CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF MECHANICAL ENGINEERING

Department of Instrumentation and Control Engineering

Bachelor's Thesis

Foundations of Expert System for Inspection of Industrial Cranes

Author: Peter Beneš

Academic Year: 2011/2012

Supervisor: Ing. Ivo Bukovský, Ph.D.

CONTENTS

Missing chapters are UPON REQUEST

Abstract

Numerous heavy industrial companies all across Europe, feature cranes in need of major structural surveys. One particular area which a majority of these cranes suffer is at their wheels and runway rails. Two main factors regarding the crane wheels and rails with respect to a structural survey are, overloading and skewing due to long travel, which till this day is an ongoing issue within the crane industry.

One objective of mine in this thesis is to design a computational program, capable of efficiently calculating the crane wheel and rail loads. Via this same program I will also look at what horizontal forces are acting on the crane as it undergoes skewing.

Following this the second main objective is to study deeper into the problem of crane skewing due to long travel. Here I will review current methods of solving crane skewing and then outline ways we may approach crane skewing via a fuzzy rule based system. Furthermore I will outline the potentials of programming language python, by composing a simplified rule based system for controlling crane skewing.

The final goal will be summarisation of the areas discussed in this thesis and, to conclude whether the designed computational program is a good system for future application in crane industrial inspections or surveys. Furthermore whether the application of a fuzzy rule based system for crane skew control is really a possible and good method for solution to the problem.

1) Introduction

In this thesis two parallel objectives are focused on. One is the design of a computational program for efficiently performing the required wheel and rail calculations of an overhead bridge crane. The second being a study into crane skewing due to long travel, with a merge of fuzzy logic and programming language python to discuss a different approach to the problem.

In this section I will outline the problems within the industry, concerning crane wheels and rails, via studied material listed in references [33]& [36]. Furthermore I will outline exactly which areas will be visited in the scope of this thesis.

1.1 - Description of Industrial Problem

Cranes are a vital tool in heavy industry. In this current time hundreds of cranes throughout Europe are overdue for major structural inspections, which by law should be carried out every 10 years for main mechanical components and 20 years for overall major structural check.

This thesis will mainly refer to one of the most common types of industrial crane which is the double girder overhead bridge crane. However the types and sizes are numerous and designed so for the many different requirements of lifting throughout the heavy industry. Focusing on the overhead bridge cranes alone, many areas throughout the structure are of key interest to any engineer who surveys them. One of the most vital elements of any of the types of crane structures is the crane wheels and long travel rails.

On inspection of crane wheels the engineers first consideration is into any wear or damage that occurs on the wheel tread and wheel flanges. When performing a structural survey by law, calculation into the permissible wheel loads must be performed and a recertification must be assigned which is valid for a further 20 years in Europe.

Should a crane be certified for a further 20 years of service, it is then approved in all structural components including the wheels. The problem thus lies within the factors that decrease this rated life for which the wheels has been certified for, particularly problems like misalignment of the plants downshop rails or effects in skewing due to travel.

Similarly can be said for the rails, where by law, calculation into the permissible rail loads must be performed, here also certification is assigned to the maximum permissible rail loads for a further 20 years of crane service. However unlike the wheels which are a more simpler, routine component to replace. Repair and replacement of downshop rails are not as simple and as expected is much more costly. The lengths of these rails vary however for a typical steel work plant crane rails can extend to as long as hundreds of meters.

Rail misalignment can occur due to several reasons, firstly inaccurate measurement of the horizontal span between the rails. Measurement over such large distance in air is a problem in itself; here it is easy to make inaccuracies in measurement which ultimately can cause unparallel rails. When considering vertical height of the rails here too inaccuracies in measurement can lead to uneven height of travel of the crane i.e.; one side may be several centimetres higher with respect to the ground compared to the other.

Misalignments may also occur from other uncontrollable factors like shifting of the soil beneath the workshop which can cause the workshop walls and ultimately rails to lean into the workshop.

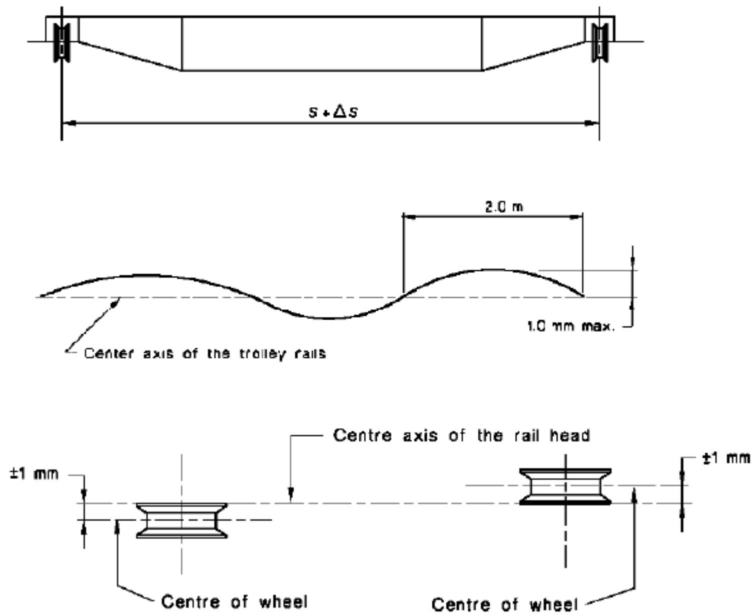

Figure 1.1 –Various overhead bridge crane wheel and rail misalignments adopted from AS1418.14 - 1996 (Figures D1,D7 &D10) Reference No. [34]

In a worst case scenario the misalignment of rails with respect to the wheel span can differ so substantially as to rupturing of the downshop rail being caused. This is a direct consequence of large horizontal forces either pulling or pushing the rails. When considering the crane structure these large axial forces act directly on the bearings within the wheel assembly. Assembly of the wheel block is such that one bearing is free to move small axial distances and one bearing is fixed. Bearings prove to be quiet weak for axial forces and hence it translates to the fixed side bearing taking in the horrible axial forces ultimately leading to failure of the bearing.

Below shows a photograph of this scenario, this photo was taken from a large overhead bridge crane No. 94 in Sydney's, Alcoa steelworks. Here horrible horizontal forces lead to rupturing outward of the crane rail, at this section the rail was joined by a weld which is harder and more brittle material than the rail section as a result rupturing was caused.

Figure 1.2 -Rupture of Downshop Rail due to severe horizontal forces (Ref. No. [36])

Solution to these problems in rail alignment till this day remains to be a tedious and above all a very expensive solution. When looking at the crane travel should there be misalignments, without any control, the wheels and rails will continually wear to the point where they become too dangerous for continued use. Till this day repair or replacement of downshop rails is the only control employed through most major crane manufactures to ensure the crane is kept in operation.

One final factor we should consider, influencing the rated life of the wheels and rails and ultimately an inducer of crane skewing due to long travel is in the driving and load distribution of the crane. Cranes are commonly independently driven on either side of the crane. However should there be difference in speeds of these drives crane skewing can take place, this is achieved for example by uneven load distribution at the drives of the crane. Should one side of the crane be more loaded than the other, a slip can occur with the less loaded side running at higher revolutions than the loaded and, hence skewing would occur.

1.2 – Scope of this thesis

When carrying out structural surveys, recalculation must be performed in order to assess any areas of concern for the crane after its years of service. Most large manufactures have programs for such recalculation however; one limitation is that they are focused on the manufactures standardized components and cranes and, not for a general range of structures. There are independent software developers who have programs to design cranes from the beginning. However when an engineer is carrying out a structural survey for recertification of the structure, he/she must perform calculations which recheck the component for almost any given type of crane structure. The theory for these calculations are based within standards as well as other subjects studied at University like applied mechanics.

Thus one of my objectives in this thesis is to design a computational program capable of efficiently calculating some of the key areas within a structural survey with focus on the wheel and rails. However the computational system presented could be extended to apply to not only wheels and rails but possibly whole crane structures. After providing input values of a typical overhead bridge crane as will be seen later, the produced calculations via this program indeed proved to be quiet effective, highlighting a general calculation process for the wheel and rails component which can be applied to most crane structures.

Another objective of mine in this thesis is investigation of skewing of the crane due to travel. An analysis and comparison will be highlighted as to researched methods for skewing measurement and control. Via the theories of fuzzy logic I will also propose ways on which a fuzzy rule based system could be applied in control of crane skewing due to travel and implement one such system I will introduce in programming language python via fuzzy logic library Peach Fuzzy for python. Furthermore I will look back at the introduced computational program for crane wheel and rail calculation, to analyse the mechanical side of this problem, as such horizontal skewing forces due to angle of skewing, to see how vast such forces are and gain further insight into situations as such in figure 1.2.

Missing chapters are UPON REQUEST

4.4 – Review of Python and potentials for implementation of a fuzzy system.

Python is an open source programming language which originated in 1989, it was in 2000 that python 2.0 was released and since then has grown to become a very successful platform for programming. Python offers a very extensive and growing list of applications, compared to other programming languages python offers a very well equipped standard library, as well as offering a simpler, natural format of programming language.

Now python offers many available libraries capable in solving problems in almost any field and one particular field that will be explored is Fuzzy Logic.

Fuzzy Libraries available to Python

In the domain of Fuzzy logic only several libraries currently exist for Python each offering different ranges of options and functions for fuzzy computation.

1) **Pyfuzzy:** Pyfuzzy is the largest available library for fuzzy logic computation in python. The main webpage for this library offers a comprehensive API documentation allowing the user to see the range of available modules.
   Pyfuzzy features all necessary modules for creation of a fuzzy rule based system from simple to complex, with the ability to choose from a variety of fuzzification and defuzzification methods, sets and operators and complements (ie; Sugeno, Yager, Zadeh).

2) **Fuzzpy:** Fuzzpy is another well supported fuzzy logic frame work, however fuzzpy is primarily focused at fuzzy mathematics as such creation of fuzzy sets, graphs and operations. This library is also setup with a variety of options particularly in choice of fuzzy sets, fuzzy set intersection methods and ability to visualise created fuzzy numbers and graphs.

3) **Peach Fuzzy:** Peach fuzzy is a newer type of fuzzy library in the realm of python, this is actually a section from the Peach module which offers other logic and neural network functions. The main features of this fuzzy library is introduction and testing of membership functions as well as customisation of fuzzy sets, defuzzification and also the ability to implement a fuzzy controller via a range of different norms, implication and aggregation methods.

4) **Gfuzzy:** Gfuzzy is a more basic library for fuzzy logic in python, gfuzzy also offers the ability to create a fuzzy engine like pyfuzzy however; the options are more limited with only choice of 3 sets and 4 operations. Though coding is simpler for the user to understand, there is a lack of supporting material for this library.

Functions and usage of Fuzzy libraries for Python

Among the above list of fuzzy libraries we will focus our attention on fuzzpy and peach fuzzy. Fuzzpy is a great library for performing fuzzy operations and analysing behaviour and properties of introduced membership functions or fuzzy sets. The below explanation is an overview adopted and modified from the Fuzzpy API documentation at http://packages.python.org/fuzzpy/ (ref No.[23]). The first feature of this library is the wide range of available membership functions, after importing the package fuzz followed by module title fnumber we are able to use any of the several membership functions.

- `Polygonal FuzzyNumber` –Here this function is defined by all vertices along the graph where by (x,mu) depicts the position of the x value followed by its corresponding value. This is how fuzzpy enables us to introduce such function.

- `Trapezoidal FuzzyNumber` –This function requires notation as follows `(kernel(x2,x3), support(x1,x4)`, it features the ability to alpha cut via **`alpha`** (`float`) - The alpha value for the cut within range [0, 1]. Furthermore the membership values can be analysed via, Graphname.mu(value) which is similarly achieved for all other membership functions.

- `Triangular FuzzyNumber` –This is a special class of the fuzzpy library, essentially it is build on top of the trapezoidal fuzzy number and therefore features the same properties only that the kernel is defined by only one x value.

- `Gaussian FuzzyNumber`- This function is constructed by via `mean(Float)` and `stdev(float)` which corresponds to the central value of the function respectively the standard deviation, similarly this function has the ability to alpha cut, as well as look up mu values, another feature of this function is to convert into an approximate polygonal fuzzy number.

Unlike the peach fuzzy module which will be later introduced, fuzzpy provides a wide range of properties and characteristics of fuzzy sets that can be looked up and performed on almost any introduced sets.

The first useful property is alpha cut of a function at any membership value. This is achieved by;

```
print (graphname.alpha(float))
```

Here this provides the x position of the kernel (upper values) of the cut. We may also wish to find the intersection or union of two resulting functions this is achieved by symbolic operator (&, |, +, -) where the plus and minus are addition and subtraction operators for combining and subtracting two fuzzy sets.

```
print (graphname1 & graphname2) or (graphname1 | graphname2)
```

Another extra feature is the ability to assess height of any given fuzzy set or membership function as such;

```
print (graphname.height)
```

As fuzzpy is easy to use and great for fuzzy mathematics and analysis of fuzzy sets, this is also the limitation for this library. Though a great tool for explanation of fuzzy sets and use of such sets, defuzzifications or implementation of a simple fuzzy control system is not possible in this library. Another drawback is the sensitivity of this module, unless the fuzzy sets or functions are simply named with clearly distinguished points along the fuzzy sets, the module can incur errors when testing some of the fuzzy mathematic properties available within the module. Fuzzpy also features a visualisation class however this is quite a “trivial task” as said by the writer (A. Mavrinac) himself. Thus a more suitable choice for visualisations is the matplotlib, which is also one of the more common libraries used for visualisation of all other types of mathematical functions and data.

When considering this, Pyfuzzy becomes worth a mention, Pyfuzzy is a well documented library that features the greatest capabilities and options in the field of fuzzy mathematics and logic however this library is not so user friendly. This library is capable of creating even complex controllers also exampled on their API documentation at http://pyfuzzy.sourceforge.net/. However for the scope of this thesis we will explore peach fuzzy which also offers defuzzification processes, programming of a simple controller and even generation of a control surface for the given data with more user friendly commands making this a more preferable library.

Peach fuzzy uses a more natural approach in definition of membership functions this library features eight possible functions. Below some of which are listed, the below explanation is an overview adopted and modified from the Peach Fuzzy API documentation at http://peach.googlecode.com/hg/doc/build/html/index.html (ref. No. [32]).

`IncreasingRamp(x1, x2)` – For value x<x1 mu value-0 x>x2 mu value=1.
`DecreasingRamp(x1, x2)` –For value x<x1 mu value-1 x>x2 mu value=0.
`Triangle(x0, x1, x2)` – x0,x1,x2 are the vertices of the triangle.
`Trapezoid(x0, x1, x2, x3)`–x0,x1,x2, x3 are the vertices of the trapezoid.
`Gaussian(x0, a)` – Uses x0 as centre of the function with width a.

Another key area of peach fuzzy is its defuzzification function. Here five possible methods are available all used to provide a conclusion from any resulting graph which is defined under the defuzzification process , these methods are as follows;

```
Centroid method = Centroid(mf, y)
Bisection method = Bisector(mf, y)
Smallest of Maxima = SmallestOfMaxima(mf, y)
Largest of Maxima= LargestOfMaxima(mf, y)
Mean of Maxima= MeanOfMaxima(mf, y)
```

Here the membership value is input with the range of the output values (y) also defined, this is a great tool as we are able to analyse and compare the ways a proposed controller would implement a resulting value.

As also mentioned perhaps the most useful feature of this library is the potential to implement a fuzzy logic controller. The main component of this is within the controller class of the peach fuzzy module. The controller class requires several key components to be defined in order to correctly deduce a concluding numerical value as follows:

```
Controller= Controller(output range, rules of the controller,
defuzzification method, norm to bind the rules, implication method
and aggregation method).
```

For use of Peach fuzzy and detailed codes of usage within this thesis refer to section 4.5- Application of Fuzzy Logic to Crane skewing due to travel with use of python. Here membership functions were introduced with design of a simple Fuzzy logic controller as per figure 4.5b).

Comparison of Python Fuzzy Libraries to Matlab

After looking at the capabilities of the above mention library we can make a quick comparison with Matlab, which is one of the most widely used software for scientific and computational purposes. Simulink featured within Matlab is a very advanced medium for fuzzy logic applications. The Fuzzy toolbox is equipped with a graphical interface for defining and tuning a fuzzy controller as well as several options in the implication and defuzzification methods, its key advantage is that it gives the user a more insightful view at construction of a fuzzy logic process, with features like a slider menu to test membership values for given inputs on the introduced fuzzy sets and automatic construction of a control surface model for analysis of the input data. Though this may be a better medium for fuzzy control, in the table below we will outline some key points of about Matlab where Python features more of an advantage as a medium for fuzzy mathematics and logic.

Matlab	Python (Fuzzy Libraries)
-Though Featuring a graphical interface for introduction and tuning of a fuzzy system, the graphical sliders can be sensitive and thus tuning is better via Matlab code in the FIS editor text file.	-Python offers more freedom with introduction and tuning of a fuzzy system, however visualisations are not automatic in most libraries and thus, this is one main plus in the sense of Matlab.
-Though great for implementing a controller or fuzzy logic process, Matlab does lack analysis of properties of the introduced fuzzy sets.	-Libraries like Fuzzpy offer a great range of commands for analysis of fuzzy properties and characteristics.
-Matlab is not free, and often available to students only by university licences.	-Python is a free open source program as well as majority of the fuzzy modules.
-Matlab is not as versatile when it comes to choice of standard membership functions and even defuzzification and implication methods as some python libraries.	-Fuzzy Modules like Peach fuzzy provide a great medium for customised fuzzy logic systems. It has a broad selection of methods as discussed above, with great response in defuzzification to evaluate a conclusion.

5) Conclusion

5.1 – Summarization and Evaluation

After completion of all discussed sections of this thesis, we can conclude several key points in regards to the overall subject of this thesis. As this papers primary goal was to look at the problem of crane wheels and rails we can conclude that first of all one must consider what loads are acting on the crane wheels and rails, as it is this alone, that is often a key issue with older cranes. Via the computational program that I've created in chapter 3, one can see the significance the loads and their distribution has, on the crane wheels and rails, furthermore the magnitude of horizontal forces that can occur in the wheels and rails due to factors of skewing.

If one desires to obtain the maximal life of their crane wheels and rails a consideration into inspection of the crane skewing should take place, control of which is not yet a readily available option for crane operators. Thus the final component of this thesis looked into such available methods and after research we can gather that due to the circumstances of the problem, such control system should require a certain sense of human intuition for the problem to optimally control each particular case of crane skewing that could be present, with deeper consideration into the particular misalignments of the crane rails and wheel geometry. Thus control was investigated in the scope of a fuzzy rule based system and from this we see that such method of control could be possible. With the aid of programming language python a simple fuzzy rule based system was implemented and from here we obtained the expected results desired for the given cases tested.

With this result we can see more research should be provided in the scope of crane skewing due to travel, the ultimate goal of which would be to implement an economical system which can be applied in a general sense of cranes. Due to different drive configurations for small and large, modern and old cranes, the better solution would be manipulation of the crane brakes whilst taking into account what misalignments are present on the crane and whether they are worth correcting or not.

Missing chapters are UPON REQUEST

7) References

[1] Fuzzy Logic Example using Matlab, URL http://arri.uta.edu/acs/ee5322/lectures/Ballal%20fuzzy%20logic%20notes.pdf Visited 23/1/12.

[2] Fuzzy Inference Systems in, URL http://www.mathworks.com/help/toolbox/fuzzy/fp351dup8.html, Visited 23/1/12.

[3] Fuzzy Logic Toolbox Documentation, URL http://www.rohan.sdsu.edu/doc/matlab/pdf_doc/fuzzy/fuzzy_tb.pdf Visited 24/1/12.

[4] Case Studies and Applications of Fuzzy Logic URL, http://www.cairo-aisb.com/webcairo/download/Fuzzy_logic/Fuzzy_Module_3.pdf, Visited 24/1/12

[5] Jerry Mendel: Uncertain Rule-Based Fuzzy Logic Systems, Introduction and New Directions, Prentice Hall, Upper Saddle river, NJ, 2001.

[6] IF-THEN Rules and Fuzzy Inference, URL http://www.intelligent-systems.info/classes/ee509/8.pdf, Visited 4/4/12

[7] Overhead Bridge Crane image URL, http://www.hiwtc.com/photo/products/16/00/97/9703.jpg , Visited 22/1/12.

[8] Crane Anti-Skew and Position control, URL http://www.magnetekmh.com/solutions_engineeredsys-craneantiskew.htm, Visited 5/2/12

[9] S.Bagaric, G. Pietsch, March 19th 1985, Crane anti skewing device, Patient No. US 4505207 Visited 5/2/12, URL http://www.google.com/patents?id=07w9AAAAEBAJ&printsec=abstract&zoom=4#v=onepage&q&f=false.

[10] N. Mitrovic, V.Kostic, M.Petronijevic and B.Jeftenic, 2010, Practical Implementation of Multi-Motor Drives for Wide Span Gantry Cranes from, Iranian Journal of Science & Technology, Vol. 34 pgs 649-654.

[11] New Connection for the Typical Synchronization Systems, URL http://www.insipub.com/ajbas/2009/4253-4260.pdf, Visited 12/3/12.

[12] Ezsdc Crane Runway Beam Design Software, URL http://www.ezsdc.com/software1.aspx, Visited 20/1/12

[13] EOT Crane Design Software –FDES, URL http://www.fdes.in/cranedesignsoftware.aspx, Visited 20/1/12.

[14] Automated Design Structural Software –CP&A, URL http://www.cranedesign.com/software.htm, Visited 20/1/12

[15] MIT Calc-Mechanical, Industrial and Technical Calculations, URL http://www.mitcalc.com/index_cz.htm, Visited 20/1/12.

[16] Cranes Design, Practice and Maintenance, URL http://www.cranecontrol.nl/img/content/kranen_ontwerp_praktijk_en_onderhoud.pdf, Visited 22/1/12

[17] Python Programming Language URL www.python.org , Visited 2/2/12.

[18] Koo Kim, Artificial Neutral Networks in Python, Bachelor's thesis, FS CVUT v Praze, 2011.

[19] Fuzzy Application Library/Technical Applications/Industrial Automation, URL http://www.fuzzytech.com/e/e_a_plc.html, Visited 5/2/12.

[20] PyFuzzy-Python Fuzzy Package, URL http://pyfuzzy.sourceforge.net/, Visited 16/2/12

[21] Fuzzpy Documentation, URL http://mavrinac.com/index.cgi?page=fuzzpy, Visited 16/2/12

[22] Examples in Fuzzpy, URL https://github.com/ezod/fuzzpy/blob/master/examples/creating_sets.py, Visited 16/2/12.

[23] Fuzzpy API Documentation URL, http://packages.python.org/fuzzpy/, Visited 16/2/12.

[24] Bearing Calculations SKF, URL http://www.skf.com/portal/skf/home/products?lang=en&maincatalogue=1&newlink=1_0_25 , Visited 20/2/12

[25] Equivalence of loading, URL http://www.nskamericas.com/cps/rde/xbcr/na_en/Life.pdf, Visited 20/2/12

[26] Gfuzzy –a python fuzzy engine, URL https://code.google.com/p/gfuzzy/source/browse/#svn%2Ftrunk, Visited 17/2/12.

[27] ADVANCES IN CONTROL AND SENSORS TECHNOLOGY ENABLING UNMANNED OPERATION OF CONTAINERS CRANES, URL http://bib.irb.hr/datoteka/204932.IL-03_Slutej_Kolonic_EDPE2005.pdf, Visited 23/2/12

[28] Miroslav Reichl, Řídicí systém minimalizující opotřebení kol jeřábu, Master's Thesis, FEL CVUT v Praze 2010.

[29] Demag Crane & Components-Geared Motors Catalogue, 2011-2012 Edition, Printed in Wetter, Germany 2011.

[30] Leuze Electronic-Optische Distanzsensoren Catalogue, 2011-2012 Edition, Printed in Germany, 2011.

[31] Matplotlib –Data Ploting, URL http://matplotlib.sourceforge.net/users/pyplot_tutorial.html, Visited 20/4/12

[32] Peach v0.3.1 Documentation, URL http://peach.googlecode.com/hg/doc/build/html/index.html, Visited 26/4/12

[33] Australia Standards Association, AS 1418.1-2002 –Cranes, hoists and winches Part 1 –General requirements, Fourth Edition, Published in Sydney, Australia, ISBN 0 7337 4372 2

[34] Australia Standards Association, AS 1418.14-1996 – Requirements for cranes subject to arduous working conditions , Published in Sydney, Australia, 1996.

[35] J. Goliaš, 2012, Discussion on Implementation of Crane Skewing Control System. (Personal Communication 16th March.

[36] I.Beneš, Structural Survey of Crane No. 94 for Alcoa Steelworks Sydney, Published by IBCE Consulting Engineering, 2010.

[37] I.Beneš, Structural Survey of Crane OET 163 for Onesteel Steelworks Whyalla, Published by IBCE Consulting Engineering, 2011.

[38] Anti-Sway Fuzzy Logic Controller Presentation – URL http://www.itengineerings.com/blog/fuzzy-logic/fuzzy-logic-anti-sway-controller-for-container-crane-control. Visited 16/2/12.