Development of a Modelica Library
for Simulation of Diffractive Optomechatronic Systems

Thomas Kaden Klaus Janschek
Institute of Automation, Faculty of Electrical Engineering
Technische Universität Dresden, 10162 Dresden
Thomas.Kaden@tu-dresden.de Klaus.Janschek@tu-dresden.de

Abstract

The proper operation and performance of optomechatronic systems is fundamentally affected by changes of the relative geometry caused by thermal influences, mechanical displacements and vibrations. Such extrinsic and intrinsic disturbances can be compensated by active control of optical elements like lenses, diffraction gratings or laser sources. In the context with system design and performance analysis tasks it is big challenge to model and simulate the coupled optomechatronic behavior including closed-loop control and disturbances properly on a representative level.

A promising approach is the integration of diffractive optic models in the well established physical object oriented modeling environment Modelica®, which offers already a broad support of multi-domain libraries, e.g. electrical, mechanical and thermal.

Therefore the basic modeling requirements for diffractive optical elements are outlined followed by a discussion of possible problems and solutions for a computationally efficient implementation of a two-dimensional spatial *optical library* for Modelica-based simulation environments.

Keywords: Modelica; Diffractive Optics; Optical library

1 Introduction

The application of optomechatronic systems is increasing constantly. Examples are telescopes with adaptive optics [4], motion compensated cameras [7], [8], diffraction based sensors for on-line textile inspection [1], optical Fourier processors and correlators [9] or interferometer arrangements.

Diffractive optical components are used in optomechatronic systems mainly for the purpose of fast on-line signal processing.

However such systems are affected by environmental influences. Mechanical displacements, vibrations and thermal distortions can significantly affect the proper function and accuracy of the optical subsystem due to changes of the optical path length. Therefore in the context with system design and performance analysis tasks the coupled optomechatronic behavior including closed-loop control and disturbances must be modeled and simulated.

In this paper the integration of diffractive optics models into the physically object oriented modeling environment Modelica is discussed. The requirements for physical object oriented optical models with diffractive optical functionality and interfaces are outlined. In particular the feasibility of two-dimensional spatial interfaces in Modelica is further evaluated and a tool-independent concept for an *optical library* is proposed. The use of Modelicas external object / function interface is motivated through the insufficient support of large matrices in existing Modelica tools.

2 Simulation of optomechatronic systems

Optomechatronic systems are usually composed of an arrangement of electronic, mechanic and optical elements [6] as shown in Fig. 1. Especially the optical functionality depends on a fixed arrangement of optical elements providing exact optical path geometry.

However the operation of mechanical and electrical components is corrupted by geometrical or thermal intrinsic disturbances of the arrangement of optical components. Additional errors are inserted by extrinsic disturbances like shock, vibration or environmental temperature changes.
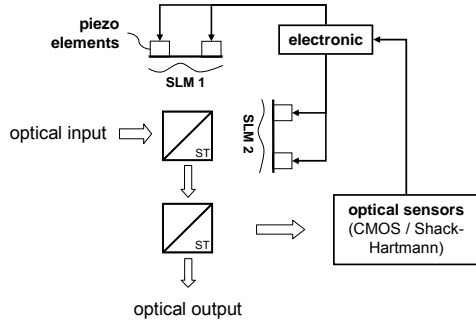
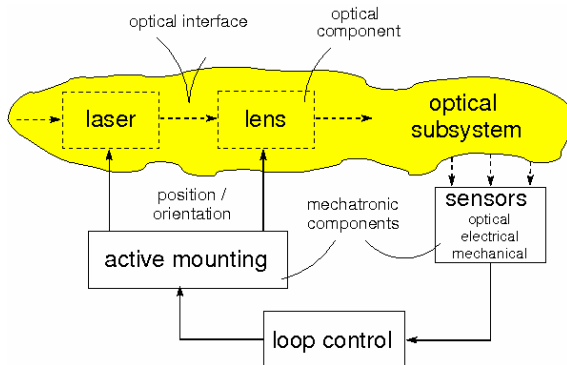**Fig. 1 Optomechatronic System**

The state of the art computer based physical object based modeling and simulation environments focus on two main areas:

- Simulation tools focused on optics like Zemax [17] including detailed geometrical and diffractive optics modeling, but without or only with insufficient mechatronic functionality,
- Multi-domain object oriented tools, e.g. Modelica-based, incorporating a broad mechatronic functionality, but up-to-date without optics functionality, in particular without any diffractive optics [16].

In recent years the object oriented, equation based modeling language MODELICA established mainly focusing at the modeling and simulation of electrical, mechanical, chemical and thermal systems.

The usual approach is to decompose a physical system down to single model components. The entities of components form a package and all packages belonging to one physical domain create a library.

Modelica currently supports electrical, mechanical, multibody mechanic or thermal libraries. Up to now there is no library in Modelica for simulating diffractive optical elements as needed for the simulation of optomechatronic systems, (Fig. 2).

Fig. 2 Optomechatronic system model

The basic problem is the missing support of Modelica for spatially distributed variables and equations [2].

Nevertheless the following analysis of one of the most important diffraction equation shows that basically matrix computation will be needed to implement a diffractive optical functionality.

3 Scalar wave optic

## 3.1 Rayleigh-Sommerfeld diffraction integral

Many optical phenomena like diffraction or interference of monochromatic coherent light waves can be described in a beneficial manner by scalar wave theory, which leads to a simplified formulation for diffraction phenomena. According to the HUYGENS-FRESNEL principle the complex amplitude distribution of light at a single point behind an aperture can be described by a weighted sum of spherical waves originating from every point within the aperture [3].

This principle can be stated in mathematical form as the well-known RAYLEIGH-SOMMERFELD diffraction integral (1st solution) given by [3], [11]:

$$A(\vec{r}_1) = \iint\limits_{\Sigma} E(\vec{r}_0) \frac{e^{(j \cdot k \cdot r_{01})}}{r_{01}} \cdot \cos(\vec{n}, \vec{r}_{01}) \cdot ds \ . \qquad (1)$$

Note that the diffracting aperture is assumed to be planar. For a statically geometric configuration, the integral (1) is independent of time and it can be characterized as algebraic relation between input and output.

3.2 Numerical implementation

As there is usually no analytical solution for (1) the spatial integration must be solved using numerical two-dimensional approximations on digital computers. The two-dimensional spatial discretization leads to a [N*M] matrix representation for a given aperture plane, (Fig. 3).
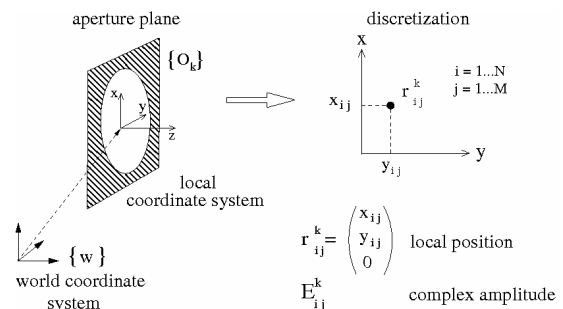

Fig. 3 Discretization of aperture plane

Integration of (1) now has to be replaced by N*M complex multiplications and summations for every output point. Therefore the calculation of an output plane of equal size requires $(N*M)^2$ complex operations. For higher optical resolutions (e.g. N=M=1024 pixel) the point wise solution of (1) is a very time consuming task.

3.3 Angular spectrum Method

An optical system can be generally considered as a two-dimensional spatial linear system. Equation (1) can then be solved using Fourier methods, in particular by application of the convolution theorem, assuming spatial invariance (parallel planes). Thus the computational effort for solving of integral equation (1) can be significantly reduced by application of standard fast Fourier transform (FFT) algorithms, complex matrix multiplication with a propagation kernel $H(f_x, f_y)$ and inverse fast Fourier transform (iFFT), (Fig. 4).
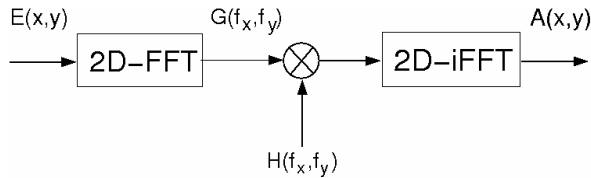
Fig. 4 Principle of angular spectrum method

The propagation kernel is given by [3]:

$$H(f_x, f_y) = e^{j \cdot k \cdot z \cdot \sqrt{1 - \lambda^2 \cdot (f_x^2 + f_y^2)}} \quad . \qquad (2)$$

This method is orders of magnitude faster than the point-wise calculation of (1).

Assuming the more general case of non parallel planes, this Fourier based method can be used as well, but frequency mapping must be implemented to compensate for the plane rotation [12], [13]. However this step involves *interpolation* due to a restriction inherent to the FFT, namely the fixed sampling of the frequency domain.

4 Concept for an optical library in Modelica

Object oriented modeling of a physical system or component in general needs to include two basic aspects [2]:

- The *interface* of the optical component to pass and access data.

- The *optical component* and its internal encapsulated functionality.

The main properties of optical elements are discussed from the point of view of object oriented modeling as well as there integration into an *optical library* for Modelica.

4.1 Interface

Object oriented modeling requires the encapsulating of the internal function. Communication with other model elements is only allowed through well defined interfaces which are called *connectors* in Modelica [2].

According to the requirements of scalar diffractive optics a connector has to represent a two-dimensional *plane in 3D-space*. Every plane is associated with a complex amplitude light distribution. The spatial discretization of such a cut plane leads to *matrix input/output connectors*; a possible interface structure is shown in Fig. 5.

The propagation of light is not bounded to material and this decoupling suggests the use of *causal* input/output connectors instead of *acausal* connectors as explained in the following.
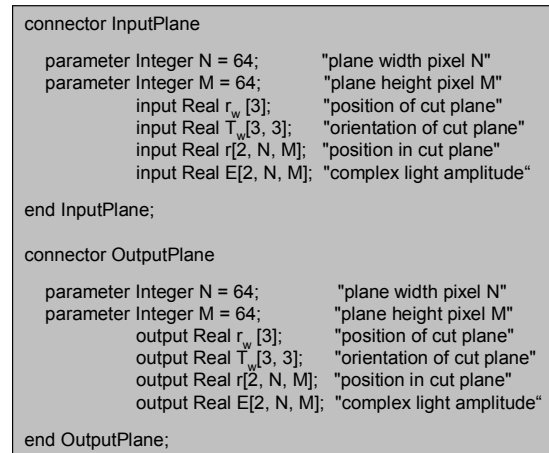
Fig. 5 Example of an optical input/output connector in Modelica

Modelicas acausal connector principle represents the physical behaviour of components at the cut points. As a consequence the direction of a connection is not specified. This approach follows the interaction between components in the physical world [2].

Generally light does not need material to propagate through space. Light travels through media as well as through vacuum. The lack of an energy carrier for that spatial domain makes it difficult for scalar wave optics to follow the concept of acaucal connectors.

Considering the propagation of light in the direction from one component to the next component an optical connection can suitably be described by Modelicas signal based or causal connectors as described above.

This component model however does not have the ability to describe a reactive behaviour of light waves to the previous optomechatronic component. This limitation is acceptable for the optical functionality because light waves at one point in space can interpenetrate without disturbing each other.

The influence of light to mechanical components in terms of forces, position and orientation can be neglected for macroscopic mechatronic systems and small optical energy. Under this assumption a light wave can be considered being non-reactive to the mechanical components.

In principle the energy of light can be transformed in heat when interacting with mechanical and electrical components. This conversion however takes place within the optomechatonic components and can be considered by an additional thermal model if needed.
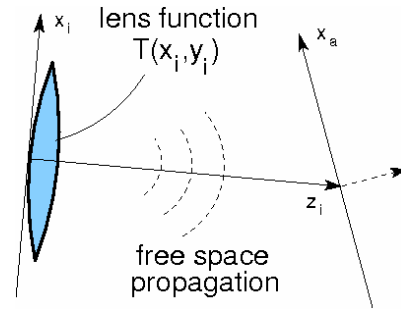
4.2 Components

Every element with a significant optical function is considered being a component. The interesting physical properties include finite mass and spring stiffness and a mechanical connection to other (optical) elements forming a multi-body system (MBS), (Fig. 6).
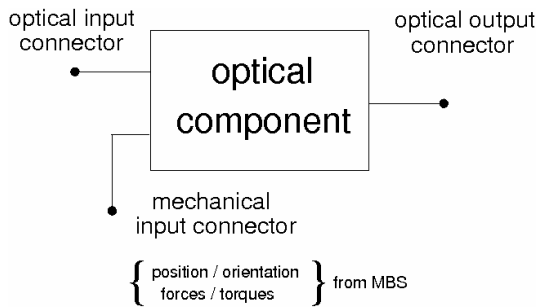
Fig. 6 Generic optical component

For a component model the following approach is chosen. The optical input plane is perpendicularly located directly in front of the optical element body. The output plane is determined by the input plane of the next optical element.

The relative geometry between the actual component and the subsequent component determines the geometry of the output plane, (Fig. 7).

Fig. 7 Optical component functionality

The optical function of the component is defined between input and output plane. It can be modeled as a complex two dimensional optical function $T(x_i, y_i)$ defined at the location of the input plane followed by free space propagation [3], [5].

Matrix calculation is formally supported be Modelica and for many diffractive optical setups the optical functionality can be specified by a complex matrix in a first step.

4.3 Additional properties for an optical library

The implementation of an easy-to-use optical library and the equation based modeling paradigm of Modelica require the description of two dimensional spatial variables in any kind. There is ongoing development for the support of partial differential equations (PDE) and its associated domain definitions which however is not yet operational so far [14], [15]. Hence an equation based description of the optical function $T(x_i, y_i)$ is not possible.

It is therefore necessary to use a physical description of the two-dimensional optical cut plane (optical domain information) and convert it into an internal two-dimensional complex matrix representation which can be handled by the simulation tool, (Fig. 8). The same holds for the description of optical equations which are defined on that domain.

At last the visualisation of two-dimensional data for examination should be available by the simulation tool. Access and visualisation should be granted not only to Modelica connector matrices but also to static internal matrix variables if they exist.
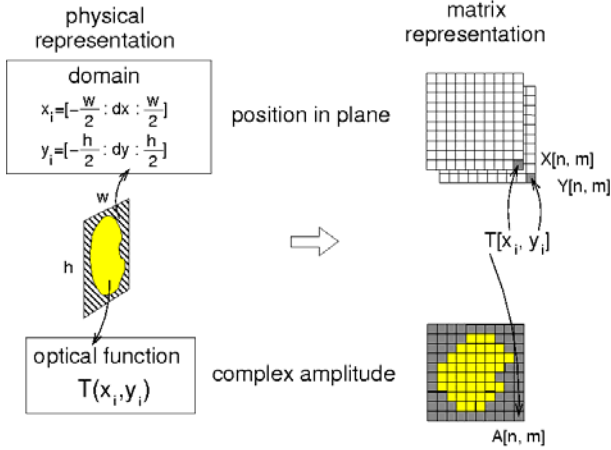
Fig. 8 Domain to matrix conversion

5 Implementation

5.1 Optical Benchmark system: using matrix functionality provided by Modelica

Before designing and implementing complex Modelica library elements, the feasibility and adequate functioning of possible matrix interfaces and internal algorithms must be evaluated. Therefore we consider first the following model of an elementary optical system consisting of a perfectly planar monochromatic light source, a pin-hole aperture and a Fourier lens (Fig. 9). The overall internal component calculations (discrete approximation of angular spectrum method for RAYLEIGH-SOMMERFELD diffraction integral) are simply two complex matrix multiplications (pinhole and lens function) and free space light propagation after the lens.
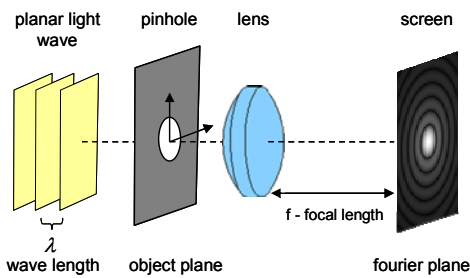

Fig. 9 Optical benchmark system

According to the object oriented modeling approach, the body elements from the existing Modelica multibody library were extended by optical input and output connectors.

The experiment results for the model translation time (Modelica to C-code) of the benchmark experiments with varying connector matrix size N is shown in Table 1. Increasing matrix size N leads to the observation that the computation time for analyzing and

flattening the model increases significantly. For technically interesting optical resolutions, e.g. N=1024, the computation time is out of scope of practical simulation experiments. Using different Modelica based tools like SimulationX$^©$, OpenModelica or MathModelica$^©$ show similar behavior.

Table 1. Model translation time (Modelica to C-code) for the benchmark model with SimulationX, matrix size [N*N] [+)]

N	8	16	32	48	64	128
t / sec	30	40	100	310	1060	-

[+)] Opteron 175, DualCore, 2GB Ram

This behavior is caused by the standard analysis and translation process from MODELICA source code to executable C-code. During the conversion of the model into executable code (flattening process) the tools normally decide which variables become a state variable [2]. This process can take a long analysis time for large matrices.

5.2 Matrix implementation using the External Object / Function interface

A possible alternative *tool independent* solution for the problem described above can be considered using the external function / object interface described in the Modelica language specification [14]. The solution benefits from a fast calculation time, handling of large matrices and can be implemented in a modular way, (Fig. 10).
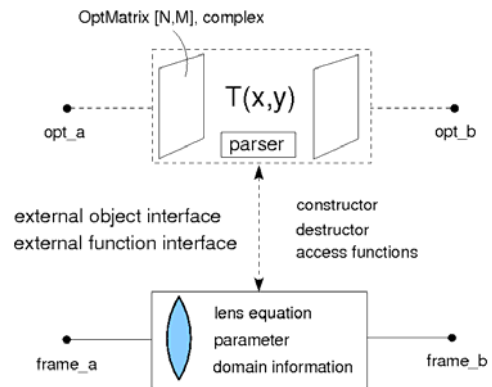

Fig. 10 Basic concept for optical library elements

While handling the matrix calculation externally with C-code the formal description of two-dimensional domains (planes in space), equations and parameter remain in Modelica.

This approach leads to a *composite optical connector* composed of a matrix with complex amplitude values representing the light wave (OptMatrix) and a coordinate frame (input: frame_a, output: frame_b) associated with the respective cut plane, (Fig. 11).

The model OptMatrix is extended from Modelicas external object class. In connection with the coordinate frame which represents the position and orientation of the plane in space it forms the desired optical connector.
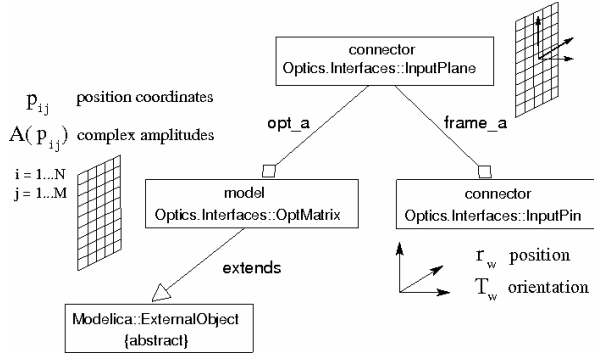
**Fig. 11 Optical connector with external object/function interface for the input cut plane**

The external object implements the following structure in C-code while constructor and destructor functions handle the memory accordingly:

```
typedef struct{
    unsigned int N;      // number of matrix rows
    unsigned int M;      // number of matrix cols
    double* AmpReal;  // complex amplitude, real part
    double* AmpImag;  // complex amplitude, imag. part
    double* PosX ;      // position in plane, x axis
    double* PosY;       // position in plane, y axis
}OptMatrix;
```

The created objects are then handled through the respective Modelica tool. Access can be granted through external functions as described in the Modelica Language Specification [14]. According to the Modelica Language Specification all access functions can be written in C-code, compiled and encapsulated in a system library ('.a' for unix / linux, '.dll' for windows).

Every external function needs a Modelica wrapper function operating with external objects and using the external system library. Those functions are placed in a separate package within the optical Modelica library, (Fig. 12).

Optical components are modeled basically as rigid body elements (maybe within a multibody system) which are extended with optical input and/or output connectors. The *generic optical components* can be further specified implementing specific optical functionality.
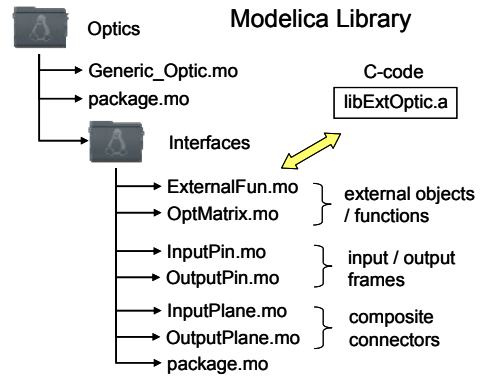

Fig. 12 Modelica optical library structure

Considering again the benchmark model (Fig 9) and its optical functionality it is known that under certain conditions the lens performs the (optical) Fourier-transform of the pinhole circle function [3, 5]. The required conditions are that that pinhole and screen are positioned perpendicular at the focal points of the lens and that the lens is thin and convex.

The Fourier transform of the circle function can be described analytically and it is called Airy function [3]. It will serve as a reference for testing the optical end-to-end performance of the benchmark system and the implementation of the internal diffraction equation.

The fist step however will be the proof that handling matrix calculation with Modelica is possible in a manageable way. The test system consists of four optical components, i.e. PlaneWave, PinHole, ThinLens and Screen) leading to the following system structure (Fig 13).
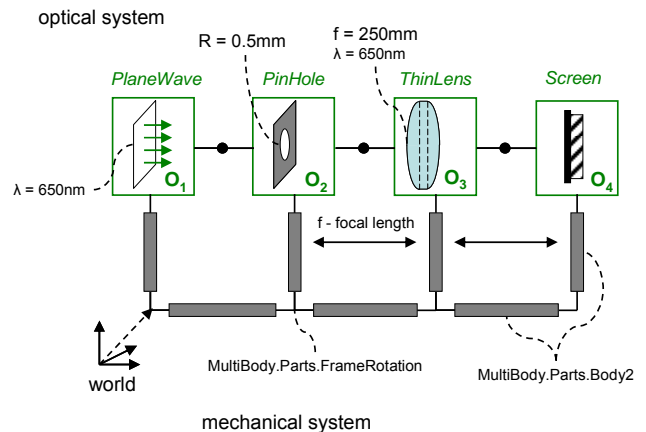
Fig. 13 optical test system in Modelica

For the implementation of the optical functionality we assume perpendicular planes first. This means that the system is static and perfectly aligned with the optical axis as shown in (Fig 13). It allows the use of the angular spectrum method for calculation of the Sommerfeld light diffraction after the pinhole and

the lens according to equation (1). Therefore the following calculation steps are executed internally for those components:

- Matrix multiplication of the respective optical function (pinhole, lens phase function)
- Fast Fourier transformation (FFT) of the input data
- Centering of FFT data (fftshift function)
- Calculation of spatial propagation filter $H(f_x,f_y)$ according to equation (2) and matrix multiplication with FFT data
- Inverse Fourier transformation (iFFT) of the result

For the implementation of the FFT / iFFT the fftw3 library is included in the external function library [18].

The radius of the pinhole (R=0.5mm) as well as typical lens parameters (focal length f=250mm) and optical parameters (λ=650nm) are parameterized.

Up to now the simulation experiments show that the overall translation time for the whole model with several complex matrix manipulations and a pure mechanical multi body mass system of an optical bench is acceptable also for large spatial matrix dimensions (Table 2)

Table 2. Model translation time for Benchmark Model with external function interface, matrix size [N*N][+)]

N	512	1024	2048
t / sec	45	50	55

[+)] Opteron 175, DualCore, 2GB Ram

A detailed evaluation of simulation performances under dynamic conditions is currently in progress.

5.3 External data access and visualization

The matrix values of every component e.g. input / output planes should be accessible. An easy solution for large matrix data is to store the matrix values as a binary file. Currently the visualization is done by Matlab which reads the binary file over a '.mex' interface and uses image functions for visualization.

6 Conclusion

The physical object oriented modeling and simulation of optomechatronic systems currently lack of an appropriate easy-to-use modeling tool. Modelica can handle electrical, mechanical and thermal physical

domains and provides well implemented libraries but does not cover diffractive optics.

Analysis of scalar wave optics shows that basically matrix calculation is needed to implement an optical functionality and represent cut planes in 3D space. As the basic matrix calculation capability for large matrices is not sufficient by existing Modelica simulation tools the implementation of an optical library needs to incorporate the external object/function interface.

Up to now a concept for optical connectors and components is introduced and a basic optical functionality using FFT algorithms is already implemented and partly tested with Openmodelica and SimulationX [16]. The approach is promising and further implementations of components for an optical library as well as detailed performance and accuracy investigations also for the calculation of diffraction between tilted planes will be executed.

References

[1] Dyblenko, S. (2009). *Optische Analyse von Bahnwaren mittels , Spektralmethoden - Lösungen und Anwendungen*. Technical report, IPP Symposium, Fakultät Elektrotechnik und Informationstechnik, TU Dresden

[2] Fritzson, P. (2004). *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. IEEE Press

[3] Goodman, J.W. (2005). *Indroduction to Fourier Optics*. The McGraw-Hall Companies, 3d edition

[4] Hardy, J.W. (1998). *Adaptive Optics for Astronomical Telescopes*, Oxford University Press

[5] Hecht, E. (2002). *Optik*, Oldenburg Wissenschaftsverlag, ISBN 3-486-27359-0

[6] Janschek, K. (2010). *Systementwurf mechatronischer Systeme: Methoden - Modelle – Konzepte*, Springer.

[7] Janschek, K., S. Dyblenko, V. Tchernykh, and T. Kaden (2007). *Robuste Verfahren zur Bildaufnahme und Bildauswertung bei Online Messung der Papierformation auf Traversierrahmen*. VDI-Berichte, 1981:57–66.

[8] Janschek, K., V. Tchernykh and S. Dyblenko (2007). *Performance analysis of optomechatronic image stabilization for a compact space camera*. Control Engineering Practice 15(3 SPEC. ISS.): 333-347

[9] Janschek, K. and V. Tchernykh (2002). *Optical correlator for image motion compensation in the focal plane of a satellite camera*. Space Technology, 21(4):127-132

[10] Juday, D.R. and Florence, M.J. (1991). *Full complex modulation with two one-parameter SLMs*, SPIE, vol. 1558, pp. 499-503.

[11] Sommerfeld A. (1999), *Vorlesungen über theoretische Physik – 4, Optik.* Akad. Verlag Geest u. Portig, 3rd edition

[12] Tommasi, T and B.Bianco (1992). *Frequency analysis of light difffraction between rotated planes.* Optics Letters, vol. 17, nr. 8

[13] Matsushima K. (2008). *Formulation of the rotational transformation of wave fields and their application to digital holography.* Applied Optics, vol. 47, nr. 19

[14] Modelica (2010). The MODELICA Language *Specification, Version 3.2*, www.modelica.org

[15] Saldami, L., Bachmann, P. Fritzson, P. and Wiesmann H. (2005). *A Framework for Describing and Solving PDE Models in Modelica*, 4th international Modelica Conference, Hamburg, March 7-8

[16] Uhlig.A, Beutlich, Blochwitz, Kurzbach and Naehring (2009). *Modellierung und Simulation mit Modelica in SimulationX*, www.iti.de

[17] Zemax (2010), software for optical system design. www.zemax.com

[18] http://www.fftw.org/