Design Sensitivity Analysis of Kinematically Driven Multibody Systems

ing. R.H.V. Kolkhuis Tanke

WFW-report 96.076
Design Sensitivity Analysis of Kinematically Driven Multibody Systems

ing. R.H.V. Kolkhuis Tanke

Eindhoven University of Technology
Faculty of Mechanical Engineering
Computational and Experimental Mechanics Group

WFW-report : 96-076
Date : June 1, 1996
Student : ing. R.H.V. Kolkhuis Tanke
Coach : ir. L.F.P. Etman
Supervisors :
  dr. ir. A.J.G. Schoofs
  prof. dr. ir. D.H. van Campen
Summary

The optimization of multibody systems using computer oriented techniques is characterized by two specific problems. Firstly, a direct link between standard optimization code and general-purpose multibody analysis code is generally difficult to implement and can be costly in terms of computational efforts. Secondly, standard optimization algorithms can not handle time-dependent problems, which often occur for multibody systems.

A solution to the first problem is the positioning of a local approximation model in between optimization and multibody analysis code. Basically, the optimization problem is replaced by an approximate problem that is explicitly known. This results in an easier implementation of the link between the two codes. Furthermore, the user has more control of the optimization process, and a reduction of the computational efforts can be obtained. A local approximation is based upon function values and derivatives with respect to the design variables in a single point of the design space. Therefore the local approximation demands for the design sensitivities of the multibody responses.

The time point constraint method can be used to remove the time-dependence of the multibody problem from the optimization problem formulation. If this method is used in combination with constraint elimination, then the number of multibody responses for which the design sensitivities are to be evaluated can be reduced. The time point constraint and constraint elimination methods can be well combined with the sequential local approximate optimization.

There exist three methods to calculate the design sensitivities needed in the local approximation model; the finite difference method, the direct differentiation method, and the adjoint variable method. These methods are widely applied in the optimization of structures. The finite difference method, a numerical method, is easy to implement but its accuracy depends on the perturbation step and the method is not computationally efficient.

The direct differentiation and the adjoint variable method are combined numerical-analytical methods. This combination of methods increases the accuracy and reduces the computational costs. The unknown factors in these two design sensitivity methods can be calculated analytically or numerically, also called the analytical and semi-analytical method respectively. An advantage of the adjoint variable method over the direct differentiation method is the possibility to calculate the design sensitivity of a selection of multibody responses instead of all responses.

All three design sensitivity analysis methods have been implemented for two kinematically driven multibody systems. These simulations clearly show that the finite difference method is computationally inefficient and less accurate for the design sensitivity analysis of kinematically driven multibody systems. From these simulations also follows that there is little difference between the accuracy and computational costs of the analytical and semi-analytical method respectively. However, the differentiation of the analytical relations is quite demanding and it is difficult to foresee all possible design variables to be defined by a user. The semi-analytical approach seems to be the most interesting.

The integration of the multibody and design sensitivity analysis methods is computationally efficient since many intermediate results from multibody analysis can be re-used in design sensitivity analysis.
Nomenclature

General mathematical symbols
\( \in \) is an element in
\( \equiv \) is equivalent to
\( \forall \) for all
\( da \) total derivative of \( a \)
\( \Delta a \) finite difference of \( a \)
\( \partial a \) partial derivative of \( a \)
\( \dot{a} \) derivative of \( a \) with respect to time
\( \ddot{a} \) second derivative of \( a \) with respect to time
\( \nabla a \) gradient of \( a \)
\( \hat{a} \) approximation of \( a \)
\( x^l \) lower limit for \( x \)
\( x^u \) upper limit for \( x \)
\( x_y \) derivative of \( x \) with respect to \( y \) (specific for this report)

Scalar, vector and matrix notation
\( x \) scalar
\( \mathbf{x} \) vector
\( \mathbf{X} \) matrix

Latin symbols
\( \mathbf{A} \) rotational transformation matrix
\( \mathbf{b} \) design variable vector
\( \mathbf{B} \) derivative of rotational transformation matrix
\( \mathbf{C} \) second derivative of rotational transformation matrix
\( \mathbf{f} \) forces
\( \mathbf{F} \) joint forces multibody system
\( \mathbf{g} \) inequality constraint equations optimization problem
\( \mathbf{h} \) equality constraint equations optimization problem
\( \mathbf{j} \) iteration number
\( \mathbf{J} \) Jacobian matrix
\( \mathbf{K} \) linear stiffness matrix
\( \mathbf{m} \) number of inequality constraints
\( \mathbf{M} \) mass matrix of multibody system
\( \mathbf{n} \) dimension of the design space
\( \mathbf{n_b} \) number of bodies in multibody system
\( \mathbf{p} \) number of equality constraints

continued on next page....
\( q \)  
positions and orientations of multibody system bodies

\( \dot{q} \)  
velocities of multibody system bodies

\( \ddot{q} \)  
accelerations of multibody system bodies

\( q_b \)  
design sensitivity positions multibody system

\( \dot{q}_b \)  
design sensitivity velocities multibody system

\( \ddot{q}_b \)  
design sensitivity accelerations multibody system

\( Q \)  
external forces acting on multibody mechanism

\( r \)  
multibody responses including \( q, \dot{q}, \ddot{q} \) and \( \lambda \)

\( R \)  
joint reaction forces and torques

\( s \)  
distance

\( t \)  
time

\( t_b \)  
start time simulations

\( t_e \)  
end time simulations

\( T \)  
joint torques multibody system

\( u \)  
displacements

**Greek symbols**

\( \alpha \)  
move limits

\( \epsilon_a \)  
approximation error finite difference method

\( \epsilon_n \)  
umerical round-off error finite difference method

\( \epsilon_t \)  
total error finite difference method

\( \gamma \)  
acceleration equations multibody system

\( \lambda \)  
Lagrange multipliers multibody system bodies

\( \lambda_s \)  
Lagrange multipliers adjoint variable method

\( \lambda_b \)  
design sensitivity Lagrange multipliers multibody system bodies

\( \nu \)  
velocity equations multibody system

\( \Phi^k \)  
kinematic constraint equations of multibody system

\( \Phi^d \)  
driver constraint equations of multibody system

\( \Phi \)  
constraint equations multibody system

\( \Psi \)  
nonlinear function

\( \psi \)  
objective function optimization problem

**Scalar and vector differentiation**

\[
\frac{d\psi(y(x),x)}{dx} = \frac{\partial \psi}{\partial y} \frac{dy}{dx} + \frac{\partial \psi}{\partial x}
\]

\[
x_y = \frac{\partial x}{\partial y}
\]

\[
x_y = \begin{bmatrix} \frac{\partial x}{\partial y_1} & \frac{\partial x}{\partial y_2} & \ldots & \frac{\partial x}{\partial y_n} \\ \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} & \ldots & \frac{\partial x_1}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_m}{\partial y_1} & \frac{\partial x_m}{\partial y_2} & \ldots & \frac{\partial x_m}{\partial y_n} \end{bmatrix}
\]

continued on next page...
Abbreviations
In order to simplify the equations the following abbreviations will be used throughout this report:

\[ \Phi \equiv \Phi(q(b, t), b, t) \]
\[ q \equiv q(b, t) \]
\[ \dot{q} \equiv \dot{q}(b, t) \]
\[ \ddot{q} \equiv \ddot{q}(b, t) \]
\[ \lambda \equiv \lambda(b, t) \]
\[ r \equiv r(b, t) \]
# Contents

Summary i

Nomenclature ii

Contents v

1 Introduction to Optimization of Multibody Systems 1
   1.1 Introduction ........................................... 1
   1.2 Numerical optimization .................................. 1
   1.3 Multibody systems ........................................ 2
   1.4 Optimization of multibody systems ........................ 3
   1.5 Local approximation concept .............................. 4
   1.6 Objectives of the research project ........................ 6
   1.7 Structure of the report ................................. 6

2 Formulation of Optimization Problem 7
   2.1 Formulation of optimization problem ..................... 7
   2.2 Sequential local approximate optimization ............... 8
   2.3 Design sensitivity analysis ................................ 9
   2.4 Optimization of time-dependent problems ................. 9
      2.4.1 Integral formulation .................................. 10
      2.4.2 Time point constraints ............................... 10
      2.4.3 Constraint elimination ............................... 11
   2.5 Design sensitivity analysis methods ..................... 12
   2.6 Implemented method ..................................... 13

3 Basic Principles of Design Sensitivity Analysis 14
   3.1 Design sensitivity analysis ............................... 14
   3.2 Finite difference method .................................. 14
   3.3 Direct differentiation ..................................... 16
      3.3.1 Direct differentiation .................................. 16
      3.3.2 Set of linear equations ............................... 16
      3.3.3 Set of nonlinear equations ........................... 17
   3.4 Adjoint variable method .................................. 18
      3.4.1 Adjoint variable method ............................... 18
      3.4.2 Set of linear equations ............................... 19
      3.4.3 Set of nonlinear equations ........................... 19
      3.4.4 Example adjoint variable method ..................... 20
3.5 Conclusions design sensitivity analysis ................................................. 20

4 Design Sensitivity Analysis of Kinematically Driven Systems .................. 21
   4.1 Formulation of multibody systems ................................................... 21
   4.2 Assembly of a multibody system ...................................................... 22
      4.2.1 Solving the set of nonlinear equations ...................................... 22
      4.2.2 Problems in the assembly of multibody system .............................. 23
   4.3 Kinematic analysis ............................................................................. 24
   4.4 Inverse-dynamic analysis ................................................................... 26
   4.5 Design sensitivity analysis ............................................................... 26
      4.5.1 Design sensitivity of positions .................................................... 27
      4.5.2 Design sensitivity of velocities .................................................... 27
      4.5.3 Design sensitivity of accelerations .............................................. 28
      4.5.4 Design sensitivity of joint reaction forces .................................... 28
   4.6 Conclusions of the design sensitivity analysis ...................................... 29

5 Examples of Design Sensitivity Analysis ............................................... 31
   5.1 Examples design sensitivity analysis ................................................... 31
   5.2 Design sensitivity analysis of a four-bar mechanism ............................ 31
      5.2.1 Four-bar mechanism ..................................................................... 31
      5.2.2 Multibody analysis ....................................................................... 32
      5.2.3 Computational costs multibody analysis ....................................... 33
      5.2.4 Design sensitivity analysis .......................................................... 34
      5.2.5 Multiple configurations ............................................................... 40
      5.2.6 System that can not be assembled ............................................... 40
      5.2.7 Lock-up configuration ................................................................... 41
      5.2.8 Bifurcation configuration ............................................................. 42
   5.3 Design sensitivity analysis of a slider-crank mechanism ........................ 43
   5.4 Conclusions ....................................................................................... 44

Bibliography .............................................................................................. 47
Chapter 1

Introduction to Optimization of Multibody Systems

1.1 Introduction

One of the tasks of an engineer is to find the optimal solution to a given problem. Written in a mathematical formulation this means that an optimal set of design variables values has to be found that will minimize some objective function subject to a set of constraints. This can be done by experience and intuition which often leads to acceptable results. However, by introducing computer oriented analytical and numerical methods time and costs can be reduced. Experience and insight into the mechanical problem is then used to appropriately formulate the optimization problem; i.e. the selection of design variables, objective functions, and constraints.

In this chapter the principles of the optimization methods and the optimization of mechanical problems will be explained, especially the optimization of multibody systems.

1.2 Numerical optimization

Numerical optimization is the process of finding the optimum solution to a given problem taking into account the constraints to which the problem is subject. Numerical design optimization includes the following steps to be taken:

- definition of a mathematical model representing the design problem
- formulation of the optimization problem; design variables, objective function, and constraints
- estimation of the initial values for the design variables
- definition of a method as to change the design variable values towards an improved design point (optimization strategy)
- iterative selection of the optimal design variable values using the above defined functions
- implementation of the optimal design variable values in the mathematical model.

The flowchart in figure 1.1 shows this method.

The optimization methods are generally numerical search techniques. From an initial design the value of the objective function is improved with small steps by changing the design point
Chapter 1: Introduction to Optimization of Multibody Systems

in compliance with the constraints. The search for the optimal design is terminated if the improvement of the objective function becomes too slow or the objective function can not be improved without violating one or more constraints.

If there are \( n \) design variables then the optimum is to be searched in a \( n \)-dimensional design space.

Examples of methods to solve optimization problems are the linear programming method, sequential quadratic programming method, and genetic algorithms [1].

![Flowchart numerical optimization process](image)

Figure 1.1: Flowchart numerical optimization process

### 1.3 Multibody systems

A multibody system is a mechanical system which consists of several interconnected bodies and has one or more degrees of freedom. The relative motion between the bodies is limited by joints linking the bodies, for example rotational joints, translational joints, rotational-translational joints, gears, or cam-followers. The motion of the mechanical system is calculated using algebraic kinematic equations of constraint describing limitations on the motion of the mechanism by joints and differential equations of motion based on physical laws. The input at each degree of freedom in the multibody system has to be defined in order to assure a unique solution to the problem.

Examples of multibody systems are the crankshaft-connecting rod-piston mechanism and the suspension in vehicles, the wheel retraction mechanism in airplanes, and the door mechanism in trains. Figure 1.2 shows two examples of multibody systems.
Chapter 1: Introduction to Optimization of Multibody Systems

Figure 1.2: Examples of multibody systems

Multibody mechanisms can be analyzed in three different ways:

- **kinematic analysis**
  The kinematic analysis is used to calculate the motion of the mechanism independent of the forces acting on this mechanism. Given the defined input at the degrees of freedom, the positions of the other elements in the system are determined by solving the set of nonlinear algebraic constraint equations. The velocities and accelerations are determined by solving linear algebraic equations.

- **inverse-dynamic analysis**
  The inverse-dynamic analysis calculates the forces required to generate a prescribed motion. First a kinematic analysis is performed to calculate the accelerations of the bodies in the mechanism given the prescribed motion. These accelerations are then used to calculate the forces needed to generate the motion. The equations are nonlinear algebraic constraint equations.

- **dynamic analysis**
  The dynamic analysis is used to calculate the motion of the bodies in the mechanism given the forces acting on the mechanism, for example the motion of the mechanisms in figure 1.2 due to gravity or other external forces. The equations in this analysis are a combination of algebraic constraint equations and differential equations of motion.

Numerical methods are used to solve the sets of differential equations of motion and/or nonlinear algebraic equations of constraint. These methods require large computational efforts, especially in the case of combined algebraic and differential equations.

Several commercial multibody analysis software packages are available which can automatically generate and solve these equations for a wide range of multibody systems [3,4,19].

1.4 Optimization of multibody systems

Numerical optimization can be applied to both static problems of e.g. structures, and to transient problems of multibody systems. In the latter case the ‘analysis of the mechanical system’ in the flowchart of figure 1.1 should be interpreted as a multibody analysis. The multibody analysis software is generally a general-purpose software package in which the user has no access to any intermediate calculations or results; only the final results are accessible to the user. Therefore, the design optimization requires a coupling between analysis software and optimization software.

A direct link, see figure 1.3, can be difficult to implement as the optimization routine demands for subroutines to evaluate the objective function and constraints, in this case multibody
Chapter 1: Introduction to Optimization of Multibody Systems

analyses. Using standard multibody software it can be difficult to implement the multibody software as a subroutine because the user has generally no access to the source code [1]. Furthermore, standard multibody software does not accept data from an optimization algorithm as input data. Special methods will have to be used to change the design variable values in the problem definition of the multibody software. The link in figure 1.3 can also lead to high computational costs as each change in a design variable value by the optimization algorithm leads to a new and costly multibody analysis.

A further problem in the optimization of multibody systems is the time-dependence of multibody dynamics. Since standard optimization algorithms can not handle this time-dependence of the mathematical problem, the time-dependence has to be removed from the optimization problem formulation.

An example of the optimization of a multibody systems is the optimization of a slider-crank mechanism, represented in figure 1.4 [2]. The mechanism should follow a desired path. The design variables in the optimization problem are the dimensions of the mechanism, indicated with $b_i$ in the figure. The optimization algorithm calculates the dimensions such that the desired path is followed as well as possible while taking into account movability constraints.

1.5 Local approximation concept

The direct link between the multibody analysis and optimization code can be avoided by an approximation model as interface between analysis and optimization code, see figure 1.5 [5]. The advantages of this concept are:

- easier implementation
The optimization algorithm demands for subroutines to evaluate the objective function and constraints, in this case multibody analyses. If the user has no access to the source code of the multibody analysis code, which is generally the case, then it is difficult to implement the multibody software as a subroutine in the optimization algorithm. In case of an approximation model the direct link is avoided and the approximation model is in control of the optimization process instead of the optimization algorithm itself.

- less computational effort

The approximation concept requires a limited amount of multibody analyses to setup the approximation model. The multibody responses needed by the optimization code are then approximated instead of calculated by performing computational costly multibody analysis.

- more control by the user on the optimization process

Using the approximation model the user has more insight and control on the optimization process than in case of a direct link; a “black box” link. The user can intervene if necessary, for example in the case of instabilities, too large variations of the design variable values, or oscillations.

![Figure 1.5: Approximation concept](image)

Starting point of this project is that the direct link is replaced by a local approximation model. In this approximation model the objective and constraints functions are approximated using a first order approximation model. Then a sequence of approximations of objective functions and constraints is made. This is called sequential approximation optimization, this method is shown in figure 1.6.

![Figure 1.6: Sequential optimization using local linear approximations](image)

First an improved design point is searched for within the region of validity of the local approximation model. Then around this new improved design point a new local approximation
model is setup and again a new improved design point is searched for, repeating the optimization cycle.

1.6 Objectives of the research project

Several effective methods are known and already implemented to calculate the design sensitivity in structural optimization problems. Starting point in this research project is the sequential local approximate optimization of multibody problems. The setup of the local approximation model demands for the sensitivity of the multibody responses with respect to the design variables. The objective of this project is to find out if and how the design sensitivity analysis methods successfully developed in structural optimization problems can be implemented in the design sensitivity analysis for the sequential approximate optimization of multibody systems.

The advantages and disadvantages of the different methods to calculate the design sensitivity and the implementation of these methods in the sequential local approximate optimization of multibody dynamics will be described in this report.

1.7 Structure of the report

The optimization problem formulation for multibody dynamics, the problems related to this type of optimization, and possible solutions to these problems will be presented in chapter 2.

In chapter three the basic principles of the design sensitivity analysis developed for structural optimization will be discussed. The differences between the different design sensitivity analysis methods regarding accuracy, computational costs, and implementation will be explained. The design sensitivity analysis methods used in structural optimization can also be applied to the optimization of multibody systems. This will be shown in the fourth chapter.

During this project mainly kinematically driven multibody systems with holonomic constraints have been studied. The implementation of the design sensitivity analysis in these multibody systems will be explained in chapter four.

Two multibody systems have been used to implement the different methods of design sensitivity analysis. The results of these test will be presented in chapter five.

Chapter six presents the conclusions and points of discussion following from this research project.
Chapter 2

Formulation of Optimization Problem

2.1 Formulation of optimization problem

In order to solve a multibody design problem using general-purpose optimization methods, the design problem must be formulated as an optimization problem. The different methods to formulate the optimization problem, the use of approximation models, and the constraint elimination method will be presented.

The optimization problem can be formulated for multibody systems as:

Minimize an objective function $\psi$:

$$\psi(b) = \psi(q(b, t), \dot{q}(b, t), \ddot{q}(b, t), \lambda(b, t), b, t) \quad t \in [t^b, t^e]$$

subject to $m$ inequality constraints and $p$ equality constraints:

$$g_j(b, t) = g_j(q(b, t), \dot{q}(b, t), \ddot{q}(b, t), \lambda(b, t), b, t) \leq c_j \quad \forall t \in [t^b, t^e]$$

$$h_k(b, t) = h_k(q(b, t), \dot{q}(b, t), \ddot{q}(b, t), \lambda(b, t), b, t) = d_k \quad \forall t \in [t^b, t^e]$$

and restricted in the $n$-dimensional design space:

$$b_i^b \leq b_i \leq b_i^u \quad i = 1, ..., n$$

where $b$ is the vector with the design variables. The vectors $q, \dot{q}, \ddot{q},$ and $\lambda$ are the position, velocity, acceleration, and Lagrange multiplier vectors of the multibody system, respectively. They follow from the multibody analysis. The multibody responses $q, \dot{q}, \ddot{q}, \lambda$ will from now on be written as $r$ in this report for simplification of the equations.

Many different formulations for the objective function and constraints are possible. Several examples of formulations will be given. The first formulation is denoted as:

$$\psi(b) = F_1(r(b, t^e), b, t^e) + \int_{t^b}^{t^e} F_2(r(b, t), b, t)dt \leq c \quad \forall t \in [t^b, t^e]$$
for constraints, or
\[
\psi(b) = F_1(r(b, t^e), b, t^e) + \int_{t^b}^{t^e} F_2(r(b, t), b, t)dt \quad t \in [t^b, t^e]
\] (2.6)

for objective functions. In this formulation the configuration of the system at \( t = t^e \) and the motion of the system over the entire time span \([t^b, t^e]\) are taken into account. An example of this kind of formulation is the ride comfort objective function in the optimization of vehicle suspensions where the objective is to minimize the overall vertical acceleration of the driver's seat:

\[
\psi(b) = \int_{t^b}^{t^e} |\dot{q}_{\text{seat}}(b, t)| dt \quad t \in [t^b, t^e]
\] (2.7)

But also constraints can be specified using this formulation, for example to specify the admissible overall vertical acceleration of the driver's seat.

Another possible formulation for the objective function is:

\[
\psi(b) = \sum_{i=0}^{n_t} \left\{ r^a(b, t_i) - r^d(b, t_i) \right\}^2 \quad i = 1, ..., n_t
\] (2.8)

where \( r^a \) is the actual motion of the system and \( r^d \) is the desired motion of the system. \( n_t \) is the number of time points into which the time span \([t^b, t^e]\) is divided. The objective is have a minimal difference between the actual motion of the system and the desired motion of the system at the discrete time points \( t_i \). This type of formulation can be used to optimize a path-follower mechanism such as indicated in figure 1.4.

A constraint in the example given in figure 1.4 could be the limitation of the horizontal motion of the slider:

\[
g(b, t) = q_2(b, t) \leq c \quad \forall t \in [t^b, t^e]
\] (2.9)

where \( q_2 \) is the horizontal position of the slider. For all time points the horizontal position of the slider should not exceed a certain limit in movability space, for example to avoid collision with an obstacle. In this case an inequality constraint is used to formulate the constraint. An equality constraint could be used to formulate the constraint that a specific point of a body in the mechanism must pass through a certain point to pick up an element.

For example, the limitations on the design space in equation (2.4) can be used to assure physically possible solutions.

### 2.2 Sequential local approximate optimization

During this project the local approximation model has been used as this research project is a part of a larger research project on the sequential local approximation optimization of multibody systems. The local approximation model is used to approximate the multibody responses, as illustrated in figure 1.5.

In the sequential local approximate concept the objective and constraint functions are approximated using a local linear approximation model. These approximations are only valid in
a limited part of the design space. In this limited part of the design space, an optimal solution to the optimization problem is calculated using these local approximations. Around this local optimum a new approximation model is calculated, valid for a new limited part of the design space. The optimization cycle is repeated.

Using the local approximation model, the optimization formulation can be written as:

Minimize an objective function $\hat{\psi}(b) = \psi(\tilde{r}(b, t), b, t)$

subject to $m$ inequality constraints and $p$ equality constraints:

$$g_j(b, t) = g_j(\tilde{r}(b, t), b, t) \leq c_j \quad \forall t \in [t^b, t^e] \quad j = 1, ..., m$$

$$h_k(b, t) = h_k(\tilde{r}(b, t), b, t) = d_k \quad \forall t \in [t^b, t^e] \quad k = 1, ..., p$$

and restricted in a limited part of the n-dimensional design space:

$$\alpha^l_i \leq b_i \leq \alpha^u_i \quad i = 1, ..., n$$

Vector $\tilde{r}$ contains the local linear approximations for the multibody responses $r$:

$$\tilde{r}(b, t) = r(b_0, t) + \sum_{i=1}^n (b_i - b_{0i}) \left( \frac{\partial r(b, t)}{\partial b_i} \right)_{b_0}$$

The size of the search region is limited by move limits defining a region around the initial design point in which the local approximations are valid, see figure 1.6.

### 2.3 Design sensitivity analysis

The setup of the local approximation model demands for the sensitivities of the multibody responses with respect to the design variables. In equation (2.14) it can be seen that the design sensitivity of the multibody responses is needed:

$$\left. \frac{\partial r(b, t)}{\partial b_i} \right|_{b_0} = i = 1, ..., n$$

### 2.4 Optimization of time-dependent problems

As indicated in the first chapter, general optimization routines can not handle time-dependent problems such as occur for multibody systems. Two methods to remove the time-dependence from the problem formulation in equations (2.1)-(2.4) will be presented; the integral formulation and the time point constraint method.
2.4.1 Integral formulation

The first possibility is the so-called integral formulation [18]. The basic idea is to represent the entire constraint violation of the time-span \([t^b, t^e]\) by a single value. This can be represented by the following formulation. The original constraint:

\[
g(b, t) = g(r(b, t), b, t) \leq c \\
\forall t \in [t^b, t^e]
\]

is replaced by:

\[
g^*(b) = \int_{t_0}^{t_e} \hat{g}(r(b, t), b, t) dt \leq c^*
\]

where \(\hat{g}\) is defined as:

\[
\hat{g} = max\{g, c\} = \begin{cases} 
g & \text{if } g \geq c, \\
0 & \text{if } g < c
\end{cases}
\]

The \(max\) function takes care that only the constraint violations are included. Figure 2.1 illustrates the concept of equation (2.18) and the functions \(g(b, t)\) and \(g^*(b)\).

![Figure 2.1: Integral formulation](image)

The advantage of this method is that the overall constraint violation is represented by a single value and the number of constraints is not augmented. A major disadvantage is however the possible introduction of discontinuities in the derivatives of the constraints by the combination of integral formulation and \(max\)-function in equation (2.17). This discontinuity can lead to numerical difficulties. For some dynamic response optimization problems these difficulties were encountered in [18]. In some of these problems the optimality condition could not be met.

There exist other integral formulations than the integral formulation using the \(max\) function. All these formulations however have the disadvantage of possible introduction of discontinuities into the optimization process.

2.4.2 Time point constraints

In the time point constraint method the entire time span \([t^b, t^e]\) is discretized into \(n_t\) time points [1]. The time-dependent constraint:

\[
g(b, t) = g(r(b, t), b, t) \leq c \\
\forall t \in [t^b, t^e]
\]

is replaced by \(n_t\) constraints:

\[
g_i(b, t_i) = g_i(r(b, t_i), b, t_i) \leq c \\
i = 1, \ldots, n_t
\]
Figure 2.2 illustrates this method. The number of time points $n_t$ should be large enough to avoid large constraint violations between two time points.

The number of constraints is increased significantly in this method to eliminate the time-dependence. But not all constraints need to be taken into account, the number of constraints can be reduced by considering only active or potentially active constraints. All other constraints can be eliminated from the problem.

### 2.4.3 Constraint elimination

Most optimization routines use a so-called active set of constraints; only the active or potentially active constraint are considered in the optimization algorithm. This reduces the computational efforts with respect to the optimization. Figures 2.3 and 2.4 depict the use of active or potentially active constraints, figure 2.3 exemplifies a time-dependent constraint. This constraint is discretized using the above-described methods and only the active or potentially active constraint are taken into account in the optimization problem, these constraint are marked in grey in the figure 2.4. The level $\alpha$ in figures 2.3 and 2.4 indicates the level at which the constraints are considered to be potentially active. In this example the time-dependent constraint will be replaced by 21 time-independent constraints (the number of grey bars in figure 2.4).

Figure 2.3: Time-dependent constraint

It is possible to further reduce the number of constraints to be taken into account in the optimization cycle if the time point constraint elimination is applied. Consider the time-independent
constraints in figure 2.4. Not all active or potentially active constraints need to be taken into account. It is possible to select only the constraint in a local maximum of constraint violations or local maximum of possible constraint violations. These constraints are marked dark grey in the figure 2.5. Selecting also some constraints around these local maxima, the evolution of the local maxima can be analyzed for maximum insight into the evolution in time of the constraint. These extra constraints are marked light grey in the figure. The constraints at the start and end time of the optimization problem are also analyzed in order to foresee possible constraint violations at the time boundaries. The examples in figures 2.3-2.5 show that the number of constraints can be reduced. In this example the number of constraints to be evaluated is reduced from 31 to 21 by selecting active or potentially active constraints. The number of constraint evaluations is further reduced to 13 applying the constraint elimination. The figures only illustrate the methods. Generally the time span will be discretized into 200 to 300 time point. In this case a reduction of 90% in constraint evaluations is possible.

The degree of constraint elimination can be increased or reduced by changing the level $\alpha$. The level $\alpha$ marks the level above which constraints are marked being (potentially) active, illustrated in figures 2.3-2.5. Furthermore, the number of discrete time point into which the time span is divided can be adapted. However, too a large reduction of the number of (potentially) active constraints can lead to important constraint violations.

2.5 Design sensitivity analysis methods

In literature two different methods have been proposed to calculate the design sensitivity of multibody systems; the method proposed by Haug and Hsiesh [16,18] and the method proposed by Bestle [6]. Bestle uses the generalized coordinate system to define the multibody system. This method leads to a small set of highly nonlinear equations of motion where the number of equations equals the degrees of freedom of the multibody system. Bestle uses symbolic parameter variation to calculate the design sensitivity of this small set of equations. All design sensitivities can be related to the design sensitivity of the degrees of freedom.
Haug [16] uses the Cartesian coordinate system to define the multibody problem, resulting in a large set of combined equations of motion and algebraic equations of constraints for the multibody system. The design sensitivity of the motion of each body in the mechanism is calculated.

2.6 Implemented method

In the following chapters the method as proposed by Haug, both for the multibody system and the design sensitivity analysis will be implemented using the local approximation model. The Cartesian coordinate has been chosen because this coordinate system is easy to implement, especially for general-purpose programs. Furthermore, it is possible to use the constraint elimination method to reduce the number of design sensitivity analyses because the design sensitivity is calculated for each body in the mechanism. This reduces the computational effort. Therefore the combination of the local approximation model and the constraint elimination method can be interesting if the method proposed by Haug is used.

In the next chapter first the design sensitivity analysis as implemented in the optimization of structures will be discussed. The setup of the local approximation model demands for these sensitivities. Several methods are applied successfully in the optimization of structures. These methods can also be applied to kinematically driven multibody systems.
Chapter 3

Basic Principles of Design Sensitivity Analysis

3.1 Design sensitivity analysis

The optimization of multibody problems using the sequential local approximate optimization method demands for the design sensitivity of the multibody responses with respect to the design variables. This design sensitivity of the multibody responses is needed to setup the local approximation model. Calculation of these derivatives is often very costly with respect to the computational time. Therefore an efficient and accurate calculation of the design sensitivities is important.

In the optimization of structural or multibody problems, the objective function and constraints are often not explicitly known as a function of the design variables. The design sensitivities can not be obtained by simple differentiation of the objective function and constraints with respect to the design variables. Numerical, symbolical, or combined numerical-symbolical methods will be needed to evaluate these derivatives.

In this chapter the basic principles of several different methods to calculate the sensitivities with respect to the design variables will be explained and illustrated with examples. These examples are time-independent. The time-dependent design sensitivity analysis of multibody systems will be discussed in chapter 4.

3.2 Finite difference method

The most straightforward method for the design sensitivity analysis is the finite difference method. This numerical differentiation method estimates the derivatives using multiple function analysis with small design variable value variations.

Consider a function:

\[ \psi = \psi(b) \]

From a reference point \( b_0 \), at which the design sensitivity is to be evaluated, the design variable value is varied over a small distance \( \Delta b \) such that the function is evaluated at \( \psi(b_0 + \Delta b) \). The first order Taylor-series expansion of the function \( \psi(b_0 + \Delta b) \) yields:

\[ \psi = \psi(b_0 + \Delta b) \approx \psi(b_0) + \frac{d\psi}{db}\bigg|_{b_0} \Delta b \]

(3.2)
Chapter 3: Basic Principles of Design Sensitivity Analysis

or:

\[
\left. \frac{d\psi}{db} \right|_{b_0} = \frac{\psi(b_0 + \Delta b) - \psi(b_0)}{\Delta b}
\]  

(3.3)

where \( \Delta b << b_0 \)

Typical optimization problems will have more than one design variable. The finite difference method in equation (3.3) will have to be performed for each of the \( n \) design variables in the optimization problem. This is an important drawback in the use of this method as it is rather inefficient especially in the case of multibody systems where each new function analysis demands a computationally expensive numerical analysis. For multiple design variables the design sensitivity vector can be denoted as:

\[
\frac{d\psi}{db} \bigg|_{b_0} = \sum_{t=1}^{n} \left( \frac{\psi(b_0 + \Delta b \cdot e_i) - \psi(b_0)}{\Delta b} \cdot e_i \right)
\]

(3.4)

where \( e_i \) is a column-vector with a one on the \( i^{th} \) position and zeros elsewhere.

With respect to the accuracy of the design sensitivity there is also an important drawback in the use of the finite difference method; that is the choice of the design variable value variation \( \Delta b \). This variation should neither be too large nor too small. Too large a variation \( \Delta b \) will lead to an approximation error \( \epsilon_a \). If the variation \( \Delta b \) is too small round-off errors \( \epsilon_n \) related to the floating-point accuracy of the computational system will occur. These errors are depicted in figure 3.1 in which the exact calculation and the finite difference approximation of the derivatives of a nonlinear system were compared while changing the finite difference step size \( \Delta b \).

![Finite difference approximation error](image)

**Figure 3.1:** Finite difference approximation error

In this figure can be seen that the total finite difference error \( \epsilon_t = \epsilon_a + \epsilon_n \) has a minimum for a certain variation \( \Delta b \). The approximation error \( \epsilon_a \) is proportional to the variation \( \Delta b \) and the round-off error \( \epsilon_n \) is inverse-proportional to \( \Delta b \). The overall finite difference error, \( \epsilon_t = \epsilon_a + \epsilon_n \), is minimal if the variable variation \( \Delta b \approx \sqrt{\epsilon_c} \), where \( \sqrt{\epsilon_c} \) is the floating-point accuracy of the
computational system [6]. Since close to the optimum in an optimization problems the difference in function values is generally very small, the round-off errors can then become very large.

The finite difference method explained above is the forward finite difference method. The approximation error can be reduced using the central or higher order finite difference methods. However the improvement in accuracy using these methods is not sufficient for optimization problems and the computational times will increase using higher order finite difference methods. Therefore these methods have not been considered furthermore.

3.3 Direct differentiation

3.3.1 Direct differentiation

The conclusion that can be drawn from the above section is that just a numerical differentiation will not result in an efficient and accurate design sensitivity evaluation [6]. Additional analytical information will be needed for the sensitivity analysis in multibody systems. Two methods exist in which this extra information is used for an efficient design sensitivity evaluation.

The first method is the direct differentiation method. This method will be explained by means of two sets of equations, a set of linear equations and a set of non-linear equations.

3.3.2 Set of linear equations

Consider a system of linear equations denoted:

\[ K(b)u = f(b) \]  \hspace{1cm} (3.5)

where \( K \) and \( f \) are dependent on the design variables \( b \). The sensitivity \( \frac{dK}{db} \) with respect to one of the design variables \( (b \in b) \) is to be evaluated. Differentiation of the equation 3.5 with respect to the design variable \( b \) gives:

\[ K \frac{dK}{db} + \frac{dK}{db}u = \frac{df}{db} \]  \hspace{1cm} (3.6)

The equation can be rearranged to:

\[ \frac{du}{db} = K^{-1} \left[ \frac{df}{db} - \frac{dK}{db}u \right] \]  \hspace{1cm} (3.7)

For most design applications the load vector \( f \) is not dependent on the design variable values \( b \) or is explicitly known. For example in the analysis of structures, the forces acting on the system will generally not depend on the dimension of the elements in the system. The vector \( u \) follows from a solver used to resolve the set of equations in equation (3.5), e.g. the LU-decomposition. The only unknown term in equation (3.7) is \( \frac{dK}{db} \).

The analysis needed to calculate this design sensitivity can be performed in two ways. The design sensitivity can be evaluated analytically or can be estimated using the finite difference method.

Analytical direct differentiation

Using the analytical direct differentiation method the design sensitivity \( \frac{dK}{db} \) is obtained by analytical differentiation of the matrix \( K \) with respect to the vector \( b \):

\[ \frac{dK(b)}{db} \]  \hspace{1cm} (3.8)
The assembly of matrix $\mathbf{K}$ in equation (3.5) is followed by the -analytical- assembly of the derivative matrix $\frac{d\mathbf{K}}{db}$. In the symbolic elements library not only the sub-matrices $\mathbf{K}^e$ for the assembly of the matrix $\mathbf{K}$ must be present but also the sub-matrices $\frac{d\mathbf{K}^e}{db}$ for the assembly of the matrix $\frac{d\mathbf{K}}{db}$. The elements library must contain the derivative of the sub-matrices $\mathbf{K}^e$ with respect to all possible design variables.

**Semi-analytical direct differentiation**

The analytic differentiation of large sets of equations can be quite cumbersome, especially for large and nonlinear systems. Furthermore, the derivatives with respect to all possible design variables must be calculated. Therefore, the design sensitivity of the stiffness matrix can also be estimated at a fixed point $b_0$ in the design space by means of the finite difference method:

$$\frac{d\mathbf{K}(b)}{db} = \left(\frac{\mathbf{K}(b_0 + \Delta b) - \mathbf{K}(b_0)}{\Delta b}\right)$$  \hspace{1cm} (3.9)

Compared to the analytical direct differentiation, this semi-analytical method is less accurate and is subject to the same drawbacks as the finite difference method for the overall sensitivity with respect to the approximation and round-off errors. The major advantage of the semi-analytic method compared to the analytical direct differentiation is that the derivatives of the system's set of equations needs not to be formulated analytically.

The semi-analytic direct differentiation is more accurate compared to the finite difference method as only inaccuracies related to the estimation of the derivative $\frac{d\mathbf{K}(b)}{db}$ are introduced into the calculations. Using the finite difference method the inaccuracies related to solving the set of equations in equation 3.5 are also introduced in the design sensitivity analysis.

The direct differentiation methods, both the semi-analytical and analytical method, are more computationally efficient than the finite difference method. A reduction to $1/10^{th}$ of the computational time is possible while maintaining or increasing the overall accuracy.

### 3.3.3 Set of nonlinear equations

Now a set of nonlinear equations is considered

$$\Psi(r(b), b) = 0$$  \hspace{1cm} (3.10)

Differentiation of this equation with respect to one of the design variables results in:

$$\frac{\partial \Psi}{\partial r} \frac{dr}{db} + \frac{\partial \Psi}{\partial b} = 0$$  \hspace{1cm} (3.11)

and rearrangement leading to:

$$\frac{dr}{db} = -\left(\frac{\partial \Psi}{\partial r}\right)^{-1} \frac{\partial \Psi}{\partial b}$$  \hspace{1cm} (3.12)

The inverse of the Jacobian matrix, $\frac{\partial \Psi}{\partial r}^{-1} = J^{-1}$, is already calculated in the Newton-Raphson algorithm. This algorithm is used to solve the equation (3.10) using equation (4.5). Only if the Newton-Raphson algorithm has been successful then the inverse of the Jacobian matrix can be used in the design sensitivity analysis. The only unknown factor in this equation is the design sensitivity $\frac{\partial \Psi}{\partial b}$ which can be calculated by means of the analytic and the semi-analytic method as indicated in the above section. In case of this set of non-linear equations, the semi-analytic
Chapter 3: Basic Principles of Design Sensitivity Analysis

direct differentiation is far more accurate compared to the finite difference method. The set of non-linear equations in equation (3.10) is solved using iterative solve methods with a limited accuracy. Using the finite difference method this limited accuracy is also introduced into the estimation of the design sensitivity and as such the design sensitivity accuracy depends on the accuracy of the solver.

An interesting aspect was found while applying the semi-analytic method to multibody dynamics. Although the set of equations in multibody dynamics can be highly nonlinear, the design variables are however very often linear in the set of multibody equations. Using the finite difference method, the nonlinearity of the set of multibody equations introduces inaccuracies into the design sensitivity analysis. However in the case of the semi-analytic direct differentiation method the finite difference method is applied to a linear system. The approximation error will be very small and independent of the finite difference step size as long as this step size is large enough such that no computational round-off errors are introduced. In chapter 4 and 5 this linearity of the design variables in the multibody equations will be illustrated.

3.4 Adjoint variable method

3.4.1 Adjoint variable method

The direct differentiation calculates the design sensitivity of all the elements in a vector. In many cases not all the design sensitivities of a vector are needed but only of a few elements in the vector. In this case the adjoint variable method can be more efficient than the direct differentiation method.

Consider the following optimization problem:

Minimize the objective function $\psi$

$$
\psi(b) = \psi(r(b), b)
$$

subject to $m$ inequality constraints on the multibody code results:

$$
g_j(b) = g_j(r(b), b) \leq c_j \quad j = 1, ..., m
$$

restricted in the $n$-dimensional design space:

$$
\hat{b}_i^l \leq b_i \leq \hat{b}_i^u \quad i = 1, ..., n
$$

The optimization problem with $n$ design variables will be limited in the $n$-dimensional design space by a maximum of $n$ active constraints. If the number of constraint equations $m$ is larger than the number of design variables $n$ still a maximum of $n$ constraints will be active and need to be evaluated. The number of design sensitivity analyses can be reduced.

In case of an optimization problem using a local approximation model the same reduction of sensitivity analysis can be used. The active and potentially active constraints are approximated using a local approximation model. In order to setup the local approximation model the sensitivity of the multibody responses with respect to the design variables are needed. The constraints do not need to depend on all multibody responses $r$ and one or more constraints may depend on the same multibody response. Generally only part of the multibody responses will be used in the active or potentially active constraints. Again the number of design sensitivity evaluations can be reduced.
Chapter 3: Basic Principles of Design Sensitivity Analysis

3.4.2 Set of linear equations

In the optimization problem formulation of equations (3.13)-(3.15) the constraint equations can be differentiated with respect to the design variables:

\[
\frac{dg}{db} = \frac{\partial g}{\partial b} + \lambda^T K \frac{du}{db} = \frac{\partial g}{\partial b} + \lambda^T \left( \frac{df}{db} - \frac{dK}{db} u \right)
\]  
(3.16)

or:

\[
\frac{dg}{db} = \frac{\partial g}{\partial b} + z^T d\lambda
\]

(3.17)

where \( z \) is defined as

\[
z_i = \frac{\partial g}{\partial a_i} \quad i = 1, ..., m
\]

(3.18)

Combining equations (3.17) and (3.18) the adjoint vector \( \lambda \) can be defined:

\[
K\lambda = z \rightarrow \lambda = \lambda^{-1}z
\]

(3.19)

This adjoint vector \( \lambda \) for the adjoint variable method is not to be confused with the Lagrange vector \( \lambda \) used in the multibody analysis. The constraint design sensitivity can be written as:

\[
\frac{dg}{db} = \frac{\partial g}{\partial b} + \lambda^T \left( \frac{df}{db} - \frac{dK}{db} u \right)
\]

(3.20)

The design sensitivity e.g. equation (3.19) for each (potentially) active constraint \( g \) instead of for each design variable \( b \). This method can be advantageous if the number of constraints is smaller than the number of design variables.

3.4.3 Set of nonlinear equations

Let us consider again the set of nonlinear equation in equation (3.10). The design sensitivity can be written:

\[
\frac{dr}{db} = -J^{-1} \frac{\partial \Psi}{\partial b}
\]

(3.21)

where

\[
J = \frac{\partial \Psi}{\partial r}
\]

(3.22)

The equation (3.10) is solved using the Newton’s method. If the approximation for \( r \) using Newton’s method is close enough to the exact solution, then \( J \) can be used to calculate the derivatives. The adjoint variable vector can be calculated using:

\[
J^T \lambda = z \rightarrow \lambda = J^{-T}z
\]

(3.23)

according to equation (3.18). The design sensitivity of the constraint can be calculated instead of the design sensitivity for each design variable:

\[
\frac{dg}{db} = \frac{\partial g}{\partial b} - \lambda^T \frac{\partial \Psi}{\partial b}
\]

(3.24)
3.4.4 Example adjoint variable method

This example illustrates the use of the adjoint variable method:

Minimize the objective function \( \psi \) (arbitrarily chosen):
\[
\psi(b) = r_1(b) + r_3(b)
\]
restricted in the \( n \)-dimensional design space:
\[
b_i^l \leq b_i \leq b_i^u \quad i = 1, ..., 2
\]
subject to \( m \) inequality constraints (also arbitrarily chosen):
\[
g_1(b) = r_1(b) \leq c_1
\]
\[
g_2(b) = r_6(b) \leq c_2
\]
and the size of the state space vector \( r \) is 6:
\[
r = [r_1, ..., r_6]
\]

In this example only the design sensitivity of the element \( r_1, r_3, \) and \( r_6 \) in the vector \( r \) are needed. The adjoint variable method can be used instead of calculating the design sensitivity of all the elements in the vector \( q \) using the direct differentiation method and afterwards selecting the design sensitivity of the three elements.

First the adjoint vectors \( \lambda \) are calculated for each constraint using:
\[
\lambda_1 = K^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ for } r_1
\]
\[
\lambda_3 = K^{-1} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \text{ for } r_3
\]
\[
\lambda_6 = K^{-1} \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ for } r_6
\]
for linear equations and for non-linear equations
\[
\lambda_1 = J^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ for } r_1
\]
\[
\lambda_3 = J^{-1} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \text{ for } r_3
\]
\[
\lambda_6 = J^{-1} \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ for } r_6
\]
The ones in the matrix in equations (3.29) and (3.30) indicate for which elements the design sensitivity is to be calculated.

Once the adjoint vectors are calculated the design sensitivity of the constraints can be calculated using equation (3.20) or equation (3.24).

3.5 Conclusions design sensitivity analysis

The finite difference method can be well used for small and linear problems. This method is however not accurate enough and demands for too much computational time to be used effectively in multibody systems [6]. The main difference between the direct differentiation method and the adjoint variable method is that the design sensitivity is calculated for each design variable using the direct differentiation method while the adjoint variable method is used to calculate the design sensitivity for each constraint.

The adjoint variable method is more effective if the number of constraint is smaller than the number of design variables.
Design Sensitivity Analysis of Kinematically Driven Systems

4.1 Formulation of multibody systems

The kinematic analysis of multibody systems is the process of calculating the positions, velocities and accelerations of the individual bodies of the mechanism. A set of Cartesian generalized coordinates $q_i = [x, y, \phi]_i^T$ can be used to define the position and orientation of each body $i$, see figure 4.1. The configuration of a planar multibody system can be described by means of the position vector $q = [q_1, q_2, ..., q_{n_b}]$, where $n_b$ is the number of bodies in the multibody system.

![Figure 4.1: Example of a planar multibody system](image)

The multibody system consists of several bodies which are linked such that relative motion is possible. For the purpose of this project only rigid bodies will be used. The motion of each individual body can be restricted by absolute constraints on the motion on the body itself, and by relative constraints on the motion between two or more bodies. These constraints are called kinematic constraints. During this research project only stationary holonomic kinematic multibody constraints will be considered; i.e. multibody constraints in which time does not appear explicitly in the kinematic constraint equations. These kinematic constraint equations can be
Chapter 4: Design Sensitivity Analysis of Kinematically Driven Systems

written as:
\[ \Phi^b(q(b,t), b) = 0 \]  
(4.1)

If the system has \( n \) bodies and \( l \) kinematic constraint equations then the (planar) multibody system has \( m = 3n - l \) degrees of freedom. Since the movement of the system has to be defined uniquely, \( m \) additional driver constraints have to be specified which prescribe the motion of one or more elements in the system, denoted as:
\[ \Phi^d(q(b,t), b, t) = 0 \]  
(4.2)

The kinematic constraint equations and the driving constraint can be combined into one set of constraint equation for the multibody system:
\[
\Phi(q(b), b, t) = \begin{bmatrix}
\Phi^b_1(q(b,t), b) \\
\vdots \\
\Phi^b_l(q(b,t), b) \\
\Phi^d_1(q(b,t), b, t) \\
\vdots \\
\Phi^d_m(q(b,t), b, t)
\end{bmatrix} = 0
\]  
(4.3)

describing the configuration of the system for each time point \( t \).

In order to simplify the equation in this chapter, the function \( \Phi(q(b,t), b, t) \) will be written as \( \Phi \) or \( \Phi(q) \). Partial derivatives will be denoted as \( \frac{\partial}{\partial q} \) instead of \( \frac{\partial}{\partial y} \).

4.2 Assembly of a multibody system

4.2.1 Solving the set of nonlinear equations

The state space vector \( q \) describing the configuration of the multibody system can be solved from equation (4.3) for each time point \( t \) under the condition that the Jacobian \( \Phi_q \) is non-singular over the time span \([t^b, t^e]\):
\[
|\Phi_q(q(b), b, t)| \neq 0 \quad t \in [t^b, t^e] \]  
(4.4)

If the Jacobian is singular for one or more time point, than there exists no unique solution for the motion of the system, or the motion is not possible.

The set of kinematic constraint equations in (4.3) is generally nonlinear for multibody problems. Iterative solvers such as the Newton-Raphson algorithm have to be used to solve the set of equations. The Newton-Raphson algorithm can be denoted for this type of equations as:
\[
q^{j+1} = q^j - \left(\Phi(q)_q^j\right)^{-1}\Phi(q)^j
\]  
(4.5)

The iteration is successful if the magnitude of all errors and changes in the estimated solutions satisfy:
\[
\begin{align*}
|\Phi_k(q^j)| & \leq \varepsilon_1 \\
|q_k^j - q_k^{j-1}| & \leq \varepsilon_2
\end{align*}
\]  
(4.6)

where \( \varepsilon_1 \) and \( \varepsilon_2 \) are preset allowable errors in the estimation of \( q \). \( j \) is the iteration number.
Chapter 4: Design Sensitivity Analysis of Kinematically Driven Systems

4.2.2 Problems in the assembly of multibody system

There is not always a unique solution to the set of equations in (4.3). Four problems can occur during the assembly or the motion of the multibody system:

- multiple configurations
- system can not be assembled for any time point
- lock-up
- bifurcation

Multiple configurations at setup

The vector $q^0$ is an initial estimation for the configuration of the multibody system, generally at $t = t^b$. This initial estimation has to be close enough to the actual desired configuration. Otherwise an unwanted configuration of the multibody system can be found if there exist several solutions for the equation (4.3), see figure 4.2. Both configurations in the figure are solutions for the set of multibody equations of this problem.

![Figure 4.2: Configurations of multibody system](image)

In the following time point iterations the estimation for the multibody system is calculated using equation (4.7). If the time steps $\Delta t$ are small enough then the estimated configuration will be close to the desired configuration. The problems related to multiple possible configurations are generally only encountered at the first setup of the multibody system ($t = t^b$).

Multibody system can not be assembled for any time point

It is possible that the elements in the system are of dimensions which do not allow assembly for any time point $t$. In this case the first condition in equation (4.6) can not be met for any time point. Figure 4.3 shows an example of a system which can not be assembled for any time point.

Lock-up configuration

Lock-up is the singular configuration beyond which motion can no longer continue. Figure 4.4 shows a lock-up situation. If the rotational motion of the crank on the left side of the figure is
Figure 4.3: Multibody system cannot be assembled for any time point

Figure 4.4: Lock-up configuration four-bar mechanism

continued, a lock-up situation will be encountered as shown on the right side of the figure. The system cannot move anymore.

The lock-up configuration is characterized by velocities $\dot{q}$ and/or accelerations $\ddot{q}$ approaching infinity near the lock-up situation [3]. In chapter five it will be shown that the increase of the velocity and/or acceleration towards infinity is an indication for a lock-up situation.

**Bifurcation configuration**

Bifurcation is the singular configuration of the system after which branching of motion to two different paths occurs. The configuration of the multibody is such that two solutions exist for the set of nonlinear equations in (4.3) and for which the solutions are close to each other. Consider the four-bar mechanism in figure 4.5. The mechanism has two bars connected to the ground. If one of the bodies is rotated then two paths of motion are possible.

Figure 4.5: Bifurcation configuration four-bar mechanism

Computationally the bifurcation configuration can be checked by analyzing the determinant $|\Phi_q|$. The determinant of the two possible solutions after the bifurcation point differ from sign [3].

**4.3 Kinematic analysis**

Figure 4.6 shows the general structure of a program to analyze the motion and the design sensitivity of a kinematically driven multibody system. The figure also depicts which analyses need to be performed first when a certain analysis is desired, e.g. if the design sensitivity of the velocities is needed then first the positions, velocities, and design sensitivity of the positions
need to be evaluated. The different analyses in figure 4.6 will be explained in details in the following sections.

**Position analysis**

The state space vector \( \mathbf{q} \) describing the configuration of the multibody system can be solved from equation (4.3) using the Newton-Raphson algorithm as described in the preceding section. The initial estimation \( \mathbf{q}_0 \) has to be estimated manually for the first time point iteration at \( t = t^b \). In subsequent time iteration points (\( t > t^b \)), an estimation for the system's configuration at proceeding time iteration point \( t + \Delta t \) can be made:

\[
\mathbf{q}_{t+\Delta t} = \mathbf{q}_t + \dot{\mathbf{q}}_t \Delta t + \frac{1}{2} \ddot{\mathbf{q}}_t (\Delta t)^2
\]

(4.7)

**Velocity analysis**

The velocities of the bodies can be calculated by deriving the equations in \( \Phi(\mathbf{q}(\mathbf{b}), \mathbf{b}, t) = 0 \) with respect to time:

\[
\dot{\mathbf{q}} = -\Phi_{\mathbf{q}}^{-1} \Phi_t
\]

(4.8)

Since only holonomic kinematic constraints equations are taken into account, the derivative \( \Phi_t \) is zero for all kinematic constraints. The derivative is only non-zero for the driver constraints:

\[
\Phi_t = \begin{bmatrix} 0 \\ \Phi_t^d \end{bmatrix}
\]

(4.9)

where 0 is a zero-vector. Generally, the vector \( \Phi_t \) is denoted as \( -\nu \) in multibody dynamics. For often used types of driver constraints a library of analytically differentiated vectors \( \Phi_t \) has been setup in general-purpose multibody software [3,7].

The inverse of the Jacobian matrix \( \Phi_{\mathbf{q}}^{-1} \) does not need to be recalculated if this inverse matrix has been stored after the last iteration step in the Newton-Raphson algorithm in equation (4.5). This re-use of the matrix is computationally efficient.

**Acceleration analysis**

Similarly to the velocity equations, the accelerations of the members of the system can be calculated by deriving the velocity equations with respect to time:

\[
\ddot{\mathbf{q}} = -\Phi_{\mathbf{q}}^{-1} \left[ (\Phi_{\mathbf{q}} \dot{\mathbf{q}})_{\dot{\mathbf{q}}} \dot{\mathbf{q}} + \Phi_{tt} + 2 \Phi_{\mathbf{q}} \ddot{\mathbf{q}} \right]
\]

(4.10)

Part \( 2 \Phi_{\mathbf{q}} \ddot{\mathbf{q}} \) is always zero for holonomic kinematic constraints. The part \( \Phi_{tt} \) is only non-zero for the driver constraint equations as the kinematic constraint equations are time-independent:

\[
\Phi_{tt} = \begin{bmatrix} 0 \\ \Phi_{tt}^d \end{bmatrix}
\]

(4.11)

For often used kinematic elements and driver types an analytical equation library has been setup for the relations \( - (\Phi_{\mathbf{q}} \dot{\mathbf{q}})_{\mathbf{q}} \dot{\mathbf{q}} - \Phi_{tt} - 2 \Phi_{\mathbf{q}} \ddot{\mathbf{q}} \), in multibody dynamics generally denoted as \( \gamma \) [3,7]. The acceleration equations can thus be written as:

\[
\ddot{\mathbf{q}} = \Phi_{\mathbf{q}}^{-1} \gamma
\]

(4.12)

Again the inverse Jacobian matrix \( \Phi_{\mathbf{q}}^{-1} \) can be re-used from the position analysis.
4.4 Inverse-dynamic analysis

The inverse-dynamic analysis calculates the forces needed to perform a prescribed motion of the system. As the acceleration of the bodies in the multibody system are needed to calculate the forces, first a kinematic analysis of the system needs to be performed in order to obtain the acceleration vector $\dot{q}$.

The joint forces and torques can be derived from the following set of mixed differential-algebraic equations of motion:

$$
\begin{bmatrix}
    M & \Phi_q^T \\
    \Phi_q & 0
\end{bmatrix}
\begin{bmatrix}
    \ddot{q} \\
    \lambda
\end{bmatrix}
= 
\begin{bmatrix}
    Q \\
    \gamma
\end{bmatrix}
$$

(4.13)

again under the strict condition that the determinant of the Jacobian $\Phi_q(q,t)$ is non-zero for all time points:

$$
|\Phi_q(q,t)| \neq 0
$$

(4.14)

$Q$ is the vector with external forces acting on the elements of the multibody systems, for example forces due to gravity. The Lagrange multiplier can be calculated using equation (4.13):

$$
\lambda = -\Phi_q^{-T} [M\ddot{q} - Q]
$$

(4.15)

These multipliers $\lambda$ uniquely define the constraint forces and torques that act on the system. The joint reaction forces $F_i$ and torque $T_i$ on body $i$ can be determined in terms of these Lagrange multipliers and based on the variational equation of motion as [3]:

$$
F_i^k = -C_i^T A_i^T \Phi_{r_i}^{kT} \lambda^k
$$

(4.16)

and

$$
T_i^k = \lambda^k \left( s_i^T B_i^T \Phi_{r_i}^{kT} - \Phi_{r_i}^{kT} \right)
$$

(4.17)

where $i$ is the body number and $k$ is the joint on which the forces are acting. $A$ is the rotation transformation matrix, $B$ and $C$ are the first and second derivative of the rotation transformation matrix $A$, respectively. The rotation transformation matrix $A_i$ describes the transformation between the global and local coordinate system of body $i$. $s_i$ is the distance between the joint $k$ on body $i$ and the center of mass of body $i$.

4.5 Design sensitivity analysis

The design sensitivity analysis of the positions, velocities, accelerations, and forces acting on the bodies in the multibody system can be calculated using the methods mentioned in the previous chapter:

- finite difference
- analytical direct differentiation
- semi-analytical direct differentiation
- analytical adjoint variable method
- semi-analytical adjoint variable method
4.5.1 Design sensitivity of positions

In the finite difference method the design sensitivity with respect to the design variables \( \mathbf{b} \) is calculated using small design variable perturbations:

\[
q_b = \sum_{i=1}^{n} \left( \frac{q(b + \Delta b e_i) - q(b)}{\Delta b} \cdot e_i \right)
\]  

(4.18)

where \( e_i \) is a column-vector with a one on the \( i^{th} \) position and zeros elsewhere. In the direct differentiation method the set of equations in equation (4.3) is differentiated with respect to the design variables:

\[
\Phi_q q_b + \Phi_b = 0
\]  

(4.19)

rearranging the equation results in

\[
q_b = -\Phi_q^{-1} \Phi_b
\]  

(4.20)

Interesting to see is the re-use of the inverse Jacobian matrix \( \Phi_q^{-1} \). The inverse Jacobian is calculated in the Newton-Raphson algorithm to estimate the position \( q \) in equation (4.5). If the inverse Jacobian is stored at that moment, it can be re-used in the calculation of the velocities and accelerations as stated before. The inverted matrix can also be used in the design sensitivity analysis of the positions in equation (4.20). In the following sections will be shown that the inverse Jacobian matrix is also re-used in the design sensitivity analysis of the velocities and accelerations.

The unknown element in equation (4.20) is \( \Phi_b \), the design sensitivity of the kinematic and driver constraint equations. It can be calculated analytically or semi-analytically as described in the chapter 3.

4.5.2 Design sensitivity of velocities

The design sensitivity analysis of the velocities is comparable to the analysis for the positions. Differentiating the velocity equation (4.8) with respect to the design variables \( \mathbf{b} \) using the direct differentiation method leads to:

\[
\Phi_q \dot{q}_b = -(\Phi_q \dot{q})_q q_b - (\Phi_q \dot{q})_b + \nu_q q_b + \nu_b
\]  

(4.21)

after replacing \( \nu \) with \(-\Phi_\nu\) and rearranging the equation:

\[
\dot{q}_b = -\Phi_q^{-1} \left[ \left( (\Phi_q \dot{q})_q + \Phi_\nu \right) q_b 

+ (\Phi_q \dot{q})_b + \Phi_b \right]
\]  

(4.22)

where

- \((\Phi_q \dot{q})_q\) is already calculated in the multibody analysis (acceleration equations)
- \(\Phi_\nu = 0\) for holonomic multibody systems
- \(\Phi_b\) is only non-zero if one or more design variables appear in driver constraints
- \(\Phi_b\) and \((\Phi_q \dot{q})_b\) can be calculated analytically or by means of finite difference
4.5.3 Design sensitivity of accelerations

The design sensitivity analysis for the accelerations is equivalent to the design sensitivity analysis of the positions and velocities:

\[
\Phi_\mathbf{a}_b = - (\Phi_\mathbf{a}_a)_b \mathbf{q}_b + \gamma_\mathbf{a}_b + \gamma_\mathbf{q}_b + \gamma_\mathbf{b}
\]  

(4.23)

or after rearranging the equation and replacing \( \gamma \) with \(- (\Phi_\mathbf{a}_a)_b \mathbf{q} - \Phi_\mathbf{tt} \)

\[
\mathbf{q}_b = - \Phi_\mathbf{a}^{-1} \left\{ \left( \Phi_\mathbf{a}_a \right)_b + \left( \Phi_\mathbf{a}_a \right)_b \left( \Phi_\mathbf{a}_a \right)_b + 2 \left( \Phi_\mathbf{a}_a \right)_b \right\} \mathbf{q}_b
\]  

(4.24)

where

- \( \left( \Phi_\mathbf{a}_a \right)_b \) can be derived from the relation \( \left( \Phi_\mathbf{a}_a \right)_b \) in equation (4.22)

- \( \left( \Phi_\mathbf{a}_a \right)_b \) must be calculated analytically or be taken from an analytical relations library.

- \( \left( \Phi_\mathbf{a}_a \right)_b \) must be calculated analytically or be taken from an analytical relations library

- \( \Phi_\mathbf{tt} \) and \( \Phi_\mathbf{tt} \) are zero-matrices for holonomic multibody systems

- \( \left( \Phi_\mathbf{a}_a \right)_b \) and \( \left( \Phi_\mathbf{a}_a \right)_b \) can be calculated analytically or by means of finite difference

- \( \Phi_\mathbf{tt} \) is only non-zero if the design variables appear in the driver constraints and can be calculated analytically or with the finite difference method

- \( \left( \Phi_\mathbf{a}_a \right)_b \) is a zero-matrix for holonomic multibody systems

The analytical relations library for the relations \( \left( \Phi_\mathbf{a}_a \right)_b \) and \( \left( \Phi_\mathbf{a}_a \right)_b \) will have to be setup if the design sensitivity analysis is included in general-purpose multibody software.

4.5.4 Design sensitivity of joint reaction forces

The design sensitivity of the Lagrange multipliers can be calculated by differentiation of equation (4.15) [8]:

\[
\Phi_\mathbf{a}^T \lambda_b = - \left[ (\Phi_\mathbf{a}^T \lambda)_b + (\Phi_\mathbf{a}^T \lambda)_q \mathbf{q}_b + M_b \mathbf{q} + M_b \mathbf{q}_b + Q_b - Q_q \mathbf{q}_b \right]
\]  

(4.25)

and after rearranging the equation:

\[
\lambda_b = - \Phi_\mathbf{a}^{-T} \left[ (\Phi_\mathbf{a}^T \lambda)_b + (\Phi_\mathbf{a}^T \lambda)_q \mathbf{q}_b + M_b \mathbf{q} + M_b \mathbf{q}_b + Q_b - Q_q \mathbf{q}_b \right]
\]  

(4.26)

If the design sensitivity of the Lagrange multipliers is known then the design sensitivity of the joint reaction forces and torques can be calculated:

\[
R^k_i = - \left[ (\Phi_\mathbf{q}^{kT} \lambda^k)_b + (\Phi_\mathbf{q}^{kT} \lambda^k)_q \mathbf{q}_b + (\Phi_\mathbf{q}^{kT} \lambda^k)_q \mathbf{q}_b + (\Phi_\mathbf{q}^{kT} \lambda^k)_q \mathbf{q}_b \right]
\]  

(4.27)

where \( k \) is the number of the joint connecting the bodies \( i \) and \( j \). \( R_b \) is the matrix with the design sensitivity of the joint reaction forces \( F^k_i \) and torques \( T^k_i \).
4.6 Conclusions of the design sensitivity analysis

The design sensitivity methods used in structural problems can be well implemented for kinematic multibody problems. Furthermore, combining the multibody and design sensitivity analysis is interesting in view of the computational costs. The computational costly inverse of the Jacobian matrix and several other results of multibody analyses are re-used in the design sensitivity analysis of the multibody mechanism.

With respect to the sequential local approximate optimization, the combination of multibody analysis and design sensitivity also has its advantages. Results from calculations for the multibody analysis can be stored and re-used for the design sensitivity analysis. The constraint elimination method can still be used in combination with the adjoint variable method. However, a very large reduction is not always possible. For example, if the design sensitivity of the velocities of bar 2 in figure 5.1 is needed then not only the design sensitivity of the positions of bar 2 two is needed. To calculate the design sensitivity of the velocities of bar 2 also the design sensitivities of the positions of bar 1 and 3 are needed. It is important to check which design sensitivities are needed.
Chapter 4: Design sensitivity analysis of kinematically driven systems

Figure 4.6: Flowchart of multibody and design sensitivity analysis
Chapter 5

Examples of Design Sensitivity Analysis

5.1 Examples design sensitivity analysis

The design sensitivity analysis which is described in chapter 4 has been implemented for two kinematically driven multibody systems; a four-bar mechanism and a slider-crank mechanism.

The multibody and design sensitivity analysis have been implemented in Matlab such that several intermediate results obtained during the multibody analysis can be stored and re-used in the design sensitivity analysis. In this way, the reduction of the computational efforts to calculate the design sensitivity can be estimated.

The implementation and the results will be presented in this chapter.

5.2 Design sensitivity analysis of a four-bar mechanism

5.2.1 Four-bar mechanism

The first example is a four-bar mechanism, see figure 5.1. In this four-bar mechanism 12 design variables have been defined, which are marked with $b_i$ in the figure and explained in the table on the next page.

The system has one degree of freedom. In this example angle $\phi_2$ of bar 1 has been chosen as degree of freedom. The kinematic analysis calculates the motion of the entire system as a function of $\phi_2$. Angle $\phi_2$ of bar 1 is defined as: $\phi_2(b, t) = \omega \cdot t + \phi_0 = b_0 \cdot t + b_3$. So bar 1 is driven with constant angular velocity.

Firstly, the multibody analysis of this four-bar mechanism will be performed, and then the corresponding design sensitivity will be calculated.
5.2.2 Multibody analysis

The multibody system is analyzed according to the flowchart in figure 4.6.

Position analysis

To begin with, the positions of the elements in the multibody system as a function of time are calculated using the Newton-Raphson method of equation (4.5) and equation (4.7). To solve this set of equations the analytically derived Jacobian matrix is needed.

Figure 5.2 depicts the motion of the joints (J) and the centers of the bodies (B) in the mechanism, and figure 5.3 shows the configuration of the mechanism in two different situations.
Chapter 5: Examples of Design Sensitivity Analysis

\( t = 0.0 \) [s] and \( t = 0.8 \) [s]).

![Motion of multibody system](image1.png)  
![Motion of multibody system](image2.png)

**Figure 5.2: Motion of mechanism - 1**  
**Figure 5.3: Motion of mechanism - 2**

**Velocity, acceleration, and inverse-dynamic analysis**

The next step is the calculation of the velocities, accelerations, and Lagrange multipliers. Equations (4.5) through (4.15) have been used for these analyses. Only if the results of these analyses are required for the interpretation of the design sensitivity analysis results, they will be provided in the next section.

### 5.2.3 Computational costs multibody analysis

The following table represents the computational costs for the multibody analysis over the entire time span. This analysis has been performed using a time discretization size \( \Delta t = 0.01 \) [s]. The maximum allowable errors in the position analysis are \( \epsilon_1 = 10^{-10} \) and \( \epsilon_2 = 10^{-10} \), see equation (4.6). The results are given such that they can be compared with the computational efforts to calculate the design sensitivities.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Flops (Matlab 4.2c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position analysis</td>
<td>564300</td>
</tr>
<tr>
<td>Velocity analysis</td>
<td>29088</td>
</tr>
<tr>
<td>Acceleration analysis</td>
<td>34340</td>
</tr>
<tr>
<td>Inverse-dynamic analysis</td>
<td>61812</td>
</tr>
</tbody>
</table>

The position analysis is the most expensive as the iterative Newton-Raphson method is used to calculate the solution of the set of equations. For the four-bar mechanism generally at least two or three iterative steps are necessary to solve this set of equations. That includes the, computationally expensive, calculation of the inverse of the Jacobian matrix. Furthermore, the computational costs of the position analysis depend on the conditions in equation (4.6).
In practice, the inverse of the Jacobian matrix should not be calculated explicitly. Instead a factorization like the LU-decomposition is far more efficient. However, for this four-bar mechanism the set of equations is quite small. In this case the *inv*-function in Matlab is less expensive than the LU-decomposition.

The velocity and acceleration analyses are less expensive than the position analysis since an analytical library is used to estimate the velocities and accelerations. The same applies for the inverse-dynamic analysis. The inverse of the Jacobian matrix, calculated in the position analysis, is re-used in this analysis.

### 5.2.4 Design sensitivity analysis

#### Design sensitivity analysis of positions

The design sensitivity of the positions has been calculated in Matlab using the different methods for the sensitivity analysis:

- finite difference method
- analytical direct differentiation
- semi-analytical direct differentiation
- analytical adjoint variable method
- semi-analytical adjoint variable method

As examples of the sensitivity analyses are given (arbitrarily chosen):

- the design sensitivity of the orientation of bar 2 with respect to the length of bars 2 and 3, denoted as \( \frac{\partial q_9}{\partial b_2} \) and \( \frac{\partial q_9}{\partial b_3} \)
- the design sensitivity of the position of bar 3 with respect to the length of bars 1 and 2, denoted as \( \frac{\partial q_{10}}{\partial b_1} \), \( \frac{\partial q_{11}}{\partial b_1} \), \( \frac{\partial q_{10}}{\partial b_2} \), and \( \frac{\partial q_{11}}{\partial b_2} \)

where \( q_9 \) is the orientation of bar 2, \( q_{10} \) is the horizontal position of bar 3, and \( q_{11} \) is the vertical position of the bar. The figures 5.4 and 5.5 show the orientation of bar 2 and the position of bar 3 as a function of time respectively. Figures 5.6 and 5.7 represent the results of the design sensitivity analysis.

Figures 5.6 and 5.7 illustrate that the mechanism is especially sensitive to changes around \( t = 0.8 \) [s], indicated in the figures by the large change around that time point. The explanation for this high sensitivity is that the mechanism is near a lock-up configuration at this time-point; the mechanism is very sensitive to small changes in the length of the bars around this point.

The computational difference between the different sensitivity methods are quite large, the next table shows these differences. The high computational costs for the finite difference method can be explained by the fact that the Newton-Raphson method in equation (4.5) has to be repeated in the position analysis each time a design variable is changed.

<table>
<thead>
<tr>
<th>Table: Computational costs sensitivity analysis positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>Finite difference</td>
</tr>
<tr>
<td>Analytical direct differentiation</td>
</tr>
<tr>
<td>Semi-analytical direct differentiation</td>
</tr>
</tbody>
</table>

(analytical direct differentiation method is used as reference)
The computational time of the design sensitivity analysis using the adjoint variable method depends on the number of positions for which the design sensitivity is to be calculated. The costs of the design sensitivity analysis of a position are about $1/12^{th}$ of the costs using the direct differentiation method in the above table. The design sensitivities of the 12 Cartesian coordinates are calculated at once using the direct differentiation method. The computational time to calculate the design sensitivity of the 12 Cartesian coordinates using the adjoint variable method are slightly higher compared to the direct differentiation method as the adjoint vector has to be calculated.

Again it is important to remark that the computational costs of the finite difference method depend on the accuracy demanded in the Newton-Raphson method to calculate the positions.

Even more important than the differences in computational costs is the difference in accuracy between the methods. In figure 5.8 the accuracy of the methods is compared by changing the finite difference step size. The maximum error over the time span is used as measure of accuracy and the analytical direct differentiation method has been used as reference.
Chapter 5: Examples of Design Sensitivity Analysis

One would expect the semi-analytical direct differentiation method to be inaccurate for a large step size $\Delta b$ as well. However, in this particular problem the design variables all appear linear in the set of equations, see equation (??). Therefore, a large step size $\Delta b$ does not result in approximation errors, as it does in the finite difference method.

For this problem the difference in accuracy for the design sensitivity analysis of the positions is very small, i.e. negligible for variable perturbations less than $10^{-8}$. Figure 5.9 shows the difference between the finite difference method and the analytical direct differentiation method in a arbitrarily chosen design sensitivity analysis using a poor step size $\Delta b$.

Figure 5.8 exemplifies also that the round-off errors are inverse proportional to the variable perturbation. The approximation errors in the finite difference method are proportional to the finite difference step size.

![Figure 5.8: Accuracy design sensitivity positions](image)

![Figure 5.9: Difference accuracies](image)

**Design sensitivity analysis of velocities**

The design sensitivity analysis for the velocities has been implemented the same way as the design sensitivity analysis for the positions. Figures 5.10 and 5.11 show the angular velocity of bar 2 and the velocities (horizontal and vertical) of bar 3. Figure 5.12 and 5.13 depict the design sensitivity of the velocities using the same design variables as in the design sensitivity analysis of the positions in figures 5.6 and 5.7. It can be seen that the lines in figures 5.10 and 5.11 represent the differentiation of the lines in figures 5.6 and 5.7 with respect to time. Again it can be recognized that the system is sensitive to changes in the length of the bars around $t = 0.8 \text{[s]}$.

The following table shows the difference in computational efforts to calculate the design sensitivities using the different methods.

<table>
<thead>
<tr>
<th>Table: Computational costs sensitivity analysis velocities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>------------------------------</td>
</tr>
<tr>
<td>Finite difference</td>
</tr>
<tr>
<td>Analytical direct differentiation</td>
</tr>
<tr>
<td>Semi-analytical direct differentiation</td>
</tr>
</tbody>
</table>

(Analytical direct differentiation method is used as reference)
Chapter 5: Examples of Design Sensitivity Analysis

The design sensitivity analysis of the velocities is more extensive and expensive than the same analysis for the positions. This can be explained for the (semi-)analytical methods by the fact that the equation 4.20 demands for more calculations than equation 4.22. The finite difference demands for large computational times because both the position vector and the velocity equation 4.8 have to be recalculated for each change in a design variable.

**Design sensitivity analysis of accelerations**

The design sensitivity analysis for the accelerations has been implemented and tested in the same way as the design sensitivity analysis for the positions and the velocities. The results for the same analysis will be given.

The error between the semi-analytical method and analytical method stays the same because only round-off errors are present in the semi-analytical method. The table shows the difference in computational costs between the different methods for the sensitivity analysis.
Chapter 5: Examples of Design Sensitivity Analysis

The design sensitivity analysis for the accelerations is more expensive than the same analysis for the positions and velocities. Here the same explication applies as for the costs of the velocity equations; the equations to be solved are larger.

### Design sensitivity analysis of Lagrange multipliers

The same sensitivity analysis as in the preceding sections have been performed on the Lagrange multipliers.

For the design sensitivity analysis of the Lagrange multiplier applies the same as for the other analysis, the mechanism is very sensitive to changes in several design variables around 38.
t = 0.8 [s]. As the accelerations in the mechanism are quite large around that time point, the forces on the mechanism will increase also. The forces and torques acting on the mechanism are directly related to the Lagrange multipliers as explained in chapter 4.

<table>
<thead>
<tr>
<th>Table: Computational costs sensitivity analysis Lagrange multipliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>Finite difference</td>
</tr>
<tr>
<td>Analytical direct differentiation</td>
</tr>
<tr>
<td>Semi-analytical direct differentiation</td>
</tr>
</tbody>
</table>

(Analytical direct differentiation method is used as reference)

Inaccuracies in the calculation of the design sensitivity of the positions, velocities, and the accelerations are cumulated in the design sensitivity analysis of the Lagrange multipliers. This can be noticed in figure 5.26. In the same figure can be seen that the difference in accuracy
Chapter 5: Examples of Design Sensitivity Analysis

5.2.5 Multiple configurations

Tests on the four-bar mechanism have shown that the estimation of the initial configuration of the multibody system is quite important. For the four-bar mechanism two configurations are possible. If bar 1 in figure 5.1 cannot make a complete revolution then even more different initial configurations are possible. The initial estimation of the configuration should be (relatively) close to the desired initial configuration.

5.2.6 System that can not be assembled

If the initial setup is such that the multibody system cannot be assembled, then the set of equations (4.3) cannot be solved. The inverse Jacobian which has been calculated in the last iteration step of the Newton-Raphson algorithm cannot be used to calculate the design
sensitivity. The inverse Jacobian matrix can only be used if the Newton-Raphson algorithm has come to a solution for the position vector.

Several tests have been performed check systems which can not be assembled. In the example figure 5.1 bar 2 has been made too short such that the system can not be assembled anymore. In a subsequent simulations bar 2 has been made longer but in such a way that the system can still not be assembled. These test show that nor the determinant of the Jacobian matrix, nor the obtained results for vector $\mathbf{q}$ after the last iteration step, give any indication of how much design variables need to be changed in order to be able to assemble the system.

5.2.7 Lock-up configuration

Lock-up is the configuration after which the motion can no longer continue; the multibody system can no longer be assembled after that time point. In the multibody mechanism in figure 5.1 a lock-up situation will be encountered if the length of the crank $b_1$ is larger than 1.6 [m]. The lock-up configuration will be found under these conditions around $t = 0.81$ [s].

In chapter 4 it was indicated that a lock-up configuration can be recognized by the velocities $\mathbf{q}$ and/or the accelerations $\dot{\mathbf{q}}$ approaching infinity. This has been tested on the four-bar mechanism. Figure 5.28 shows the absolute velocities of the bars for two different lengths of the crank ($b_1$). In case of $b_1 = 1.0$ [m] no lock-up configuration will be encountered. In case of a crank that is too large, $b_1 = 1.7$ [m], a lock-up configuration will appear around $t = 0.81$ [s]. In figure 5.29 the absolute acceleration of the bars are represented. In both figures 5.28 and 5.29 it can be seen that the velocities and the accelerations of two bars approach infinity around the lock-up configuration. The increase of the velocity of several bars is also shown in figure 5.30. The dots representing the position of the centers of the bodies at discrete time intervals become more separated just before the lock-up configuration.

The lock-up configuration has also been tested by changing other design variable values. Each time a lock-up situation was found, the velocities and acceleration of several bodies were approaching infinity.

A lock-up configuration can be foreseen in the design sensitivity analysis of the velocities and accelerations. Near a lock-up configuration the design sensitivity of design variables leading to a lock-up configuration will show a very large increase in the velocities and accelerations of several bodies in the mechanism.
Chapter 5: Examples of Design Sensitivity Analysis

In the optimization of multibody systems a lock-up configuration could be detected and avoided by setting constraints on the maximum velocity and acceleration of the bodies in the mechanism.

5.2.8 Bifurcation configuration

The bifurcation configuration has not been tested numerically. In case of a bifurcation point, the Jacobian matrix is singular ($|\Phi_q| = 0$); the inverse of the Jacobian matrix can not be calculated.

Two multibody analysis have been performed with the possible configurations after the bifurcation point chosen as initial conditions for the multibody analysis. These simulations have shown that the determinant of the Jacobian matrix of these configurations only differ in sign, their absolute values are equal.

The bifurcation point can not be analyzed numerically as the numerical multibody analysis will stop before the bifurcation point due to numerical instabilities.
5.3 Design sensitivity analysis of a slider-crank mechanism

A slider-crank mechanism as shown in figure 1.2 has been tested for the design sensitivity of the positions. The reason to test slider-crank mechanism is that the set of equations describing the configuration of the slider-crank mechanism has a few equations in which the design variables are nonlinear in the equations. The nonlinearity is due to the kinematic constraint equations describing the translational joint [3]. The expectation is that the semi-analytical direct differentiation method will now show a difference with the analytical direct differentiation method for large finite difference steps. Figure 5.31 shows the orientation of the bar connecting the crank and slider. Figure 5.32 depicts the design sensitivity of the orientation of this bar with respect to the length of the crank and the design sensitivity with respect to the length of the connecting bar.

![Figure 5.31: Orientation connecting bar](image1)

![Figure 5.32: Sensitivity orientation bar](image2)

![Figure 5.33: Accuracy design sensitivity positions](image3)

The results in figure 5.33 illustrate that the semi-analytical direct differentiation method is now less accurate than the analytical method. This is due to the nonlinearities in the set of multibody equations.
5.4 Conclusions

The simulations show that all three design sensitivity methods can well be implemented in the optimization of kinematic multibody problems. However, the finite difference method is very costly in terms of computational times and is less accurate than the other methods. Furthermore, the accuracy depends on the finite difference step size.

The analytical direct differentiation method, and the analytical adjoint variable method, are the less expensive and most accurate methods to calculate the design sensitivities. The differentiation of the analytical relations of the set of equations with respect to design variables and multibody responses is however quite demanding.

The semi-analytical method demands more computational time than the analytical methods. The difference in computational costs is however negligible if compared to the finite difference method. The advantage of this method is that there is no need to differentiate all relations analytically. They are estimated by means of finite difference. The loss of accuracy in the simulations is only marginal.

The semi-analytical methods seems to be the most interesting method as the increase in computational time with respect to the analytical method is not very large, the loss in accuracy seems to be acceptable, and there is no need for cumbersome differentiation of equations.
Conclusions

The sequential local approximate optimization concept, the constraint elimination method, and the (semi)-analytical design sensitivity methods can well be combined in the optimization of kinematically driven multibody systems. Combining these methods leads to a reduction of the number of design sensitivity evaluations. This reduction is advantageous for the overall computational efforts needed to solve the multibody optimization problem.

The setup of the local approximation model demands for the design sensitivities of the multibody responses. Three methods for the design sensitivity analysis, which are already implemented in the optimization of structures, can be used for the design sensitivity analysis of kinematically driven multibody systems; the finite difference method, direct differentiation method, and adjoint variable method.

The finite difference method is most easy to implemented because there is no need for access to or even any knowledge of the multibody analysis code. This ease of implementation has to be paid for by large computational costs and loss of accuracy with respect to other methods.

The direct differentiation method and adjoint variable method, which are combined numerical-analytical methods, are more difficult to implement. However, these methods are more accurate and demand relatively little computational time. The difference between the adjoint variable method and the direct differentiation method in kinematically driven multibody systems is that the design sensitivities are calculated per constraint or per design variable, respectively. If the design sensitivities need not to be calculated for all multibody responses, then the use of the adjoint variable method can be interesting. The sequential local approximate optimization concept and constraint elimination method reduce the number of design sensitivities which need to be evaluated. Therefore, combining these methods with the adjoint variable method for the design sensitivity analysis further reduces the computational efforts.

The difference in computational costs and accuracy is almost negligible between the analytical and the semi-analytical methods, if the step size for the finite difference method within the semi-analytical method is well chosen. Analytical methods have some advantage in computational accuracy and efficiency but need a demanding and cumbersome differentiation of the set of multibody equation with respect to design variables and multibody responses.

Further reduction of the computational costs of the optimization process can be obtained by integrating the multibody and design sensitivity analysis. Several intermediate results obtained in the multibody analysis can be stored and later on be re-used in the design sensitivity analysis. Especially the re-use of the computationally expensive inverse of the Jacobian matrix is computationally efficient.
Recommendations

I would like to give the following recommendations concerning the design sensitivity analysis in the optimization of multibody dynamics:

- **optimization of multibody problems**
  The design sensitivity analysis of the four-bar and slider-crank mechanisms can be implemented in an optimization process. It is interesting to study the influence of the different design sensitivity methods on the number of cycles needed in the sequential local approximate optimization process. The finite difference step size could be varied to study its influence on the finite difference and semi-analytical direct differentiation method in the optimization process. Furthermore, multibody problems such as a lock-up configuration could be tested for.

- **analysis of dynamically driven multibody systems**
  For kinematically driven multibody systems the multibody analysis and design sensitivity analysis can be performed quasi-static, i.e. each time point can be studied individually. Information of the motion of the system in preceding time point is not needed ('history of motion'), except for a global estimation of the system’s positions. For dynamically driven multibody systems this is not the case. These systems have to be analyzed from the first time point on, using initial conditions for the positions and velocities of the bodies. This means that the combination of sequential local approximate optimization, constraint elimination, and adjoint variable method can be less effective compared to the case of kinematically driven multibody systems. The implementation of these methods in dynamically driven multibody systems needs to be studied.

- **analysis of direct differentiation and adjoint variable method in dynamical problems**
  There is only a slight difference between the direct differentiation and the adjoint variable method in the design sensitivity analysis of kinematically driven multibody systems, which is merely a question of computational efforts.

  In dynamically driven multibody systems the implementation of the two methods is quite different. The choice of the design sensitivity method also has an influence on the implementation of the constraint elimination method in dynamical systems. Therefore research regarding the implementation of the two design sensitivity analysis methods in this type of multibody systems is interesting.
Bibliography


47


