to very different results sometimes favoring single shooting, sometimes ISS. On average the computational cost seem to be slightly higher for ISS.

Different changes to enhance the performance are possible. Coarse optimizer and integrator tolerances could, for instance, be applied in the first ISS intervals. Also, the choice of ISS intervals is based on a heuristic favoring robustness over speed, here. Various strategies to determine the ISS intervals are possible and might help reducing the computational load of the method. However, we did not implement any of these ideas, since one of the major advantages of the method is it's structural simplicity, which allows an easy and robust implementation. In the above described form the implementation effort is almost negligible if a classical single shooting algorithms is available however the robustness is increased significantly.

6 Comparison of ISS with Classical Approaches

In the previous sections we extensively compare ISS with classical single shooting. However, we do not compare it to multiple shooting and full discretization, although these two methods are known to be robust alternatives to single shooting. However, at the current point in time no sound comparison of ISS with these methods is possible for various reasons. First of all, we do not have access to any full discretization or multiple shooting code. In fact, neither open source nor commercially available multiple shooting or full discretization codes are known to us. Hence, for comparison purposes we would have to implement these methods on our own. Both methods are not straight forward to implement and hence a comparison of ISS with our own (technically not mature) multiple shooting and full discretization code seems senseless. Alternatively, we could collaborate with other groups and compare ISS with sophisticated packages as MUSCOD of Leineweber et al. (2003a,b). However, such a comparison does not make sense at this point in time, since our implementation of ISS is a MATLAB based prototype that is not optimized for efficiency. In addition, the sophisticated packages usually come with their own, tailored numerical integrators and NLP codes. Hence the comparison would primarily test the underlying numerical integrators and NLP algorithms instead of the parameter estimation methods. However, we want to stress that the main goal of ISS is to offer a robust alternative for single shooting that can easily and quickly be implemented. Such a method is desirable since almost all available software packages for the parameter estimation in dynamic systems rely on single shooting, besides the drawbacks mentioned.

7 Outlook

Some ideas to increase the computational performance of the algorithm have already been presented in Section 5. Besides further improving the algorithm, future work will focus on testing and evaluating the algorithm. Furthermore we plan to incorporate the method into the efficient large scale dynamic optimization software package DyOS (Schlegel et al., 2005), such that a sound comparison of ISS with the major parameter estimation techniques for dynamical systems (single shooting, multiple shooting and full discretization) becomes possible. In addition, different strategies to determine the ISS intervals will be implemented and compared in terms of numerical robustness and computational efficiency.

8 Acknowledgements

The financial support by the the *Deutsche Forschungsgemeinschaft* (DFG) via *collaborative research center 540* (SFB 540) and the *Fonds der Chemischen Industrie* is greatly appreciated.

References

Aspentech, 2007. Process and equipment model development and simulation environment, http://www.aspentech.com/products/aspen-custom-modeler.cfm.

Bard, Y., 1974. Nonlinear Parameter Estimation. Academic Press, New York.

Biegler, L. T., 1984. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. Comput. Chem. Eng. 8, 243–248.

Bock, H. G., 1983. Recent advances in parameter identification techniques for ODE. In: Deuflhard, P., Hairer, E. (Eds.), Numerical Treatment of Inverse Problems in Differential and Integral Equations. Birkhäuser, Boston, pp. 95–121.

Bock, H. G., Plitt, K. J., Juli 1984. A multiple shooting algorithm for direct solution of optimal control problems. In: 9th IFAC World Congress Budapest. Pergamon Press, pp. 242–247.

Bulirsch, R., 1971. Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung. Tech. rep., Carl-Cranz-Gesellschaft, Oberpfaffenhofen.

Cervantes, A. M., Wächter, A., Tütüncü, R. H., Biegler, L. T., 2000. A reduced space interior point strategy for optimization of differential algebraic systems. Computers Chem. Eng. 24 (1), 39–51.

Clark, C., 1976. Mathematical Bioeconomics. Wiler-Interscience, New York.

Cuthrell, J. E., Biegler, L. T., 1987. On the optimization of differential-algebraic process systems. AIChE Journal 33 (8), 1257–1270.

Edsberg, L., Wedin, P. A., 1995. Numerical tools for parameter estimation in ODE-systems. Optimization Methods and Software 6, 193–218.

Espie, D., Macchietto, S., 1989. The optimal design of dynamic experiments. AIChE Journal 35 (2).

Kameswaran, S., Biegler, L., 2006. Simultaneous dynamic optimization strategies: Recent advances and challenges. Computers Chem. Eng. 30 (10-12), 1560–1575.

Keller, H. B., 1968. Numerical Methods for Two-Point Boundary-Value Problems. Blaisdell Waltham, Mass.

Kraft, D., 1985. On converting optimal control problems into nonlinear programming problems. Computational Mathematical Programming, 261–280.

Leineweber, D. B., Bauer, I., Bock, H. G., Schlöder, J. P., 2003a. An efficient multiple shooting based reduced sqp strategy for large-scale dynamic process optimization – part I: Theoretical aspects. Computers Chem. Eng. 27 (2), 157–166.

Leineweber, D. B., Bauer, I., Bock, H. G., Schlöder, J. P., 2003b. An efficient multiple shooting based reduced sqp strategy for large-scale dynamic process optimization – part II: Software aspects and applications. Computers Chem. Eng. 27 (2), 167–174.

Numerica Technology LLC, 2008. JACOBIAN dynamic modeling and optimization software, `http://www.numericatech.com/`.

Osborne, M. R., 1969. On shooting methods for boundary value problems. J. Math. Anal. Appl 27, 417–433.

Papamichail, I., Adjiman, C. S., 2004. Global optimization of dynamic systems. Computers Chem. Eng. 28, 403–415.

Peifer, M., Timmer, J., 2007. Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting. IET Syst Biol 1, 78–88.

ProcessSystemsEnterprise, 2006. gproms, modelling software, http://www.psenterprise.com/gproms/index.html.

Schittkowski, K., 1988. Solving nonlinear least squares problems by a general purpose sqp-method. In: K.-H. Hoffmann, J.-B. Hiriart-Urruty, C. L. J. Z. (Ed.), Trends in Mathematical Optimization. pp. 295–309.

Schittkowski, K., 2002. Numerical Data Fitting in Dynamical Systems: A Practical Introduction With Applications and Software. Kluwer Academic Publishers.

Schlegel, M., Stockmann, K., Binder, T., Marquardt, W., 2005. Dynamic optimization using adaptive control vector parameterization. Comput. Chem. Eng. 29 (8), 1731–1751.

Schwatz, J., Bremermann, H., 1975. Discussion of parameter estimation in biological modelling: Algorithms for estimation and evaluation of the estimates. Journal of Mathematical Biology 1 (3), 241–257.

Singer, A. B., Taylor, J. W., Barton, P. I., Green, W. H., 2006. Global dynamic optimization for parameter estimation in chemical kinetics. Journal of Physical Chemistry A 110 (3), 971–976.

Tsang, T. H., Himmelblau, D. M., Edgar, T. F., 1975. Optimal control via collocation and nonlinear programming. Int. J. Control 21, 763–768.

Varah, J. M., 1982. A spline least squares method for numerical method for numerical parameter estimation in differential equations. SIAM Journal on Scientific Statistical Computing 3, 28–46.