Norwegian University of Science and Technology

Master's degree thesis

IP501909 MSc thesis, discipline oriented master

Virtual Winch Prototyping-Design, Modeling, Simulation and Testing of A Marine Hydraulic Winch System with Active Heave Compensation

Candidate number:1107/Author: Dahai He

Number of pages including this page: 74

Aalesund, June 3, 2016

Mandatory statement

Each student is responsible for complying with rules and regulations that relate to examinations and to academic work in general. The purpose of the mandatory statement is to make students aware of their responsibility and the consequences of cheating. **Failure to complete the statement does not excuse students from their responsibility.**

Please complete the mandatory statement by placing a mark ***in each box*** *for statements 1-6 below.*		
1.	**I/we hereby declare that my/our paper/assignment is my/our own work, and that I/we have not used other sources or received other help than is mentioned in the paper/assignment.**	☒
2.	**I/we herby declare that this paper** 1. Has not been used in any other exam at another department/university/university college 2. Is not referring to the work of others without acknowledgement 3. Is not referring to my/our previous work without acknowledgement 4. Has acknowledged all sources of literature in the text and in the list of references 5. Is not a copy, duplicate or transcript of other work	Mark each box: 1. ☒ 2. ☒ 3. ☒ 4. ☒ 5. ☒
3.	I am/we are aware **that any breach of the above will be considered as cheating, and may result in annulment of the examination and exclusion from all universities and university colleges in Norway for up to one year, according to the Act relating to Norwegian Universities and University Colleges, section 4-7 and 4-8 and Examination regulations .**	☒
4.	**I am/we are aware that all papers/assignments may be checked for plagiarism by a software assisted plagiarism check**	☒
5.	**I am/we are aware that NTNU will handle all cases of suspected cheating according to prevailing guidelines.**	☒
6.	**I/we are aware of the NTNU's rules and regulation for using sources.**	☒

Publication agreement

ECTS credits: 120

Supervisor: Vilmar Æ søy/: Houxiang Zhang

Agreement on electronic publication of master thesis

Author(s) have copyright to the thesis, including the exclusive right to publish the document (The Copyright Act §2).
All theses fulfilling the requirements will be registered and published in Brage, with the approval of the author(s).
Theses with a confidentiality agreement will not be published.

I/we hereby give NTNU the right to, free of charge, make the thesis available for electronic publication: ☒**yes** ☐**no**

Is there an agreement of confidentiality? ☐**yes** ☒**no**
(A supplementary confidentiality agreement must be filled in and included in this document)
- If yes: **Can the thesis be online published when the period of confidentiality is expired?** ☐**yes** ☐**no**

This master's thesis has been completed and approved as part of a master's degree programme at NTNU Ålesund. The thesis is the student's own independent work according to section 6 of Regulations concerning requirements for master's degrees of December 1st, 2005.

Date: June 3, 2016

Preface

This thesis contains all the work done in my master thesis at NTNU in Ålesund. The thesis contains the development of a Virtual Prototyping for marine winch system design. This Virtual Prototyping for overall winch system design is a new platform for product development process and virtual prototyping in the maritime operations.

A huge thanks to Vilmar Æsøy, my supervisor. His way to inspire me not to take shortcuts, but to keep going strong and doing a thorough job has a huge influence on the results in this thesis. He has a perpetual source of knowledge that I have benefited.

Another especially thanks to Houxiang Zhang, my co-supervisor. His consistently patient guidance and support gave me pretty much courage to carry on until implement this master thesis.

Last but not least, a huge thanks to Ph.D. candidates in NTNU I Ålesund, Yingguang Chu and Lars Ivar Hatledal. Meanwhile, the researcher in NTNU I Ålesund, Yuxiang Deng. They have shared quite a few experience and ideas to inspire me throughout the whole process.

Dahai He

Ålesund, June 3, 2016

Summary

This thesis is to develop a standard virtual prototyping system for hydraulic winch system including developing a library of standard sub-models of hydraulic system, mechanical system and control system (AHC), and visualizing the simulation and operation of the virtual winch prototyping system. To be more specific:

Chapter 1. Motivation and background of winch prototyping is introduced so as to break down the problems and formulate the objectives of this projects.

Chapter 2. Theoretical background is shown in this part. It contains the briefly descriptions of the important theory applied in virtual prototyping winch system.

Chapter 3. Methodology is shown in this part. It contains the detailed theory basis applied in the modelling of hydraulic and mechanical sub-systems of winch system.

Chapter 4. This chapter describes the implementation of mechanical sub-system. 3D modelling, parameterization and visualization are implemented by using WebGL technology with three.js library. The outcome of the mechanical part can be easily integrated into the virtual prototyping framework.

Chapter 5. This chapter elaborates the method and the process of hydraulic and control (AHC) sub-model modelling. Bond graph theory is applied during the modelling process. Different alternatives of hydraulic system structure are analysed and compared to finalize a better solution of hydraulic system structure.

Chapter 6: This chapter explains the integration method of virtual prototyping winch system framework.

Chapter 7: Results of mechanical modelling, hydraulic with control modelling and co-simulation of integrated virtual prototyping winch system are shown and discussed to evaluate the performance of virtual prototyping system.

Chapter 8: This part makes conclusions, modelling alternatives and future work for the virtual prototyping winch system.

Contents

Lists of Figures and Tables

Abbreviations

NTNU	Norwegian University of Science and Technology
AHC	Active Heave Compensation
VP	Virtual Prototyping
VWP	Virtual Winch Prototyping
FEA	Finite Element Analysis
BG	Bond Graph
CAD	Computer Aided Design
WebGL	Web Graphic Library
MRU	Motion Reference Unit
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit

1. Introduction

1.1 Motivation and Background

Winch, a machine for lifting and lowering cargo, and for other purposes that cannot be performed by manual power, generally consists of a drum or barrel around which a rope or cable is wound to achieve either a lifting or lowering motion; the drum rotates in its horizontal axis and can be powered by steam or hydraulic motor or electric motor.

Among all kinds of winches, hydraulic winch is one the most commonly and widely employed device in offshore operation considering the complexity of marine operation, highly demands of efficiency and accuracy, and extreme working environment of pull line etc. However, the rapidly growing offshore industry increasingly exposes big challenges on the existing designing process of hydraulic winch system which is still implemented in a traditional way, for current product and system development requires fast and dynamic changeable process. Meanwhile, since lead-time of design and production process is continually decreasing, any mistakes or system fails may cause vital accidents, project delays and excessive costs.

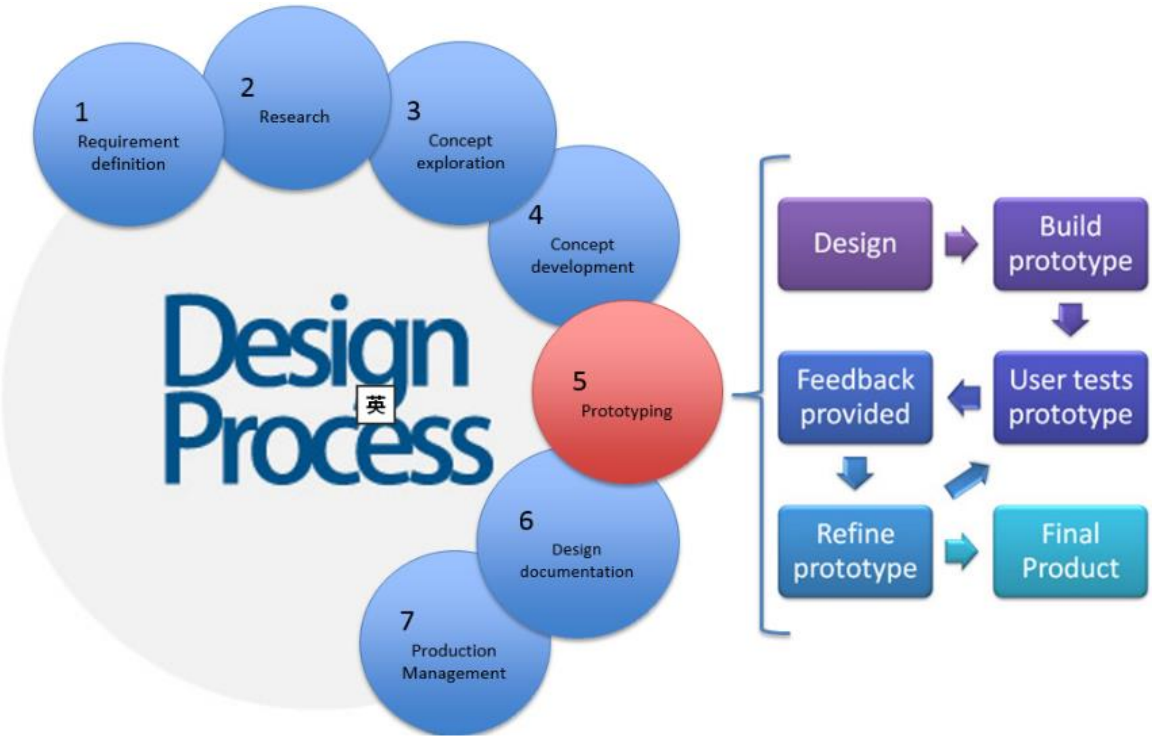


Figure 1.1 Traditional product design and development process

Traditional product design and development processes (see Figure 1.1) mostly rely on engineers' experience and judgment in conceptual design phase. Generally, a physical prototype shall be constructed and tested in order to validate the initial ideas and to evaluate its performance. Previous statistical reports showed that most of the initial physical prototypes was highly unlike to meet the initial expectations. As a result, large amount of money and time were wasted on re-designing the concept and construction of new physical prototypes back and forth. Therefore, a new method called Virtual Winch Prototyping (VWP) is proposed so as to overcome the bottlenecks in overall system design, operation and training.

If large number of products are being developed in the form of virtual prototypes in which engineering simulation software are used to predict performance prior to constructing physical prototypes, engineers can quickly explore the performance of thousands of design alternatives without investing the time and money required to build physical prototypes.

Virtual Prototyping (VP), also referred to as 'digital prototyping' or 'virtual modelling', has been stimulated by interests in simulation and computer modelling techniques. Depending on the domain of application, different definitions can apply, but Tim Hodgson (Comptek Federal Systems Inc.) offers an apt one for product design: Virtual prototyping is a software-based engineering discipline that entails modelling a mechanical system, simulating and visualising its 3D-motion behaviour under real-world operating conditions, and refining/optimising the design through iterative design studies prior to building the first physical prototype. Generally, we use virtual prototyping to design, optimize, validate, and visualize the products digitally and evaluate different design concepts before incurring the cost of physical prototypes. By doing this, we can visualize realistic machine operation, estimate the cycle time throughput, and glean important information about the dynamic behaviour of the system design.

1.2 State of The Art and Problem Formulation

The existing domain-specific modelling and simulation software tools, such as Flexcom for structure FEA, MSC Adams for multi-dynamics, PSCAD for the power systems and dSpace for control system etc., have already proved of significant values in its discipline in the process of product development and system design. However, since a great number of cross-disciplinary complex engineering systems exist in maritime industry, "stand-alone" software are not capable of interconnection to interface these sub-models for the design process of product and system design.

Functional Mock-up Interface (FMI), a standard interface to be used in computer simulations to develop complex cyber-physical systems, was cooperatively developed in the project called MODELISAR. In practice, the FMI implementation by a software modelling tool enables the creation of a simulation model that can be interconnected or the creation of a software library called FMU (Functional Mock-up Unit). FMU consists of XML-file describing the dynamic variables to be exchanged and the simulation model either as C source code or a complied linked library. The FMI functions are used by a simulation environment to create one or more instances of the FMU and to simulate them, typically together with other models. An FMU may either have its own solvers (FMI for Co-Simulation) or require the simulation environment to perform numerical integration (FMI for Model Exchange).

With the help of FMI, sub-models from different disciplines can be developed in "stand-alone" software tools and then be integrated in a separate integration platform. A Virtual

Prototyping Framework has developed by using Java to import FMUs for co-simulation in NTNU in Aalesund. Thus sub-models from different disciplines can be integrated from software tools which are able to export FMU.

As described in previous paragraphs, so far, in order to achieve the VWP simulation system, several key fearures must be implemented:

- Winch mechanical sub-models library which can be parametrically configured by user
- Winch hydraulic sub-models library with generic control algorithm.
- Integrate sub-models into VWP framework for co-simulation.
- Winch system 3D visualization.

1.3 Objectives Formulation

In order to develop a standard virtual prototyping system for hydraulic winch system including developing a library of standard sub-models of hydraulic system, mechanical system and control system (AHC), and visualizing the simulation and operation of the virtual winch prototyping system. The working packages are as follow:

- Develop methods for parameterisation of mechanical sub-models
- Develop a library of standard hydraulic sub-models and control sub-models
- Integrate winch prototyping framework with domain-specific sub-systems
- Co-simulate, test and visualize the whole system performance of the Virtual Winch Prototyping system

2. Theoretical Background

2.1 Bond Graph Theory

Modelling of the hydraulic system can be done in 20sim based on Bond Graph (BG) method. BG method is a modelling approach based on identifying the energetic structure of the system. By this approach, a physical system can be represented by symbols and lines, identifying the power flow paths. The lumped parameter elements of resistance, capacitance and inertia are interconnected in an energy conserving way by bonds and junctions resulting in a network structure. From the pictorial representation of the bond graph, the derivation of system equations is so systematic that it can be algorithmized.

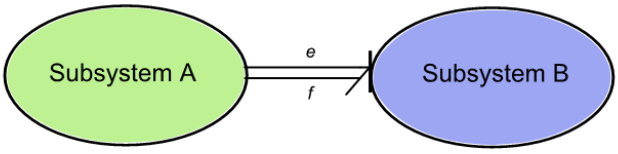


Figure 2.1 Power bond connecting subsystem A and B

The language of bond graphs aspires to express general class physical systems through power interactions. The factors of power i.e., Effort and Flow (see the Power bond connecting subsystem A and B in Figure 2.1), have different interpretations in different physical domains. Yet, power can always be used as a generalized co-ordinate to model coupled systems residing in several energy domains. In Table 2.1, effort and flow variables in some physical domains are listed.

Table 2.1: Basic bond graph elements summary

Systems	Effort (e)	Flow (f)
Mechanical	Force (F)	Velocity (v)
	Torque (τ)	Angular velocity (ω)
Electrical	Voltage (V)	Current (i)
Hydraulic	Pressure (P)	Volume flow rate (dQ/dt)
Thermal	Temperature (T)	Entropy change rate (ds/dt)
	Pressure (P)	Volume change rate (dV/dt)
Chemical	Chemical potential (μ)	Mole flow rate (dN/dt)
	Enthalpy (h)	Mass flow rate (dm/dt)
Magnetic	Magneto-motive force (e_m)	Magnetic flux (ϕ)

In bond graph theory different elements are used. These are summarized in Table 2.2. Both the Se and Sf elements are sources, effort and flow respectively. If hydraulics are modeled then Se is a pressure source and Sf is a fluid flow source. The R element is describing energy dissipation like friction forces or viscous forces. It can also be used to model valves as seen later on. The C element describes the stored energy in the system, like a spring in a mechanical system or an accumulator in a hydraulic system. Inertia in a mechanical system or an inductor in an electrical circuit is given as a I element. Transformation of efforts and flows between subsystems is usually done by using a $T\ F$ element. To sum different contributions of effort the 1-junction is used and to sum different contributions of flow the 0-junction is used. There is also one more basic element that is not included in the table. This is the gyrator element, GY that transforms flows to efforts and vice versa. This element can be associated with a generator that gets a rotational velocity, a flow, and transforms it to voltage, an effort. The relation among those elements can also be drawn in the trihedral chart shown below in Figure 2.2

Table 2.2 Bond graph theory basic elements

Symbol	Relation
$S_e \rightharpoondown\!\mid$	$e = e(t)$, given
$S_f \mid\!\rightharpoondown$	$f = f(t)$, given
$\mid\!\rightharpoondown$ R $\rightharpoondown\!\mid$ R	$e = \Phi_R(f)$ $f = \Phi_R^{-1}(e)$
$\mid\!\rightharpoondown$ C $\rightharpoondown\!\mid$ C	$e = \Phi_C^{-1}(\int_0^t f dt)$ $f = \frac{d}{dt}[\Phi_C(e)]$
$\rightharpoondown\!\mid$ I $\mid\!\rightharpoondown$ I	$f = \Phi_I^{-1}(\int_0^t e dt)$ $e = \frac{d}{dt}[\Phi_I(f)]$
$\overset{1}{\mid\!\rightharpoondown} TF \overset{2}{\mid\!\rightharpoondown}$ $\overset{1}{\rightharpoondown\!\mid} TF \overset{2}{\rightharpoondown\!\mid}$	$e_1 = me_2$ $f_2 = mf_1$ $e_2 = \frac{1}{m}e_1$ $f_1 = \frac{1}{m}f_2$
$e_2 \mid f_2$ $\xrightarrow[f_1]{e_1} \mathbf{1} \xrightarrow[f_3]{e_3}$	$e_1 - e_2 - e_3 = 0$ $f_1 = f_2 = f_3$
$e_2 \mid f_2$ $\xrightarrow[f_1]{e_1} \mathbf{0} \xrightarrow[f_3]{e_3}$	$e_1 = e_2 = e_3$ $f_1 - f_2 - f_3 = 0$

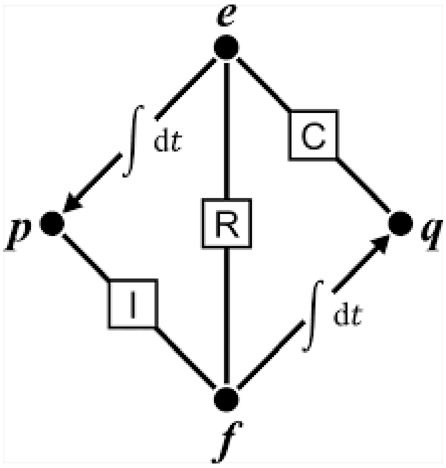


Figure 2.2 Thetrahedron relation chart for bond graph elements

2.2 Parameterization Design

3D solid models of winch are often built in Computer Aided Design (CAD) software such as SolidWorks, NX, CATIA, etc. During the conceptual design phase, it is beneficial to check the main geometrical and functional features by simulations and visualization of the design. However, traditional solid model cannot meet the real-time simulation requirement, for solid models contain all the design details. In order to implement both parameterization and visualization features, an approach based on WebGL technology for marine winch conceptual design is developed. This method is to generate mesh models and geometric models for visualization and simulation directly on the website with a set of predefined parameters that determine the main geometric dimensions of the winch. The main advantage of the web-based tool are the flexibility for design concept verification by defining the main geometric dimensions, and the user interface through the web-browser for data exchange and information sharing. The generated models can be wrapped and directly used for simulations in the VWP framework.

2.3 Active Heave Compensation

Active Heave Compensation (AHC) can be used to control the relative position of a load to a fixed object. AHC differs from PHC (Passive Heave Compensation) by having a control system, for example a Programmable Logic controller (PLC), which actively tries to compensate for any movement. The control system requires the knowledge about the vessel's heave motion. AHC systems often use a Motion Reference Unit (MRU) which measures the vessel's heave, pitch and roll motion.

AHC systems makes the hoisting, lowering and handling of loads on floating vessels safer. Without such systems many operations performed on rough sea would be extremely difficult or highly risky. AHC systems are widely used to minimize unwanted drill string movement. In Figure 2.3, it shows a typical diagram of winch system with AHC.

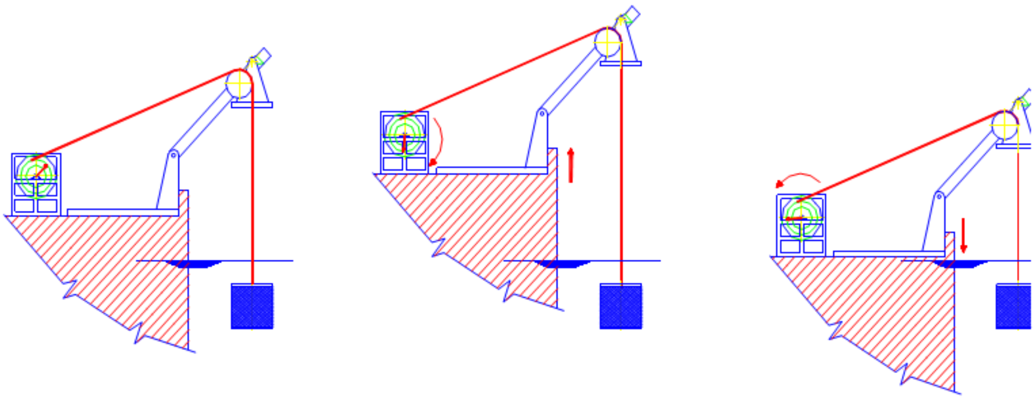


Figure 2.3 Diagram of winch system with AHC

2.4 AGX dynamics

AgX Dynamics is a professional multi-purpose physics engine for simulators, engineering, large scale granular simulators and more. It consists of hundreds of C++ classes of highly optimized and portable code and is the obvious choice when we need to simulate mechanical constrained systems with frictional contacts. It is truly scalable in all senses of the word. Built upon a solid foundation of original scientific research, including discrete variational and physics based time integration methods for constrained systems, parallel high performance hybrid equation solvers and novel multi-physics models. Therefore AgX Dynamics combines accuracy with speed in a way that cannot be found in any other competing physics engine or product. A typical simulation in AgX is shown in Figure 2.4.

The wire simulations in AgX Dynamics are fast enough for demanding real time simulators and yet the models have high physical accuracy and are stable under extreme loads and mass ratios. Dynamic resolution which depends on load and curvature ensures stable simulations. Wires can be tuned to represent ropes, wires and chains with realistic material parameters including bend and stretch Young's modulus.

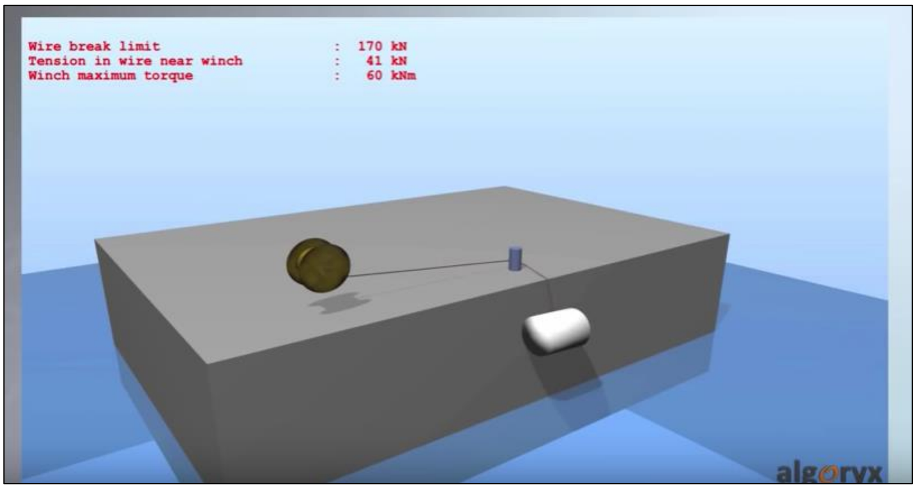


Figure 2.4 Typical simulation in AgX

2.5 WebGL

WebGL brings 3D to the browser, providing a JavaScript interface to the graphics hard -ware on your machine. Based on OpenGL ES (the same graphics running in the smartphone and tablet), it is developed and supported by the makers of major desktop and mobile web browsers. With WebGL, any programmer can create stunning graphics that reach millions of users via the Web. Programming WebGL directly, however, is very complex. We need to know the inner details of WebGL and learn a complex shader language to get the most out of WebGL. The Three.js library provides a very easy-to-use JavaScript API based on the features of WebGL, so that we can create beautiful 3D graphics, without having to learn the WebGL details.

2.6 FMI for Co-simulation

The FMI for Co-Simulation interface is designed both for the coupling of simulation tools (simulator coupling, tool coupling, see Figure 2.6), and coupling with subsystem models (see Figure 2.5), which have been exported by their simulators together with its solvers as runnable code. The goal is to compute the solution of time dependent coupled systems consisting of subsystems that are continuous in time (model components that are described by differential-algebraic equations) or time-discrete (model components that are described by difference equations, for example discrete controllers).

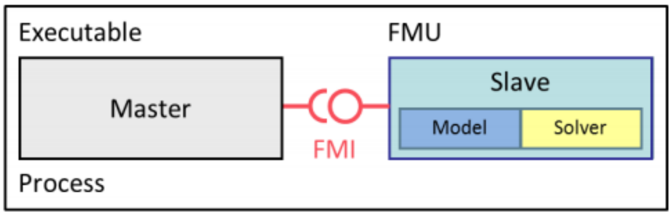


Figure 2.5 Co-simulation with generated code on a single computer

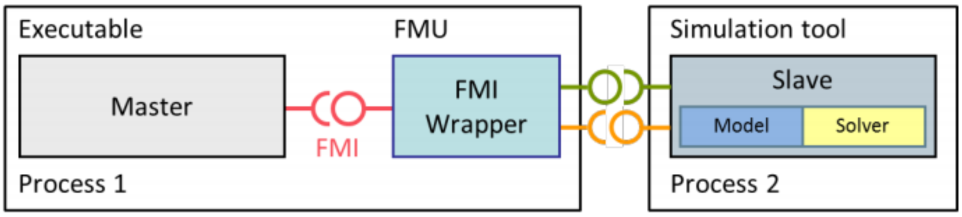


Figure 2.6 Co-simulation with tool coupling on a single computer

In case of tool coupling, the modular structure of coupled problems is exploited in all stages of the simulation process beginning with the separate model setup and pre-processing for the individual subsystems in different simulation tools. During time integration, the simulation is again performed independently for all subsystems restricting the data exchange between subsystems to discrete communication points. Finally, also the visualization and post-processing of simulation data is done individually for each subsystem in its own native simulation tool.

3. Methodology

3.1 Applied Theories

3.1.1 Hydraulic

- **Fluid Velocity:**

The velocity of a section is not constant throughout the cross sectional area. Instead, it varies with location. The velocity is zero where the fluid is in contact with the conduit wall. The variation off low velocity with in a cross-section complicates the hydraulic analysis, so the engineer usually simplifies the situation by looking at the average (mean) velocity of the section for analysis purposes. This average velocity is defined as the total flow rate divided by the cross sectional area, and is in units of length per time.

$$\mathrm{V} = \frac{Q}{A}$$

Where V = average velocity [m/s]   Q = flow rate (m^3/s)   A = area (m^2)

- **Reynolds Number:**

To classify flow as either turbulent or laminar, an index called the Reynolds number is used. If the Reynolds number is below 2000, the flow is generally laminar. For flow in closed conduits, if the Reynolds number is above 4000, the flow is generally turbulent. Between 2000 and 4000, the flow maybe either laminar or turbulent, depending on how insulated the flow is from outside disturbances. It is computed as follows:

$$R_e = \frac{4VR}{v}$$

Where R_e = Reynolds number (unitless) V = average velocity (m/s)

R = hydraulic radius (m)   v = kinematic viscosity

- **Valves Dynamics:**

For all valves, the flow is controlled by flow area size. Therefore, we have equation.

$$C_d * A = \frac{Q}{\sqrt{\frac{2\Delta P}{\rho}}}$$

Where C_d = discharge flow coefficient (0.6~0.9)   A = flow area (m^2)

ρ = density [kg/m^3]   Q = flow [m^3/s] ΔP = pressure difference [Pa]

- **Hydraulic Pump:**

$$Q = \frac{V \bullet n \bullet \eta_{vol}}{1000} \text{ [l/min]}$$

$$P_{an} = \frac{p \bullet Q}{600 \bullet \eta_{ges}} \text{ [kW]}$$

$$M = \frac{1{,}59 \bullet V \bullet \Delta p}{100 \bullet \eta_{mh}} \text{ [Nm]}$$

$$\eta_{ges} = \eta_{vol} \bullet \eta_{mh}$$

Q = Volume flow [l/min]
V = Nominal volume [cm^3]
n = Drive speed of the pump [min^{-1}]
P_{an} = Drive power [kW]
p = Service pressure [bar]
M = Drive torque [Nm]
η_{ges} = Total efficiency (0,8-0,85)
η_{vol} = Volumetric efficiency (0,9-0,95)
η_{mh} = Hydro-mechanic efficiency(0,9-0,95)

- **Hydraulic Motor:**

$$Q = \frac{V \bullet n}{1000 \bullet \eta_{vol}}$$

$$n = \frac{Q \bullet \eta_{vol} \bullet 1000}{V}$$

$$M_{ab} = \frac{\Delta p \bullet V \bullet \eta_{mh}}{20 \bullet \pi} = 1{,}59 \bullet V \bullet \Delta p \bullet \eta_{mh} \bullet 10^{-2}$$

$$P_{ab} = \frac{\Delta p \bullet Q \bullet \eta_{ges}}{600}$$

Q = Volume flow [l/min]
V = Nominal volume [cm^3]
n = Drive speed of the pump [min^{-1}]
η_{ges} = Total efficiency (0,8-0,85)
η_{vol} = Volumetric efficiency (0,9-0,95)
η_{mh} = Hydro-mechanic efficiency (0,9-0,95)
Δp = Pressure difference between motor inlet and outlet (bar)
P_{ab} = Output power of the motor [kW]
M_{ab} = Output torque [Nm]

- **Friction Loss in Pipeline:**

$$\Delta P = f \cdot 0.5 \cdot \frac{\rho}{d} \cdot L \cdot v^2$$

Here, f = friction factor [unitless]   ρ = fluid density [kg/m^3]

d = pipe diameter [m] v = fluid speed [m/s]

For laminar flow (Re<2000), $f = \frac{64}{Re}$

For turbulent flow (Re>2000), $f = \frac{0.25}{[log_{10}(\frac{r}{3.7d}+\frac{5.74}{Re^{0.9}})]^2}$

Here,

Re = Reynolds Number [unitless] r = pipe roughness height [m] d = pipe diameter [m]

3.1.2 Mechanical

With respect to winch design specifications, some terms shall be defined. Figure 3.1 shows the illustrations of the following terms.

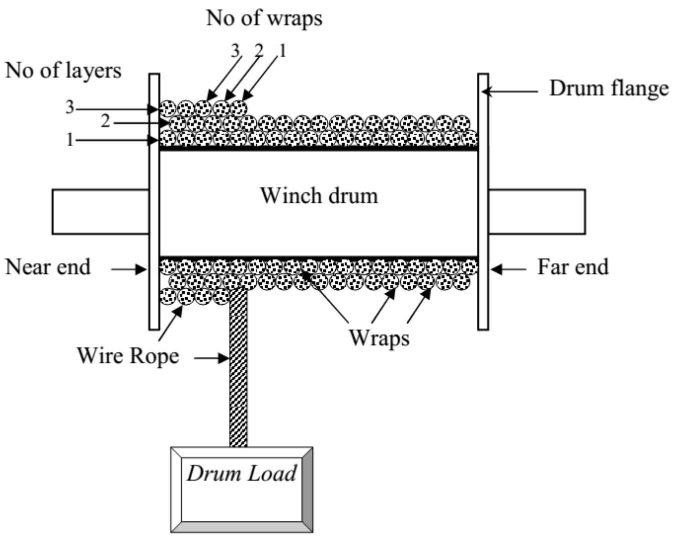


Figure 3.1 Schematic illusrtations of winch

Drum Load: maximum tension measured at the rope exit when the winch is hoisting or hauling in at the nominal speed with the rope wound on the drum.

Drum Flange: flange height shall be at least 2.5 times the rope diameter beyond the outermost layer when the rope is fully and evenly reeled onto the drum.

Wire rope:steel wire rope or fibre rope and the ratio between winch drum diameter and rope diameter shall not be less than 18 depending on the type of application.

Wraps: each turn of the rope around the full circumference of the drum is called a wrap.

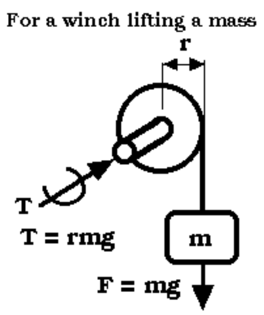
Layers: a complete number or wraps extending from flange (near end) to flange (far end) is referred to as a layer.

- **Drum**

$$T = r \cdot m \cdot g \, ; \quad \omega = v \cdot r$$

Here, T = Torque [Nm]   m = Mass [kg]   g = Gravity Acceleration [N/kg]

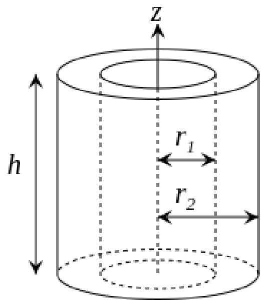
ω = Rotational Velocity [rad/s]   v = linear velocity [m/s]   r = drum diameter [m]



$$I_z = \frac{m}{2} \cdot (r_1^2 + r_2^2) = \frac{\pi \rho h}{2} \cdot (r_2^4 - r_1^4)$$

Here, I_Z =Mass Moment of Inertia [kgm^2]   m = Mass [kg] r_1 = Inner Diameter [m]

r_2 = Outer Diameter [m] ρ = Density [kg/m^3]   h = Width [m]



$$\sigma_h = C \cdot \frac{T}{d \cdot t}$$

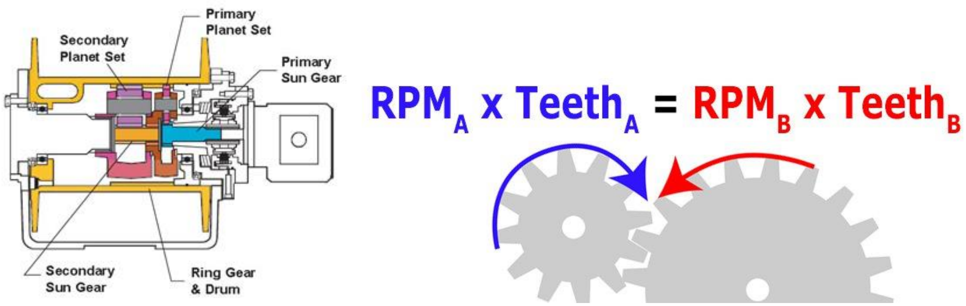
Here, σ_h = Hoop Stress [MPa]   T = Load [N]

d = Wire Diameter [mm]   t = Drum Thickness [mm]

C = 1.75 (For more than one layer)

- **Gearbox**

Generally, the drum body shall be driven by planetary gear sets. Considering the complication structure of gear sets, in this paper gearbox is simplified as a simply gear set shown below.



- **Wire**

Steel wire rope safety factor for running application or forming part of sling and for mast stays, pendants and similar standing applications shall be the greater of : Not less than the greater of 3 and but need not exceed 5.

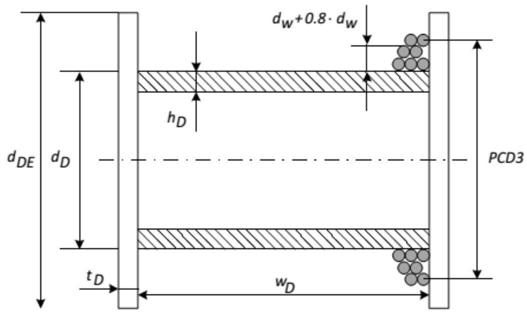
$$S_F = \frac{10^4}{0.885 \cdot SWL + 1910}$$

Here, S_F = Safety Factor [Unitless]   SWL = Safety Working Load [kN]

The minimum breaking load B of steel wire ropes shall not be less than :

$$\mathrm{B} = S_F \cdot S$$

Here, S is the maximum load in the rope resulting from the effect of the working load (suspended load).



$$\text{PCD} = d_D + \ d_w + 2 \cdot (n-1) \cdot 0.8 \cdot d_w$$

$$n_{wraps} \ = \frac{w_D}{d_w}$$

$$L_w \ = \ \pi \ \cdot PCD \ \cdot \ n_{wraps}$$

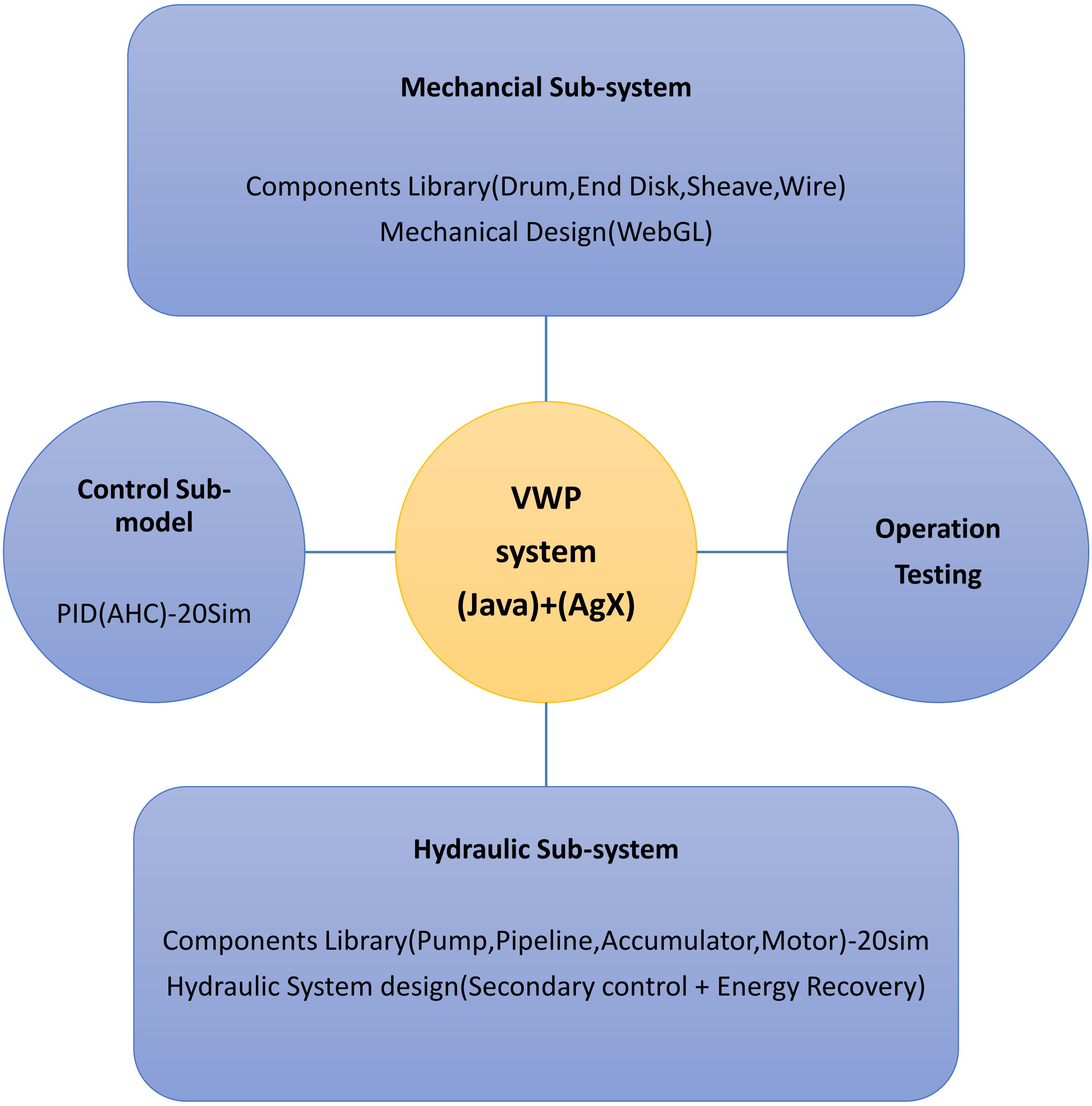
$$Total\_L_w \ = \ \sum_{i=1}^{n} L_{w\_n} = \ \ n \cdot \pi \ \cdot \ n_{wraps} \cdot (d_D + \ d_w + (n-1) \cdot 0.8 \cdot d_w)$$

Here, PCD = Pitch Circle Diameter [m], d_D = Drum Diameter [m], d_W = Wire Diameter [m] , n = layers [Unitless], n_{warps} = Warps of Layer [Unitless], w_D = Drum Width [m],

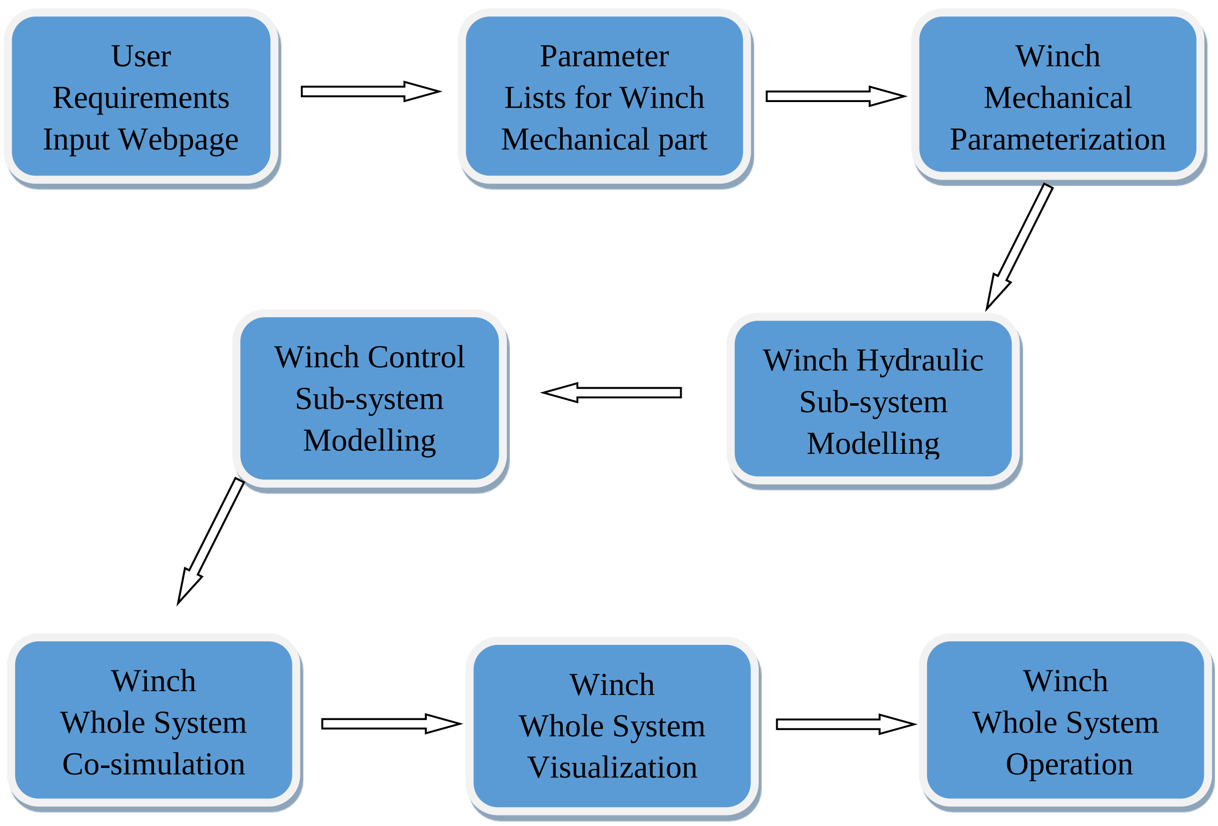
L_W = Wire Length of Layer [m], Total_ L_W = Total Wire Length on Drum [m]

3.2 Virtual Prototyping Method

As analysed in Chapter 1.2, VWP system structure can be modelled and shown as below.



Actually, there are visualization operation web page for this virtual winch prototyping system which is supported by background database. This database exactly consists of three model packages, as shown in the model structure above, which is 3D model package, hydraulic model packages and control model packages. Thus, according to the structure of system, I can obtain the flowchart of Virtual Winch Prototyping System below:


User Requirements Input Webpage
Parameter Lists for Winch Mechanical part
Winch Mechanical Parameterization
Winch Hydraulic Sub-system Modelling
Winch Control Sub-system Modelling
Winch Whole System Co-simulation
Winch Whole System Visualization
Winch Whole System Operation

4. Mechanical Sub-system Modelling

4.1 Parameterization

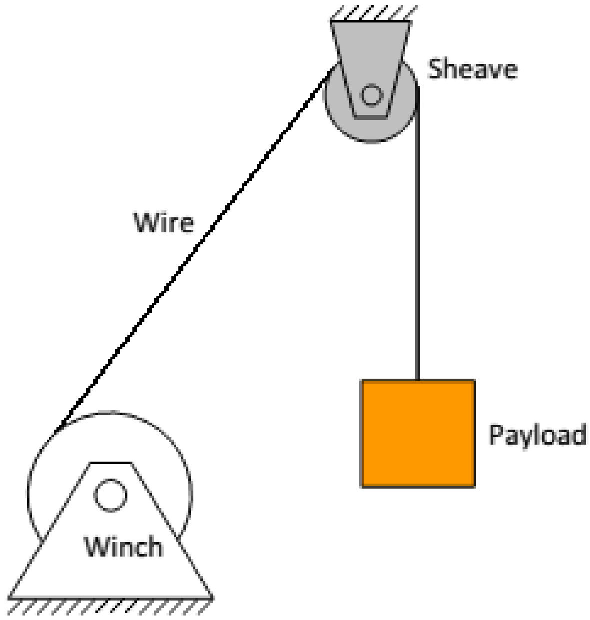


Figure 4.1 Simplification diagram of winch system and payload

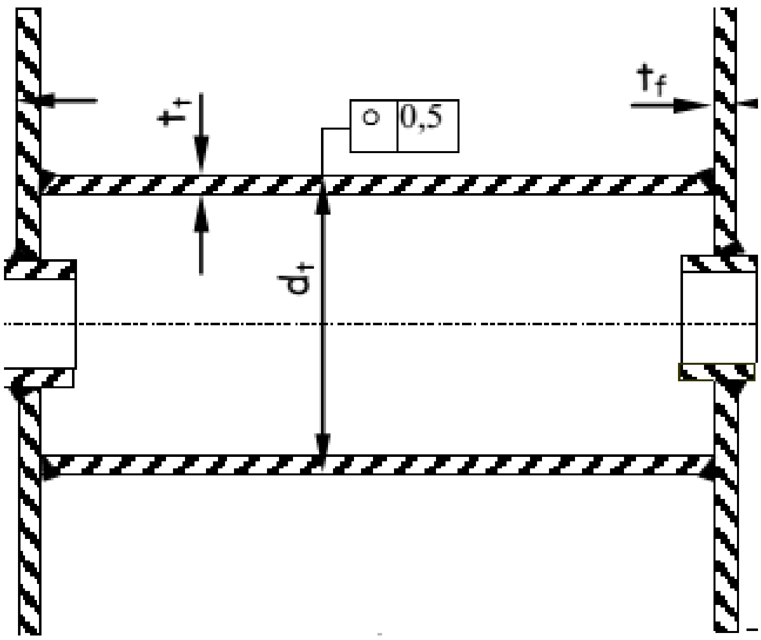


Figure 4.2 Simplification diagram of winch drum (section view)

4.1.1 Mechanical Parameter List

According to the simplification diagram shown in Figure 4.1 and 4.2, the key parameters can be summarized and shown in Table 4.1 below.

Component		**Key Parameter**
Mechanical Component Library	Drum	Diameter
		Width
		Thickness
		D/d ratio
	End Disk	Diameter
		Width
		Thickness
	Wire	Diameter
		Layer
		Length
	Planetary Gear	Gear Ratio
	Sheave	Minimum Distance

Table 4.1 Mechanical key parameter lists

4.1.2 Mechanical Parameter Dimensioning

As shown in Table 4.1, several key parameters shall be considered when dimensioning the mechanical part of the winch. After checking the parameters and DNV 2.22 Lifting Appliances, it is found that the parameters are not absolutely independent among them. Therefore, the specific relations among them shall be analysed here.

Generally, the design process starts from the end requirements, i.e. wire and payload. In this paper, take payload 10Te (SWL) for example. As mentioned in Chapter 3.1.2, wire minimum breaking force is calculated by the equation below, the minimum breaking load B of steel wire ropes shall not be less than :

$$\mathrm{B} = S_F \cdot S$$

$$S_F = \frac{10^4}{0.885 \cdot SWL + 1910}$$

Where, S_F = Safety Factor [Unitless]   SWL = Safety Working Load [kN]

Thus, minimum breaking force (MBF) = 5 * 10 * 9.81 = 490.5 kN. Then, the recommended wire diameter can be determined by checking the selection catalogue of steel wire ropes. Wire diameter for 10Te (SWL) is 25mm. Thus the winch drum diameter can be determined according to Chapter 2.2.1 in DNV 2.22 Lifting Appliances. Drum diameters shall be suitable for the selected wire rope, as directed by the rope manufacturer. The ratio D_p/d shall normally not be less than 18, where,

D_p = pitch diameter of drum

d = nominal diameter of steel wire rope.

In this paper, ratio = 25 is chosen. Then the drum recommended thickness can be determined based on the DNV design code below:

$$\sigma_h = C \frac{T}{d\,t}$$

Where,

σ_h = hoop stress in winch drum (MPa) T = load on the rope (N)

d = diameter of the rope (mm) t = thickness of winch drum (mm)

C = 1 (for 1 layer) or 1.75 (for more than 1 layer)

And according to the guidelines, the maximum permissible hoop stress σ_{pe} should not exceed 85% of the material yield stress ($\sigma_{pe} \leq 0.85\ \sigma_y$, where, σ_y = 520MPa).

Thus, $T_d \geq$ 15.5mm.

Winch drum width shall be determined considering the fleet angle shown in Figure 4.3 below. According to the DNV standard code, drum shall be designed with a width sufficient to reel up the rope in no more than 3 layers. More than 3 layers may be accepted if the wire rope has an independent wire rope core (IWRC) and one of the following conditions is complied with:

---spooling device is provided

---drum is grooved

---fleet angle is restricted to 2 °

---spilt drum is arranged

---separate traction drum is fitted

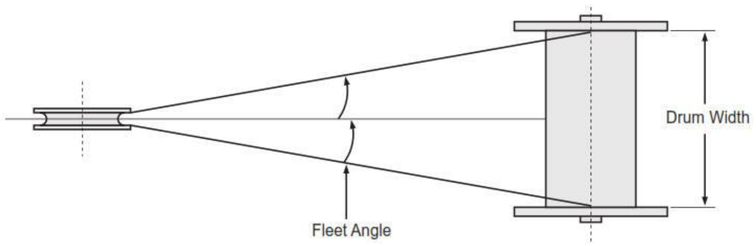


Figure 4.3 Fleet angle of winch drum

In this paper, the condition that fleet angle is restricted to 2 ° is selected. Meanwhile, minimum distance from drum shaft to the first sheave shall be acquired from the user (e.g. 8 meters), then recommended width of drum can be determined below:

$$\text{Width\_drum} = 2 * \tan(Angle_{fleet}) * \text{Distance}_{\min}$$

As minimum distance = 8 meters, the recommended drum width = 559 mm.

As for the end disk diameter, since the wire capacity of the winch is in connection with drum diameter, drum width, wire diameter, and end disk diameter, it shall be rectified based on the initial assumption after verifying the total wire length.

4.2 User Interface

4.2.1 User Input Requirements

According to the detailed analysis above, both independent and dependent parameters shall be determined before 3D parameterized modelling. Thus, a winch capacity estimator can be proposed to pre-process the user input requirements.

The estimator can be easily implemented via Excel (See Figure 4.4). The main parts includes user inputs, layers and results. User inputs consists of several key parameters of winch dimensions such as SWL, wire diameter, drum diameter, drum width, drum thickness, end disk diameter and end disk width. With key input parameters above, amount of the wraps on each layer, drum available wire capacity (available wire length) and total mass can be determined. Meanwhile, wire length on each layers can be estimated by simple calculations. According to the standard codes in Chapter 2.2.1 in DNV 2.22 Lifting Appliances, number of layers exceeds 7, special consideration and approval will be required. Thus, the maximum amount of layers is determined to be 7 in this thesis.

Winch Capacity Estimator							
User Inputs	SWL(kg)	10000		Layers	1	1st Layer Length(mm)	174343
	Wire Diameter(mm)	25			2	2nd Layer Length(mm)	177798
	Drum Diameter(mm)	984			3	3rd Layer Length(mm)	181254
	Drum Width(mm)	1396			4	4th Layer Length(mm)	184710
	Drum Thickness(mm)	19			5	5th Layer Length(mm)	188166
	End Disk Diameter(mm)	1456			6	6th Layer Length(mm)	191621
	End Disk Width(mm)	19			7	7th Layer Length(mm)	195077
					8	..	198533
Results	Wraps Each Layer	55			Verification		
	Drum Max Wire-Capacity(m)	1292.970					
	Available Wire-Capacity(m)	1257.120			General Max Layers		7
	Wire Total Mass(kg)	3892			Estimated Max Layers		11

Figure 4.4 Layout in winch capacity estimator in Excel

4.3 3D Model Implementation

In order to implement the 3D parametric model, as described in Chapter 2.2, the mesh models with kinematic and mass properties of the drum can be built and visualized by utilizing the three.js WebGL JavaScript library.

4.3.1 3D Object Meshing

Generally, the mesh models can be generated by creating vertices and faces with JavaScript from a very low level in WebGL. In this thesis, winch can be simply divided into

3 three parts, they are drum and two end disks. After checking the shape of those three parts, they all can be abstracted into a basic component demo with tube geometry.

The method to build the basic component demo is to create a set of vertices in a good order by several predefined key parameters of the tube geometry, then link three adjacent vertices to form a face which is necessary to cover the geometry. After "pushing" all the faces into the tube "geometry", a basic component demo are well built.

The following programme pseudo-code in Figure 4.5 illustrate how the tube geometry is generated in JavaScript.

```
CreateTube = function(geo)
{   //key parameters of tube
    this.innerR = 1;
    ...
    //Initial rotation and position of tube
    this.rot = [0,0,0];
    this.pos = [0,0,0];
    //Generate CreateTube function
    this.generate = function()
    {   var geometry = new THREE.Geometry();
        ...
        //Generate vertices of tube(length = 4*segments)
        for (var i=0;i<this.segments;i++)
        {   var RAD = i/this.segments;
            var XB = this.outerR * Math.cos(RAD * twopi);//outer Bottom Vertices X axis value
         ...
            geometry.vertices.push(new THREE.Vector3(XB,YB,-0.5*this.width));
         ...
        }
        //Generate closed faces of tube(length = 4*segments-8)
        for (i=0;i<this.segments-1;i++)
        {
            geometry.faces.push(new THREE.Face3(i*4,i*4+5,i*4+1));//outer face
         ...
        }
         ...
         ...
        //Define position and rotation of tube
        geometry.rotateX(this.rot[0]);
        geometry.rotateY(this.rot[1]);
        geometry.rotateZ(this.rot[2]);
        geometry.applyMatrix(new THREE.Matrix4().makeTranslation(this.pos[0],this.pos[1],this.pos[2]));

        //Merge new geometry into former geometry
        geo.merge(geometry,geometry.matrix);

        //Dispose the temporary geometry
        geometry.dispose();
    }
};
```

Figure 4.5 JavaScript program pseudo-code of generating tube geometry

Then the whole winch mesh model can be built by meshing all three geometries together (i.e. drum, 2 end disks) with material and added into the "scene". The following program pseudo-code in Figure 4.6 shows how to implement winch generation by combining three basic tube geometry in JavaScript.

```
Winch = function(mesh)
{   //key parameters of tube
    this.OuterREnd = 2.5;
    ...
    //Initial rotation and position of tube
    this.rot = [0,0,0];
    this.pos = [0,0,0];
    //Generate Drum function
    this.generate = function()
    {   mesh.geometry.dispose();
        mesh.geometry = new THREE.Geometry();
        var tube1 = new CreateTube(mesh.geometry);
        tube1.outerR = this.OuterRDrum;
        tube1.innerR = this.OuterRDrum - this.ThickDrum;
        tube1.width = this.WidthTotal -this.ThickEnd*2;
        tube1.generate();
        ...
    }
};
//Basic Winch Model Initiation
var material = new THREE.MeshPhongMaterial({color:0xFF0000,specular:0x555555,shininess:100});
var winch_mesh= new THREE.Mesh(new THREE.Geometry(),material);
winch_mesh.castShadow = true;
scene.add(winch_mesh);
var winch = new Winch(winch_mesh);
winch.generate();
```

Figure 4.6 JavaScript program pseudo-code of generating winch geometry

4.3.2 3D Object User Interaction and Visualization

User interaction can be implemented by using dat.GUI in WebGL technology. User can alter the key parameters of the winch in control box shown in the Figure 4.7 below and obtain the meshing object with required dimensions.

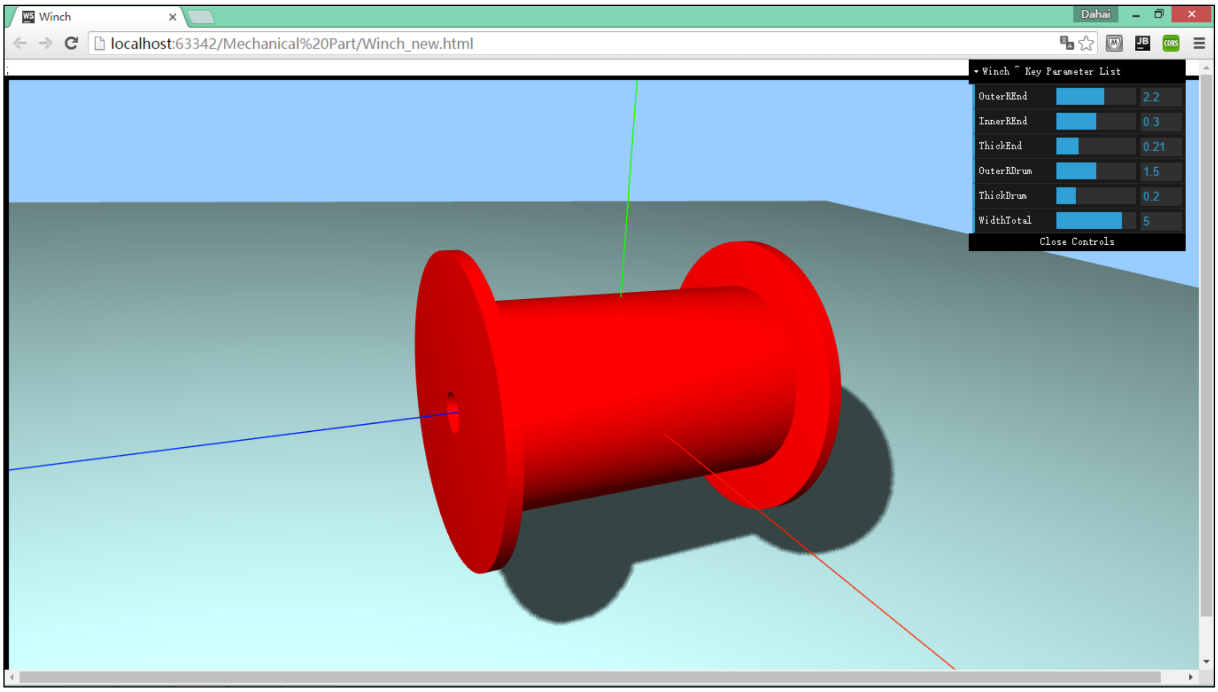


Figure 4.7 Web page of winch 3D model interaction and visualization

As for the 3D model visualization (Animation), "three.js" can creates a HTML canvas element, on which the projections of a scene are drawn. Thus, a 3D parametric winch model can be interfaced and visualized in web-browser for data exchange and information sharing. The generated models can be wrapped, saved as .OBJ-formatted objects and directly used for simulation in the virtual prototyping framework.

5. Hydraulic and Control (AHC) Sub-model Modelling

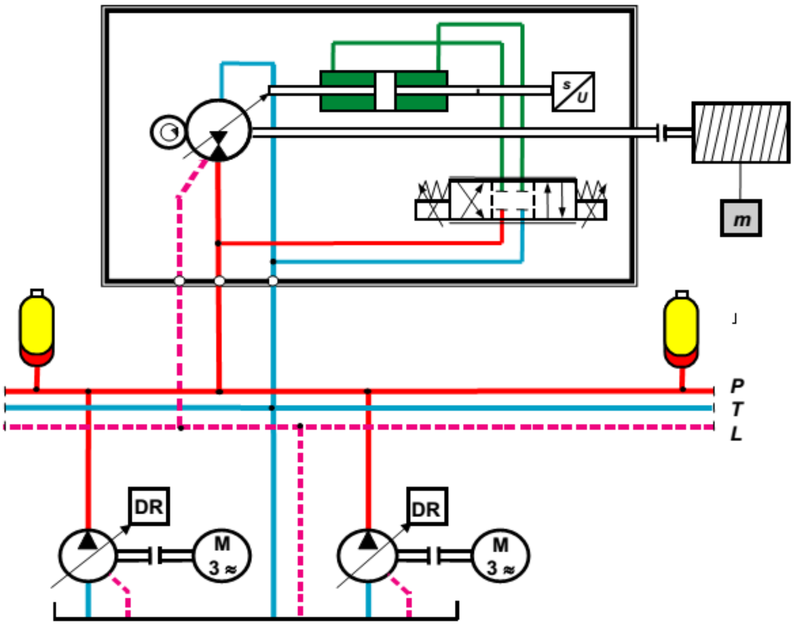


Figure 5.1 Typical schematics of secondary control in winch system

Before modelling the hydraulic winch system, the energy flow (Figure 5.2) shall be analyzed. As we can see, total energy transmission efficiency mainly includes the volumetric and mechanical-hydraulic efficiency of the secondary unit, efficiency of transmission, travel resistance and pressure losses.

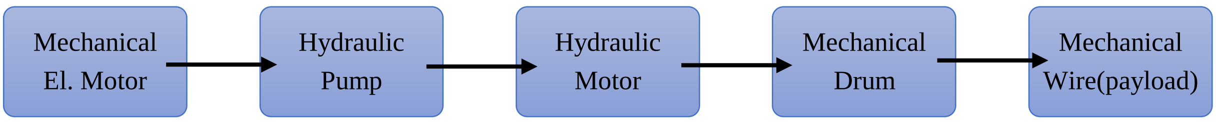


Figure 5.2 Energy flow of hydraulic winch system

5.1 Hydraulic Components Library Modelling

In drive technology two power transmitting parameters are of importance: Torque (Nm) and Speed (rpm). These mechanical parameters correspond to the following parameters in hydrostatic drives: Pressure (Bar) and Flow (L/min). Depending on the coupling of the mechanical and hydraulic parameters, we differentiate between two types of drives. They are: drive system with flow coupling (conventional systems) and drive system with pressure coupling (secondary control).

Conventionally, a common way to control the winch movement is to use a normal transmission solution where the main pump is doing all the work (i.e. primary controlled system). The principle can be seen in Figure 5.3 on the right. In order to move the winch drum back and forth, the pump will have to build up pressure on each sides of the connected pipelines, creating torque in both directions. Since this forces the pump to work against the "hydraulic spring" every time a change in rotational speed or required torque, that is oil column between the control element and actuator will be either compressed or expanded, resulting in a significant delay in the efforts to control the winch drum motion stably. Furthermore, controlling on the primary side also has the big drawback that to rotate the winch drum a certain angle, a fixed amount of flow is always needed, which causes over dimension of the pumps if high speed at low load is wanted.

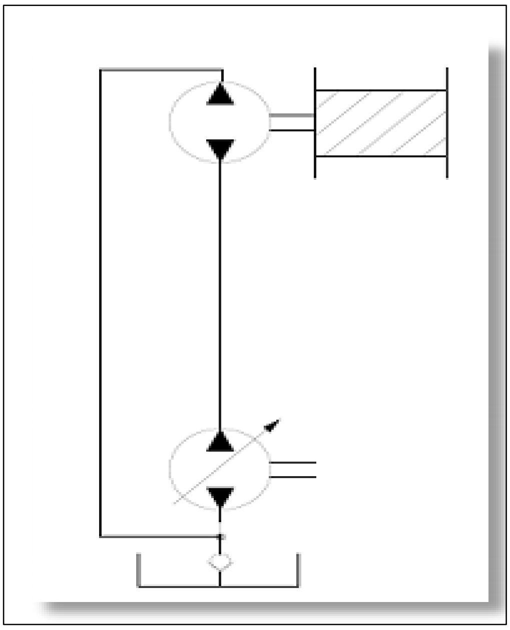


Figure 5.3 Primary control winch system

Secondary control, a hydrostatic drive concept in a hydraulic system with impressed operating pressure, has been in worldwide use since 1980. It is used predominantly where a conventional drive is no longer able to fulfill the requirements in terms of dynamic response, positioning and precision control of speed and torque, which is becoming increasingly demanded in maritime industry especially in motion compensation field (i.e. AHC). A typical hydraulic system with secondary control schematics and specific structure is shown in Figure 5.4 and 5.5 below. This type of transmission design differs from conventional ones in that the system pressure is dependent on the loading condition of the accumulator and can no longer be freely matched to the required output torque. Such hydraulic drive system are known as hydrostatic transmissions with "impressed" operating pressure, a term taken from electro-technology. Output drive torque M_{d2} is determined by varying displacement V_{g2} of secondary unit via changing the swivel angle of swash plate inside the secondary unit.

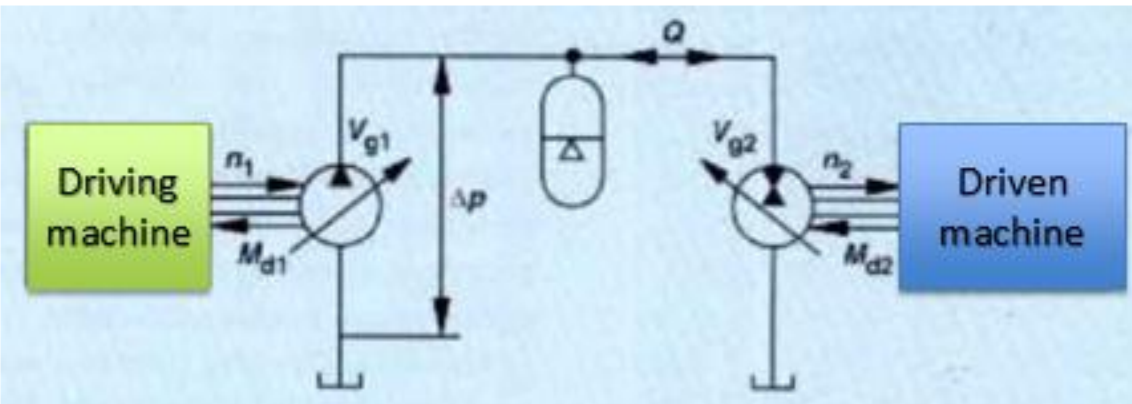


Figure 5.4 Typical secondary control system schematics

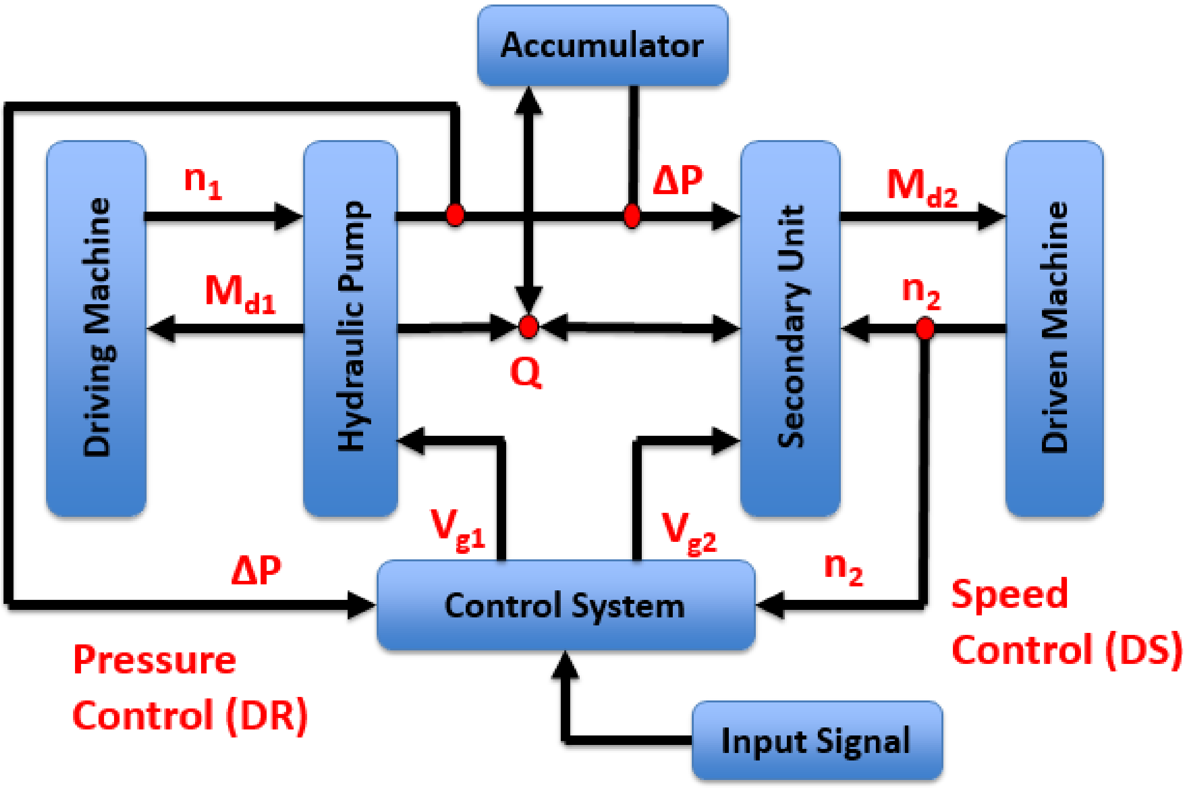


Figure 5.5 Typical secondary control specific structure

Similar to traditional hydrostatic transmission, a secondary controlled system can be used in either open or closed circuits. The main difference is where the pump suction comes from. In an open circuit, the reservoir needs to be either pressurized or mounted higher than the pump so as to avoid cavitation. Meanwhile, a so-called pre-fill operation where low pressure side is pressurized with a boost pump can also be arranged in open circuits. The closed circuit transmission has an architecture that low pressure side will be directly connected the secondary unit. Thus, the same amount of oil will always circulate in the circuits back and forth. Generally, in order to keep the balance of oil volume in the circuit, closed circuit shall not only be equipped with hydraulic accumulator on the high pressure side but also be equipped with another hydraulic accumulator on the low pressure side. Meanwhile, lack of hydraulic oil filtering and oil cooling in closed circuit may pose huge threats to system normal operation. Furthermore, onboard hydraulic power source is usually arranged with a central hydraulic power station, closed circuit is obviously prone to bring hydraulic oil pollutions in the central station. Considering about those, in this thesis, open circuit is employed.

According to the analysis above, in this thesis, hydraulic winch system can be mainly divided into several components with key parameters shown in Table 5.1 below.

Component		Key Parameter
Hydraulic Component Library	Pump(HPS)	Displacement
		rpm
		Pressure
		Flow
	Motor(SCDU)	Displacement
		rpm
		Pressure
		Flow
	Pipe Line	Diameter
		Length
	Accumulator	Charging Pressure
		Mechanical Volume
		Gas Volume

Table 5.1 Hydraulic key parameters of winch system

5.1.1 Hydraulic Power Supply

According to bond graph theory, generally power source (i.e. Pump) can be modeled as flow source (Sf). In this thesis, since "impressed" operating pressure shall be maintained by hydraulic power supply model, pressure compensated pump is required. Pressure compensated pump can be an axial piston pump (see Figure 5.6). The Rexroth A4VSO pump (see Figure 5.7) is a swash plate pump suitable for this concept. Thus, the pump can be modeled referring to the operation characteristic of A4VSO.

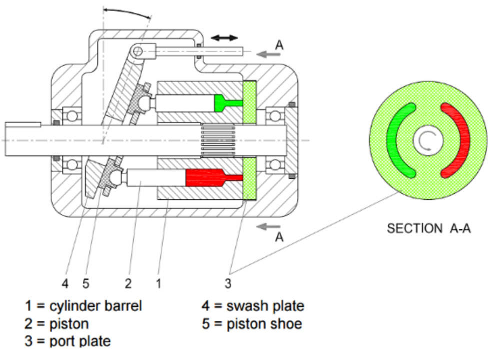


Figure 5.6 Variable displacement axial piston pump section view (left)

Figure 5.7 Rexroth A4VSO with pressure controller pump (right)

Compared with axial piston pump with fixed swivel angle of swash plate, variable displacement of pump can be achieved, resulting in variable flow of pump, which is the key precondition of attaining the pseudo-constant pressure. Thus, the dimensioning of the pump can be calculated by the equations in Figure 5.8.

Calculating Pump Size		
	Fixed displacement swashplate pump	**Variable displacement swashplate pump**
Flow	$Q = \frac{V_g \cdot n \cdot \eta_{vol}}{1000}$ (l/min)	$Q = \frac{V_{g\,max} \cdot n \cdot \tan\alpha \cdot \eta_{vol}}{1000 \cdot \tan\alpha_{max}}$ (l/min)
Drive speed	$n = \frac{Q \cdot 1000}{V_g \cdot \eta_{vol}}$ (rpm)	$n = \frac{Q \cdot 1000 \cdot \tan\alpha_{max}}{V_{g\,max} \cdot \eta_{vol} \cdot \tan\alpha}$ (rpm)
Drive torque	$M = \frac{V_g \cdot \Delta p}{20\,\pi \cdot \eta_{mh}} = \frac{1{,}59 \cdot V_g \cdot \Delta p}{100 \cdot \eta_{mh}}$ (Nm)	$M = \frac{V_{g\,max} \cdot \Delta p \cdot \tan\alpha}{20\,\pi \cdot \eta_{mh} \cdot \tan\alpha_{max}} = \frac{1{,}59 \cdot V_{g\,max} \cdot \Delta p \cdot \tan\alpha}{100 \cdot \eta_{mh} \cdot \tan\alpha_{max}}$ (Nm)
Drive power	$P = \frac{2\,\pi \cdot M \cdot n}{60\,000} = \frac{M \cdot n}{9549}$ (kW) $P = \frac{Q \cdot \Delta p}{600 \cdot \eta_{vol} \cdot \eta_{mh}} = \frac{Q \cdot \Delta p}{600 \cdot \eta_t}$ (kW)	$P = \frac{2\,\pi \cdot M \cdot n}{60\,000} = \frac{M \cdot n}{9549}$ (kW) $P = \frac{Q \cdot \Delta p}{600 \cdot \eta_{vol} \cdot \eta_{mh}} = \frac{Q \cdot \Delta p}{600 \cdot \eta_t}$ (kW)

Where:

Q	=	flow	(l/min)
M	=	drive torque	(Nm)
P	=	drive power	(kW)
V_g	=	geometric displacement per rev.	(cm³)
$V_{g\,max}$	=	max. geometric displacement per rev.	(cm³)
n	=	speed	(rpm)

α_{max}	=	max. swivel angle (varies according to design)
α	=	set swivel angle (between 0 and α_{max})
η_{vol}	=	volumetric efficiency
η_{mh}	=	mechanical - hydraulic efficiency
η_t	=	overall efficiency ($\eta_t = \eta_{vol} \cdot \eta_{mh}$)
Δp	=	differential pressure (bar)

Figure 5.8 Data sheet of variables displacement axial piston pump

As can be seen from Figure 5.8, flow of pump is proportional to the displacement of pump which is proportional to tan (α), α is called the swivel angle between swash plate and vertical axis.

Then a pressure controller (DR) shall be used together with A4VSO pump so as to maintain the constant operating pressure. The DR schematics is shown in Figure 5.9 below.

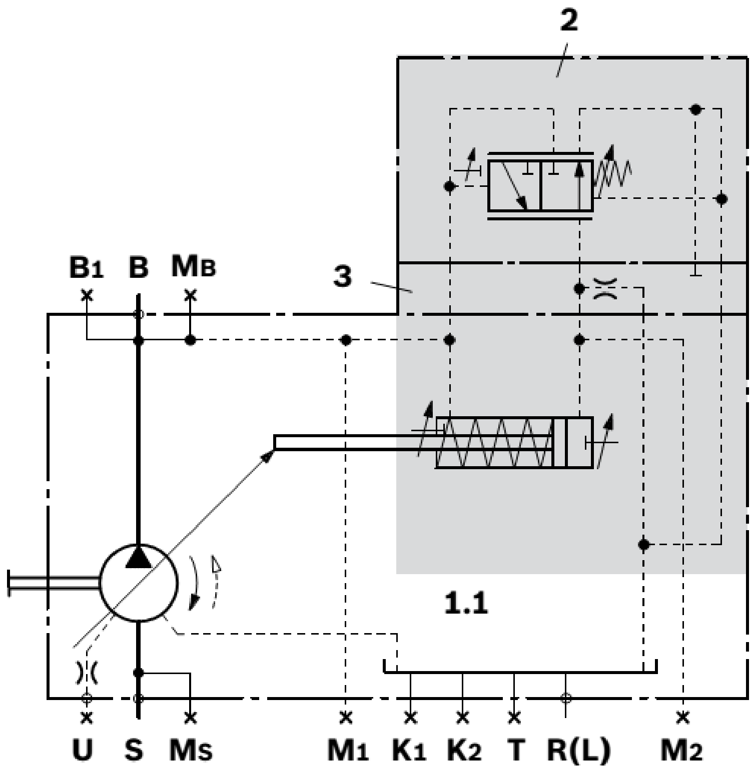


Figure 5.9 A4VSO pressure controller (DR) schematics

In Figure 5.9, Section 1.1 is A4VSO, Section 2 is attached pressure control valve (sequence valve) and Section 3 is intermediate plate. The controller limits the maximum pressure at the pump outlet (Port B) within the control range of the variable pump. The variable pump only moves as much hydraulic fluid as is required by the consumers. If the operating pressure exceeds the pressure setting at the pressure control valve, the pump will regulate to a smaller displacement to reduce the control differential.

As analyzed above, A4VSO with DR is the pressure compensated pump that shall be modelled in 20Sim. However, this pump model becomes too complicated if we model all the component details. Proper simplification is necessary. After checking the information in Figure 5.8 and 5.9, it is found that swivel angle of swash plate has something to do with spring elongation inside the control cylinder. It can be expressed as the equation below.

$$\frac{\tan(\alpha)}{\tan(\alpha)\_max} = \frac{\Delta x}{\Delta x\_max}$$

Thus, $Q_{pump} = Q_{max} * gain$ (gain is between 0 and 1). That gain value is related to the spring characteristics in the control cylinder. And as for the pressure control valve, it can be simplified as "if and else" logic clauses in 20Sim to set the gain value for the pump flow. Obviously, this pump model seems a little ideal that the hydraulic actuators devices (i.e. pressure control valve and control cylinder with swash plate) attached on the pump which is used to alter pump displacement are not detailed modelled. However, that is already enough to roughly reflect the axial piston pump behaviors and characteristics in a whole.

5.1.2 Pipelines and Reservoir

Generally there are three types of pipelines, they are pressure line, return line, draining line. Fluid velocity differs among them. As hydraulic oil flow through pipelines, hydro dynamics such as fluid inertia, friction, fluid flexibility (compression and extension) can pose effects on the hydraulic energy transmission. To simplify the pipeline model in 20Sim, spring-damping model in kinematics can be brought into hydraulic transmission model of pipelines. Meanwhile, fluid flexibility and fluid inertia in draining line are not considered. (See 20Sim model Figure 5.10)

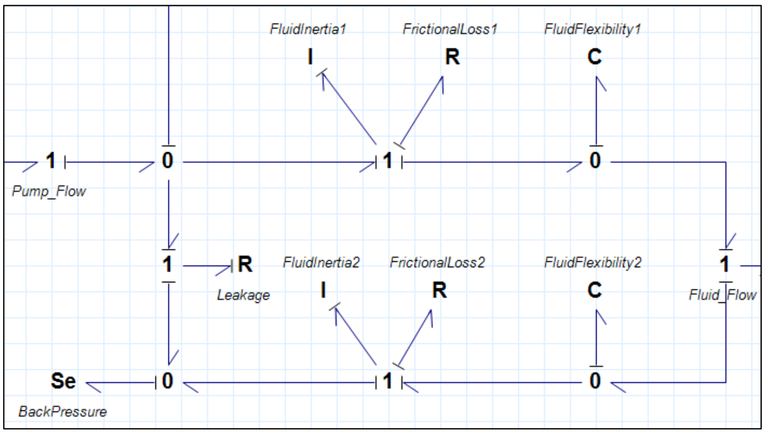


Figure 5.10 Hydraulic pipelines and reservoir bond graph model in 20Sim

It can be seen in Figure 5.10, I, R and C elements represent fluid inertia, frictional loss and fluid flexibility (pipe segment) respectively in both pressure line and return line. As a matter of fact, this pipeline can be divided into several small segments. And with regard to reservoir, Se element is used.

5.1.3 Secondary Control Drive Unit (SCDU)

As an actuator, hydraulic motor convert hydraulic energy into mechanical energy (rotational). Similarly in hydraulic pump, the Rexroth A4VSO (Figure 5.11) is also a suitable secondary unit which meets the requirements of secondary control in this thesis. The difference is the pressure controller (DR) is replaced by speed controller (DS). SCDU dimensioning can be calculated by the equations in Figure 5.12.

Figure 5.11 Bosch Rexroth A4VSO with DS (SCDU)

Calculating Motor Size

	Fixed displacement swashplate motor	Variable displacement swashplate motor
Consumption (Flow)	$Q = \frac{V_g \cdot n}{1000 \cdot \eta_{vol}}$ (l/min)	$Q = \frac{V_{g\,max} \cdot n \cdot \tan\alpha}{1000 \cdot \eta_{vol} \cdot \tan\alpha_{max}}$ (l/min)
Output speed	$n = \frac{Q \cdot 1000 \cdot \eta_{vol}}{V_g}$ (rpm)	$n = \frac{Q \cdot 1000 \cdot \eta_{vol} \cdot \tan\alpha_{max}}{V_{g\,max} \cdot \tan\alpha}$ (rpm)
Output torque	$M = \frac{V_g \cdot \Delta p \cdot \eta_{mh}}{20\pi} = \frac{1{,}59 \cdot V_g \cdot \Delta p \cdot \eta_{mh}}{100}$ (Nm)	$M = \frac{V_{g\,max} \cdot \Delta p \cdot \eta_{mh} \cdot \tan\alpha}{20\pi \cdot \tan\alpha_{max}} = \frac{1{,}59 \cdot V_{g\,max} \cdot \Delta p \cdot \eta_{mh} \cdot \tan\alpha}{100 \cdot \tan\alpha_{max}}$ (Nm)
Output power	$P = \frac{2\pi \cdot M \cdot n}{60\,000} = \frac{M \cdot n}{9549}$ (kW) $P = \frac{Q \cdot \Delta p}{600} \cdot \eta_{vol} \cdot \eta_{mh} = \frac{Q \cdot \Delta p \cdot \eta_t}{600}$ (kW)	$P = \frac{2\pi \cdot M \cdot n}{60\,000} = \frac{M \cdot n}{9549}$ (kW) $P = \frac{Q \cdot \Delta p}{600} \cdot \eta_{vol} \cdot \eta_{mh} = \frac{Q \cdot \Delta p \cdot \eta_t}{600}$ (kW)

Where:

Q	=	consumption (flow)	(l/min)
M	=	output torque	(Nm)
P	=	output power	(kW)
V_g	=	geometric displacement per rev.	(cm³)
$V_{g\,max}$	=	max. geom. displacement per rev.	(cm³)
n	=	speed	(rpm)

α_{max}	=	max. swivel angle (varies according to design)
α	=	set swivel angle (between 0 and α_{max})
η_{vol}	=	volumetric efficiency
η_{mh}	=	mechanical - hydraulic efficiency
η_t	=	overall efficiency ($\eta_t = \eta_{vol} \cdot \eta_{mh}$)
Δp	=	differential pressure (bar)

Figure 5.12 Data sheet of variables displacement axial piston motor (SCDU)

It can be illustrated in Figure 5.12, flow of hydraulic motor (SCDU) is proportional to the displacement of motor, that is, proportional to tan (α). Meanwhile, output torque is also proportional to the displacement of motor under the constant differential pressure, which means that speed control of motor can be achieved by balancing the required torque and output torque via changing the swivel angle of swash plate. (See 20Sim model of SCDU in Figure 5.13)

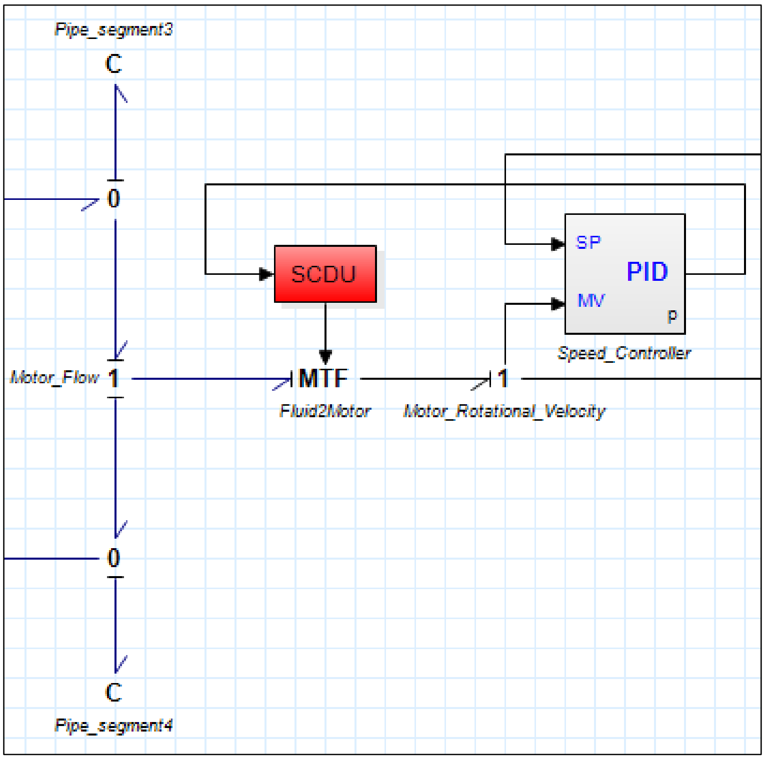


Figure 5.13 Hydraulic motor (SCDU) bond graph model in 20Sim

Different to the pump modelling in Chapter 5.1.1, hydraulic actuator devices in SCDU (i.e. servo valve, control cylinder and swash plate) shall be modeled considering the complexity of control method in closed loop speed control. Figure 5.14 illustrates the basic principle of swivel angle altering in secondary control.

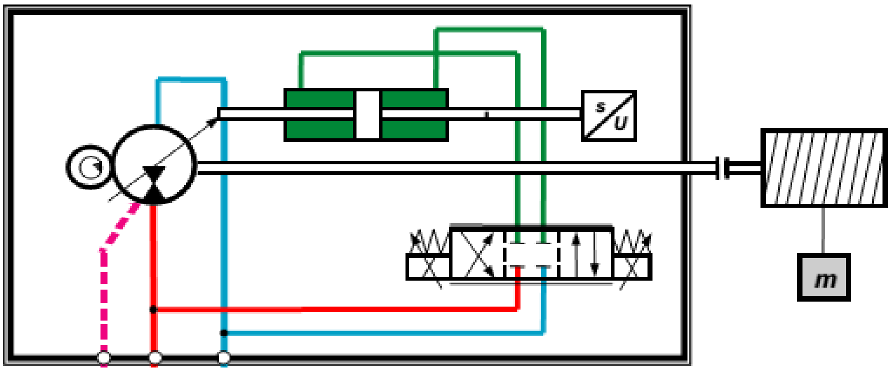


Figure 5.14 Schematics of swivel angle altering in secondary control drive unit

As we can see from the diagram inside the black frame above, swivel angle of the swash plate is changed by the cooperation between servo valve (or directional valve) and control cylinder, which means that we can control the swivel angle via controlling the output stage of servo valve. In this paper, the high response 4/4 way servo solenoid directional control valve-4WRPH from Rexroth can be used. (See 20Sim model in Figure 5.15).

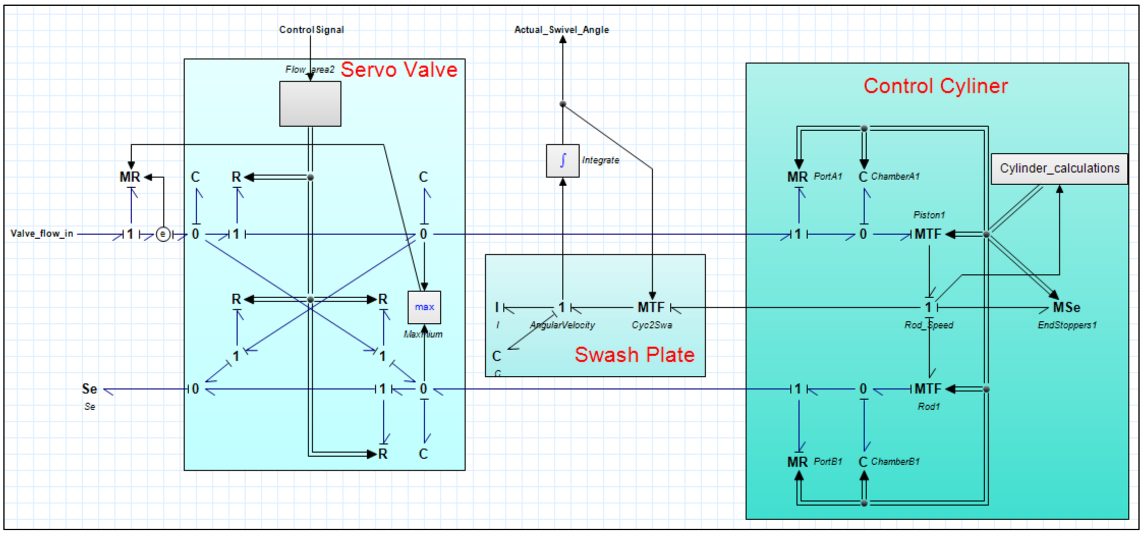


Figure 5.15 Secondary control drive unit bond graph model in 20sim

5.1.4 Accumulators

As mentioned previously in Chapter 5.1, the conventional drive system (flow coupling) reacts to a change in torque with a change in operating pressure differential. By contrast, the secondary controlled drive system (pressure coupling) reacts to a change in torque with a change in flow requirements i.e. as with an electric motor a high loading will result in a large current. However, in order to cover a sudden peak flow requirement, a hydraulic accumulator is usually required. This accumulator also can be used to absorb potential energy produced from lowering a load with the winch. Meanwhile, due to the reciprocal influences of speed and pressure control secondary controlled drive systems without hydraulic accumulators are difficult to stabilize.

With the hydraulic accumulator used, energy is by means of compression and expansion of the volume of nitrogen. System pressure is therefore closely related to the relevant loading condition of the accumulator. There are three types of accumulators generally, they are diaphragm, bladder and piston accumulators. For the former two types of accumulators, they are both characterized by features of quick response, not large volume and long service time compared with piston accumulators. However, considering the large volume requirements of energy recovery, in this paper, it is better to use piston accumulators with gas bank stations.

With regard to the accumulator dimensioning, since the power requirements of the secondary unit at an impressed pressure is matched only by varying the flow, it is quite difficult to dimension the volume by means of speed and displacement. Therefore, it is better to use the equation of energy as follows:

$$E_{stor} = E_{mech} * \eta_{total}$$

Where, η_{total} is the total efficiency of the whole drive path, mainly including the volumetric, mechanical-hydraulic, transmission, travel resistance and pressure losses. E_{stor} represents the energy to be stored in accumulator and E_{mech} represents the type of storage energy. In this winch system, potential energy is the main type of storage energy. Thus, E_{mech} = mgh.

With regard to the calculation of accumulator volume for a secondary controlled winch drive, the equations below can be used.

$$V_1 = \frac{E_{stor}(1-n)}{P_1 \cdot 10^2 \left[1 - \left(\frac{P_1}{P_2}\right)^{\frac{1-n}{n}}\right]}$$

$V_0 = \frac{V_1}{0.85}$ ($P_0 = P_1 - 5$ Bar for piston accumulators)

$$V_2 = V_1 \cdot \left(\frac{P_1}{P_2}\right)^{\frac{1}{n}}$$

Where,

E_{stor} = energy to be stored in Nm.

P_1 = minimum accumulator pressure in Bar.

P_2 = maximum accumulator pressure in Bar.

V_0 = volume of gas in L.

V_1 = accumulator volume at P_1 in L.

V_2 = accumulator volume at P_2 in L.

n = polytrophic exponent.

As for the polytrophic exponent n for nitrogen, the value it is shown in the table below.

Operating Pressure		Bar				
N_2 temperature		50	100	200	300	400
^{o}C	K	n values				
-23	250	1.540	1.677	1.835	1.842	1.801
27	300	1.485	1.563	1.669	1.707	1.705
77	350	1.465	1.507	1.581	1.618	1.629

Table 5.2 Polytrophic exponent n for nitrogen

Hence, based on the calculation formulas and analysis, the gas bank station capacity can be determined.

5.1.5 Check valve and pressure relief valve

All the hydraulic valves can be expressed as a mathematical model shown below:

$$Q = C_d A \sqrt{\frac{2\Delta P}{\rho}}$$

Check valve, also called non-return valves, are used to stop flow in one direction and allow unrestricted flow in the other. In this thesis, the check valve is used to prevent reverse flow coming from secondary control unit when lowering the payload. Meanwhile, when lowering the payload reverse flow can be stored in the accumulator for energy recovery. However, the large amount of reverse flow cannot be all restored into the gas tank, therefore pressure relief valve is used to keep the operating pressure within the control range in the overall system.

5.2 Control (AHC) Sub-model Modelling

As for the control method for this hydraulic system, as analyzed in Chapter 5.1, hydraulic motor speed and operating pressure shall be controlled properly in this system.

5.2.1 Speed Control Circuit

Unlike conventional hydrostatic drives, speed control of motor is implemented by continually varying swivel angle of the swash plate. Therefore, the positional control of swivel angle may be a subordinate circuit of the whole speed control loop. That is speed and position double feedback control. However, the simplification principle, in this thesis, only single speed loop is employed. The structure of double feedback control circuit is shown in Figure 5.16 below.

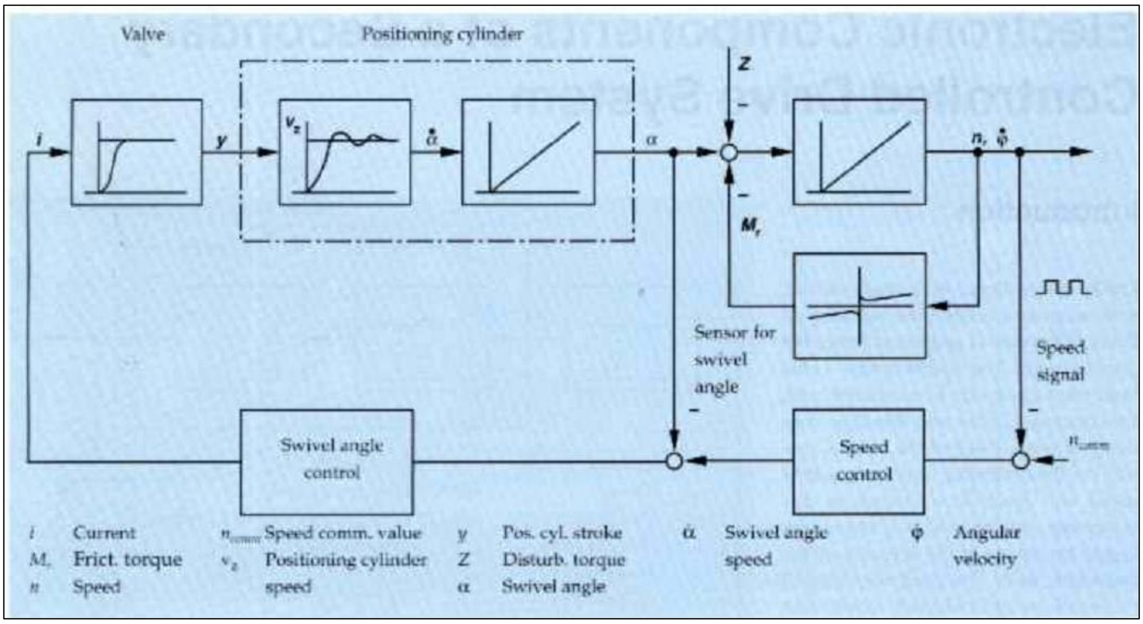


Figure 5.16 Control circuits of closed loop speed control in SCDU

5.2.2 AHC Model

As for the AHC model, as a matter of fact, this part has almost completed in speed control part. There is only one difference is that the reference signal is not sent from joystick or control panel, but from the MRU which can provide the wave prediction model of the vessel or platform. Thus, the speed controller can guarantee the motion of wire or cable end tip follow the predicted reverse movement of the wave or vessel, i.e. compensate the movement imposed on the vessel.

5.3 Mechanical Sub-model Modelling

This mechanical sub-model modelling aims at testing the hydraulic models and make comparisons between Bond Graph simulation and co-simulation in the final virtual winch prototyping framework. This parts consists of gearbox, winch main body, wire and payload.

5.3.1 Gearbox

As mentioned in Chapter 3.1.2, gearbox can be modeled as a transformer considering the relation of torque and rotational speed between the input and output sides of gearbox. Thus, in bond graph model, a TF element can be used.

5.3.2 Winch Main Body

As a whole, for simplified principle, the winch main body can be regarded as a drum with two end disks, the support part of the winch will not be considered. Main physical properties of the drum with two end disk are mass inertia and transformer function, the friction can also be considered. Therefore, the drum can be modeled as transformer with I (inertia) element. However, one thing shall be noted that the transformer has variable transforming ratio, for the wire continued wind on or off the drum resulting in the drum diameter changes.

5.3.3 Wire

The wire model seems a little complicated since the wire has both stiffness and flexibility properties. In order to balance the necessary physical property details and simplification principle, wire can be modeled as elasticity (C element) with damping (R element).

5.3.4 Payload

The payload can be easily modeled as an effort resource with mass inertia.

6. Virtual Winch Prototyping Framework Integration

After developing the models of mechanical sub-system and hydraulic with control sub-system by independent tools, a good integration framework shall be properly implemented.

6.1 Overall integration structure

Since sub-system components developed in independent tools, it poses big challenge to integrate domain-specific models so as to implement system co-simulation. As mentioned in Chapter 1.2, with the help of FMI, sub-models from different disciplines can be developed in "stand-alone" software tools and then be integrated in a separate integration platform. A Virtual Prototyping Framework can be developed by using Java to import FMUs for co-simulation. The struture is shown in Figure 6.1 below.

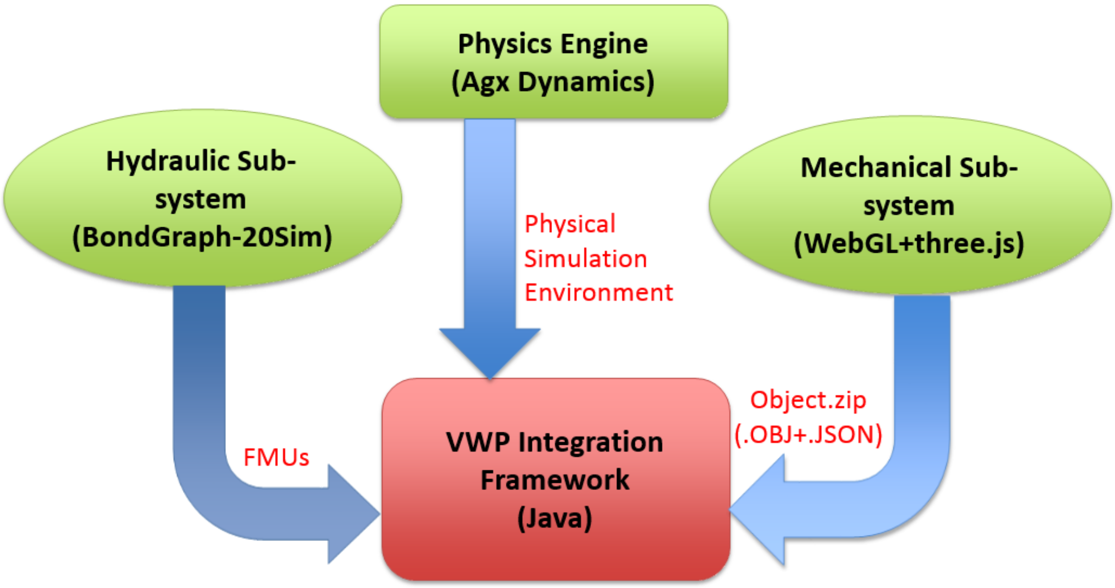


Figure 6.1 VWP overall integration structure

As we can see from Figure 6.1, hydraulic with control model developed in 20Sim can be exported as FMUs and integrated into the framework supported by Java platform. And mechanical model can be exported with a ZIP formatted file which consists of two files with OBJ and JSON formatted. OBJ formatted file contains the information of the shape or geometry and JSON formatted file contains the information of the key parameters' dimensions of the 3D model.

6.2 Wire model

In mechanical 3D modelling, the wire modelled is ignored. But this part can be modelled in AgX dynamics considering that wire simulations in AgX Dynamics are fast enough for demanding real time simulators and yet the models have high physical accuracy and are stable under extreme loads and mass ratios.

In AgX, the namespace agxWire contains numerous classes for simulation wires, ropes and chains. These classes can be used to simulate cranes, winches etc. The wire is an implementation of a lumped element structure with dynamic resolution.

The wire is placed into the world by a routing process in AgX. During the routing, nodes of various types are created and positioned attached to rigid bodies. The nodes are then added to the wire. A typical wire model is shown in Figure 6.2 below in AgX.

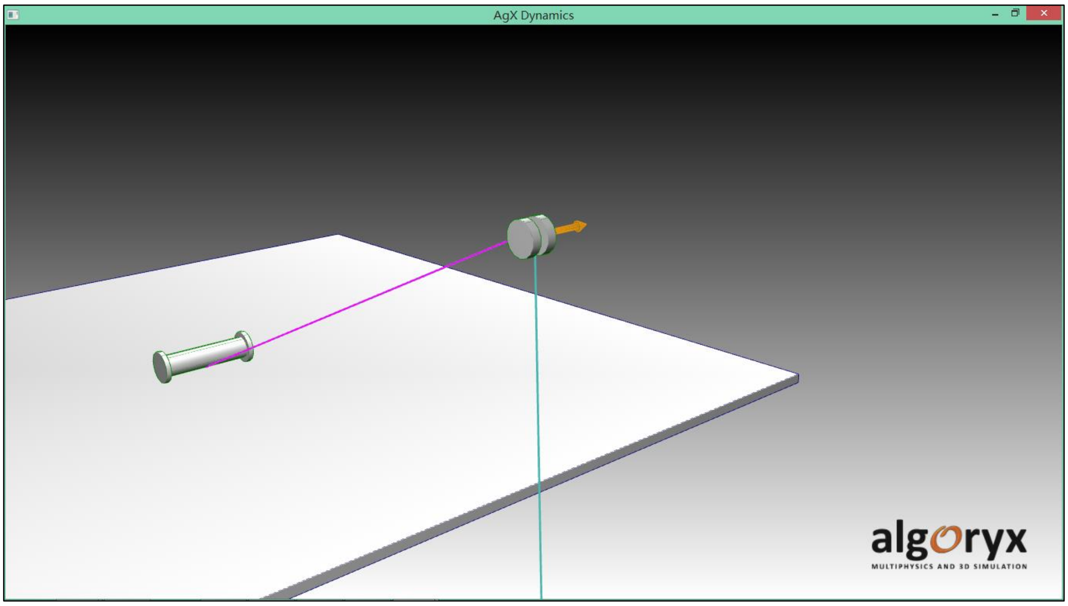


Figure 6.2 Wire model in AgX

6.3 Virtual prototyping framework

This framework is entirely implemented in Java. Since independent modelling tools are used to develop the overall system (hydraulic and mechanical), a proper system integration method shall be developed to integrate the domain-specific system models. The key is mainly focused on interfaces or data sharing of models. The specific software architecture is shown in Figure 6.3 below.

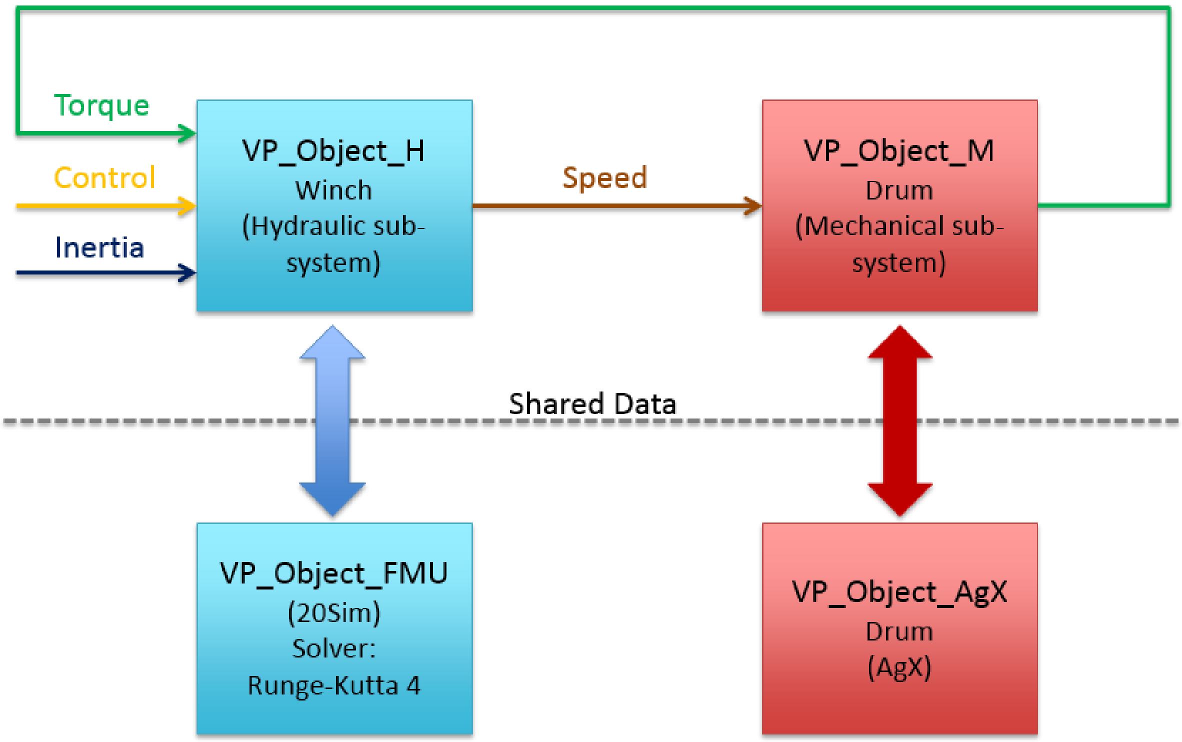


Figure 6.3 Virtual prototyping framework software architecture

To be more specific, the upper two virtual prototyping objects consists of main physical properties of winch system aiming at implementing co-simulation which contains data processing, testing and visualization of entire system. The lower two objects are FMU formatted hydraulic sub-system and physics engine provided by AgX. The data shall be shared between the upper and lower objects.

7. Results and discussion

General specification table:

SWL	10/15/20/30/40/50/75/100 T
AHC capacity	wave amplitude = 2m wave period = 10s
Hoisting speed	1.257 m/s
Working depth	800m

7.1 Case study (SWL = 10T)

7.1.1 Mechanical sub-system results

Assume the safety working load (SWL = 10T), then we need to choose the proper pre-defined SWL value in the pull-down menu of the winch capacity estimator (see Figure 7.1). Actually, once the SWL value is chosen, the recommended wire diameter is matched and shown below the SWL item. Then we can insert the value of drum diameter, drum width, drum thickness and end disk diameter. The value has pre-defined minimum and/or maximum value. If the inserted value is not proper, excel gives warning and hint information for user to follow (see Figure 7.2).

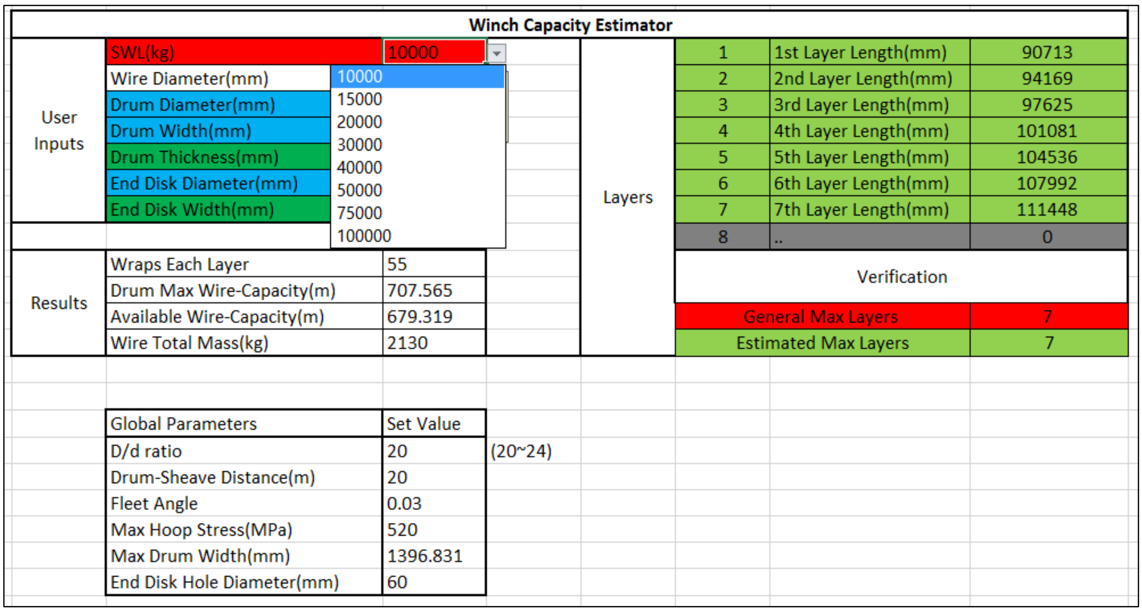


Figure 7.1 SWL = 10T Winch Capacity Estimator-a

Winch Capacity Estimator

User Inputs	SWL(kg)	10000	1	1st Layer Length(mm)	48553
	Wire Diameter(mm)	25	2	2nd Layer Length(mm)	52009
	Drum Diameter(mm)	256	3	3rd Layer Length(mm)	55465
	Drum Width(mm)	1396	4	4th Layer Length(mm)	58921
	Drum Thickness(mm)	16	5	5th Layer Length(mm)	62376
	End Disk Diameter(mm)	820	6	6th Layer Length(mm)	65832
	End Disk Width(mm)	16	7	7th Layer Length(mm)	69288
			8	..	72744
Results	Wraps Each Layer	55			
	Drum Max Wire-Capacity(m)	412.444			7
	Available Wire-Capacity(m)	388.030			13
	Wire Total Mass(kg)	1241			

Global Parameters	Set Value	
D/d ratio	20	(20~24)
Drum-Sheave Distance(m)	20	
Fleet Angle	0.03	
Max Hoop Stress(MPa)	520	
Max Drum Width(mm)	1396.831	
End Disk Hole Diameter(mm)	60	

Drum Diameter
Wire Diameter *
D/d ratio

Layers

Microsoft Excel
Value is not proper,check input information
重试(R) 取消 帮助(H)

Figure 7.2 SWL = 10T Winch Capacity Estimator-b

Once all the proper parameter inputs are inserted, wire length of each layers, wire total length and wire total mass can be calculated by the pre-set functions inside excel. Among them, there is a verification part in which the maximum amount of layers can be estimated according to the dimensions of key parameters. Generally, in DNV standard code, layers winded on the drum may be no more than 7 layers. This part can provide the necessary information for layer amount verification. After the verification, the pre-process of parameterization is almost done. The results is shown in Figure 7.3 and Table 7.1 below.

Winch Capacity Estimator

User Inputs	SWL(kg)	10000	Layers	1	1st Layer Length(mm)	107992
	Wire Diameter(mm)	25		2	2nd Layer Length(mm)	111448
	Drum Diameter(mm)	600		3	3rd Layer Length(mm)	114904
	Drum Width(mm)	1396		4	4th Layer Length(mm)	118360
	Drum Thickness(mm)	16		5	5th Layer Length(mm)	121815
	End Disk Diameter(mm)	888		6	6th Layer Length(mm)	125271
	End Disk Width(mm)	16		7	7th Layer Length(mm)	128727
				8	..	0
Results	Wraps Each Layer	55		Verification		
	Drum Max Wire-Capacity(m)	828.517		General Max Layers		7
	Available Wire-Capacity(m)	798.699		Estimated Max Layers		7
	Wire Total Mass(kg)	2494				

Global Parameters	Set Value	
D/d ratio	20	(20~24)
Drum-Sheave Distance(m)	20	
Fleet Angle	0.03	
Max Hoop Stress(MPa)	520	
Max Drum Width(mm)	1396.831	
End Disk Hole Diameter(mm)	60	

Figure 7.3 SWL = 10T Winch Capacity Estimator results

Component	Description	Parameter	Units	Value
Winch Drum	Drum Diameter	d_D	mm	600
	Drum Width	w_D	mm	1396
	Drum Thickness	T_D	mm	16
	End Disk Diameter	d_{ED}	mm	888
	End Disk Thickness	t_{ED}	mm	16
Wire	Wire Diameter	d_W	mm	25
	Wire Total Mass	m_W	Kg	2494
	Wire Length	L_W	m	800
Payload	Mass of Payload	m_{PL}	kg	10000

Table 7.1 SWL = 10T mechanical key parameter specifications

Now we can configure the parametric model on the web page developed by WebGL with Three.js library. The result is shown in Figure 7.4 below.

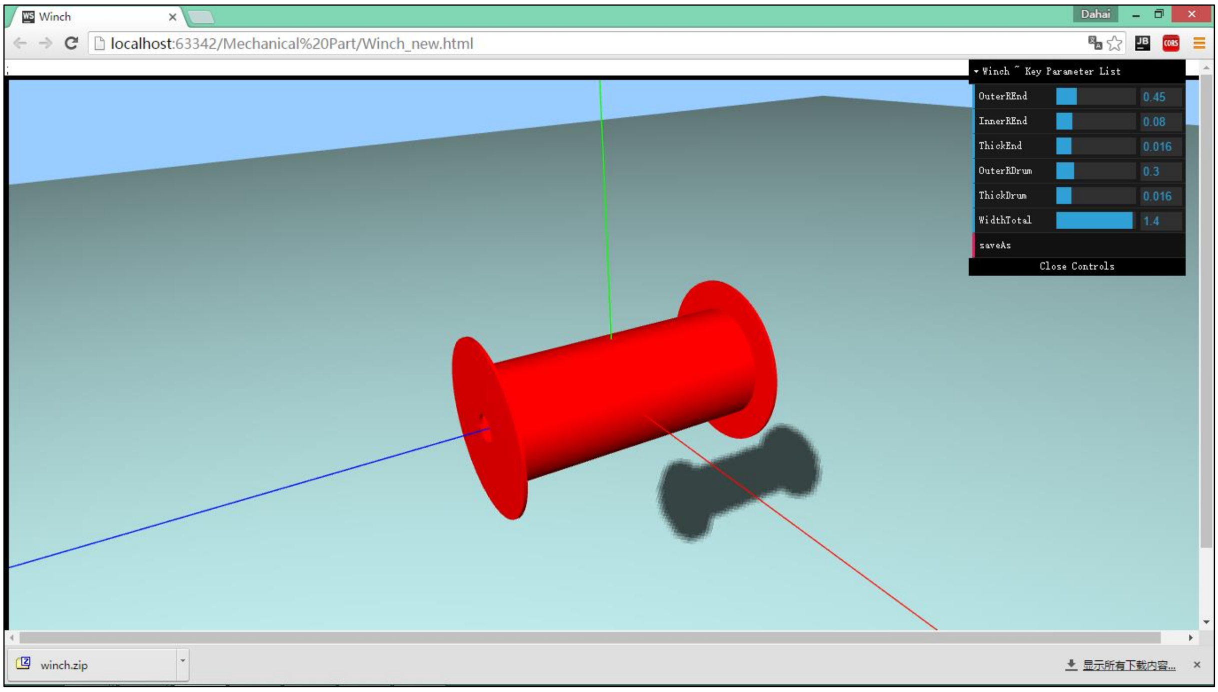


Figure 7.4 SWL = 10T Winch mechanical model based on WebGL

So far, mechanical modelling is done via easily pre-processing parameterization method by spreadsheet and flexible parametric 3D mesh object modelling on web page. Obviously, compared with traditional 3D solid modelling method, this simplified winch mesh model is lack of many design details. However, simplified model is beneficial for real time simulations, rapid parameterization, user interaction and key geometrical features visualization, which particularly decreases the time consumption of manual transcription during the new product development process.

7.1.2 Hydraulic sub-system results

A good mathematical modelling and simulation of physical system depends on the necessary modelling details with proper simplification. This part has already analysed and implemented in Chapter 5, then next step is hydraulic components dimensioning.

The dimensioning starts from payload (AHC requirement) until energy source (El. Motor)

AHC capacity mentioned in the general specification table can be expressed in equations below.

$$\mathrm{x(t)} = A\sin\left(\frac{2\pi}{T}t\right) = 2\sin(0.2\pi t) \qquad \mathrm{v(t)} = 0.4\pi\cos(0.2\pi t) \qquad v_{max} = 1.257\ m/s$$

$$F_{max} = mg = SWL \cdot g = 10^4 \cdot 9.81 = 98.1kN$$

$$\omega_{drum\_max} = \frac{v_{max}}{R_{drum\_1st\_layer}} = \frac{v_{max}}{R_{drum} + R_{wire}} = \frac{1.257}{0.3 + 0.0125} = 4.173\ rad/s$$

$$\tau_{drum\_max} = F_{max} \cdot R_{drum_{7th_{layer}}} = 98.1 \cdot (0.3 + 0.0125 + 6 \cdot 0.8 \cdot 0.025)$$
$$= 42428.3Nm$$

$$n_{motor\_max} = \omega_{drum\_max} \cdot r_{gear} = 4.173 \cdot 46 = 192\ rad/s = 1833\ rpm$$

$$\tau_{motor\_max} = \frac{\tau_{drum\_max}}{r_{gear}} = \frac{42428.3}{46} = 922Nm$$

After checking the technical data of A4VSO axial piston units, A4VSO250DS2 is chosen. That is motor displacement is $250\mathrm{cm}^3$/rev. The maximum speed is 2000rpm.

$$q_{motor\_max} = \frac{V_{g\_motor} \cdot n_{motor}}{1000} = 500\ l/min = 0.00833m^3/s$$

$$D_{pipe\_min} = \sqrt{\frac{4q_{motor\_max}}{\pi \cdot v_{pressure\_line}}} = \sqrt{\frac{4 \cdot 0.00833}{\pi \cdot 8}} = 0.0364m = 36.4mm$$

Since the pump supply all the energy required in this hydraulic system, thus the output flow of pump shall be larger than the required flow. After checking the technical data of A10VO, at least the size 355 is chosen. That is the displacement of the pump is 355cm^3/rev. The maximum drive speed is 2000rpm, which means the output speed of electrical motor is 2000rpm. Then several cases shall be tested for hydraulic models and the simulation integration method is chosen as Runge-Kutta 4.

- Zero-input response

First of all, "zero-input" response shall be tested. "Zero-input" means no extra excitation signal is imposed into the system. The response is caused only by initial state of the system. One of the main purpose is to find proper PID parameters for controllers of the system, such as speed controller. Another purpose is to check whether the dimensioning of hydraulic system is proper or correct. After several trials and modifications, a relatively acceptable hydraulic system with control model is obtained, the result is shown in Figure 7.5, 7.6 and 7.7 below.

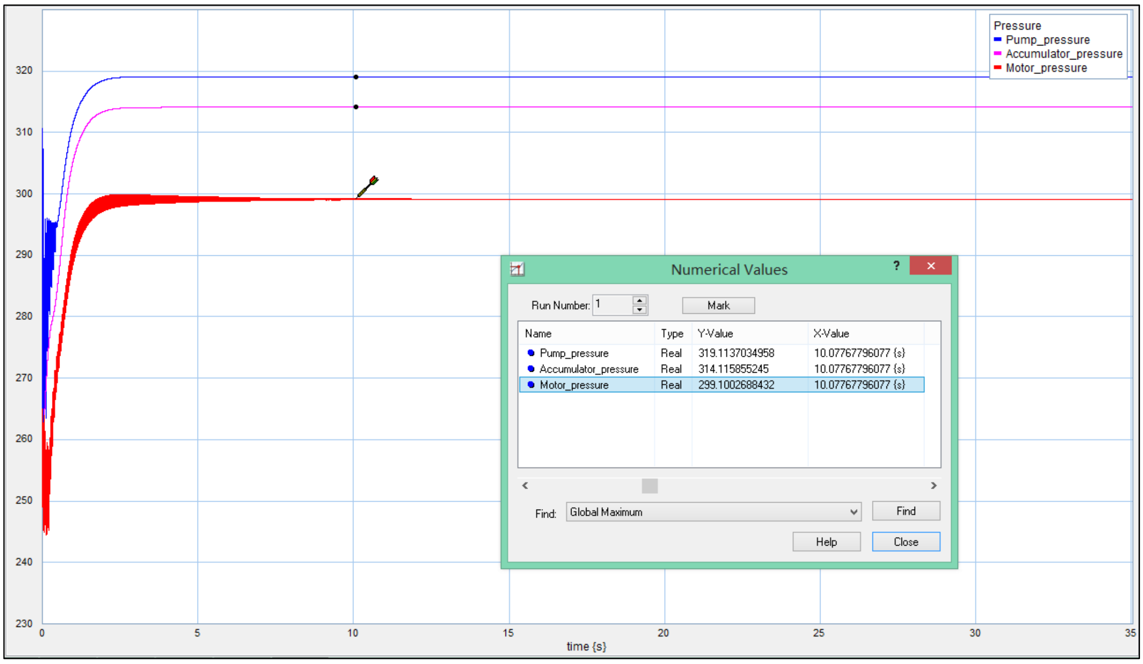


Figure 7.5 SWL = 10T Winch system zero-input results-Pressure

As we can see from Figure 7.5, after approximate 10s settling time, all the pressure values are approaching to steady state. To be more specific, motor operating pressure (red curve) becomes almost constant (299 bar) after settling time, which means the pressure compensated pump is working well. Since pseudo-constant operating pressure is one the pre-conditions to implement secondary control of motor (secondary control unit), the model is able to meet one of the basic requirements of secondary control.

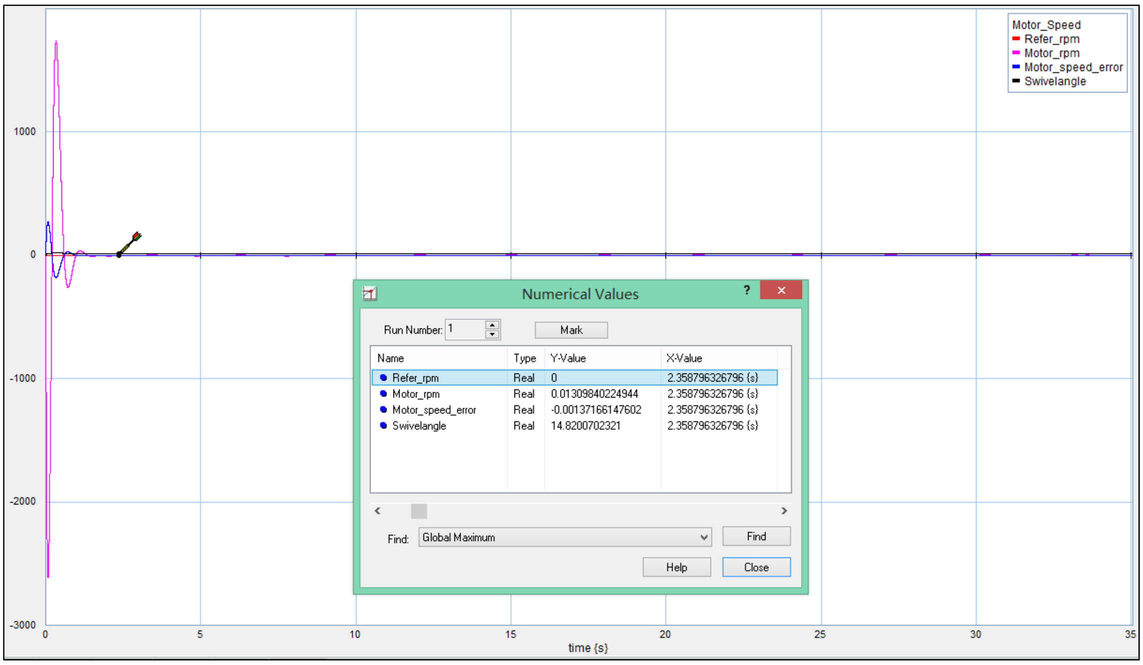


Figure 7.6 SWL = 10T Winch system zero-input results-Motor speed

As we can see from Figure 7.6, after around 2s settling time, motor speed curve and swivel angle start approaching to steady state. More specifically, the error value shown in numerical values table is closing to 0 gradually after settling time, which means the motor speed controller is working properly with current PID parameters. Motor rotational speed (pink curve) fluctuates for a while and then gradually approaches to inputs value (zero-input) which is exactly in accordance with the initiation process in real physical system.

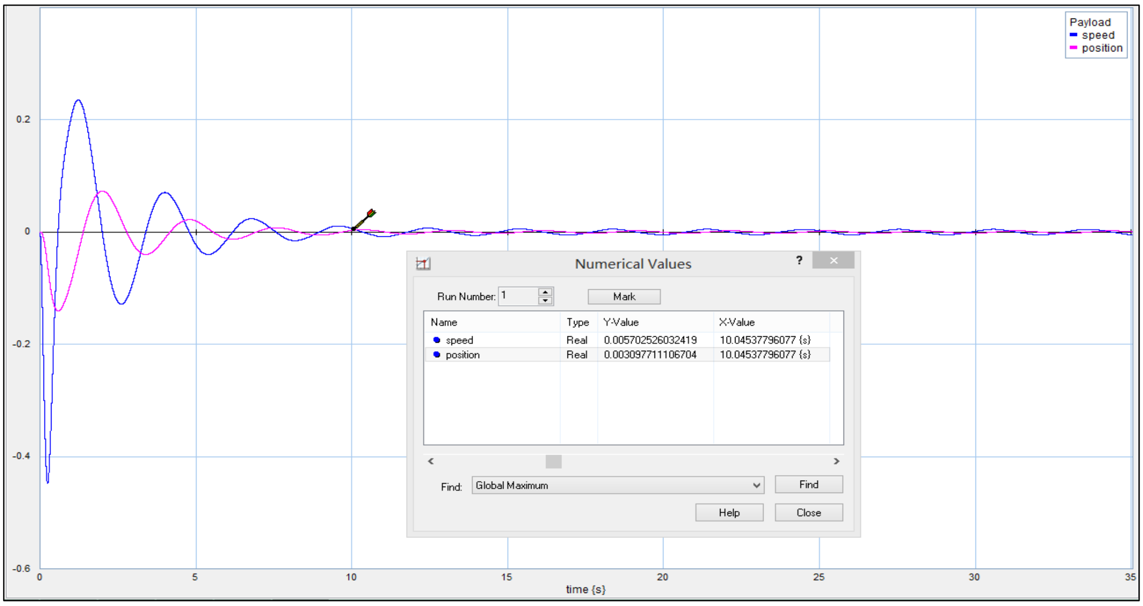


Figure 7.7 SWL = 10T Winch system zero-input results-Payload

Meanwhile, we can also check the initiation results of payload velocity and position in Figure 7.7 below. As can be seen from Figure 7.7, after approximate 10s damped oscillation, payload speed (blue curve) and payload position (pink curve) are both approaching to steady state. The steady state value shown in numerical values illustrates the motor torque and required torque of payload is in balanced state, which is exactly in accordance with the motor speed results shown in Figure 7.6.

- Step-input response

Then we can test the step-input results so as to check the response to the fast change control signal after long time steady state. The starting time of step is set 12s. The results is show below in Figure 7.8, 7.9 and 8.0.

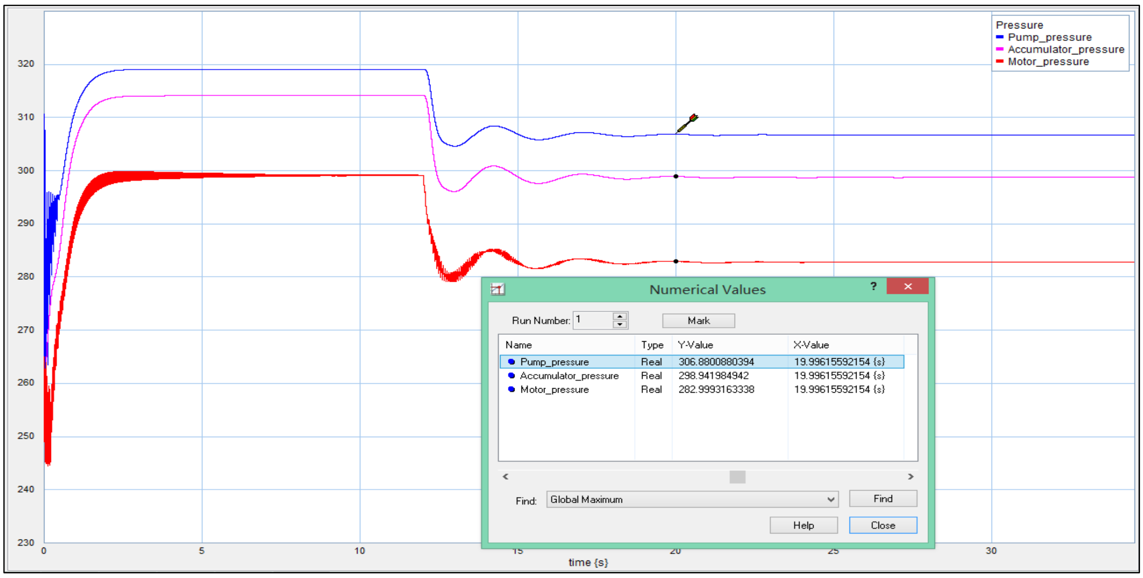


Figure 7.8 SWL = 10T Winch system step-input results-Pressure

As we can see from Figure 7.8, after 12s the pressure drops due to the rapid change, after around another 10s, the pressure values become into steady state again.

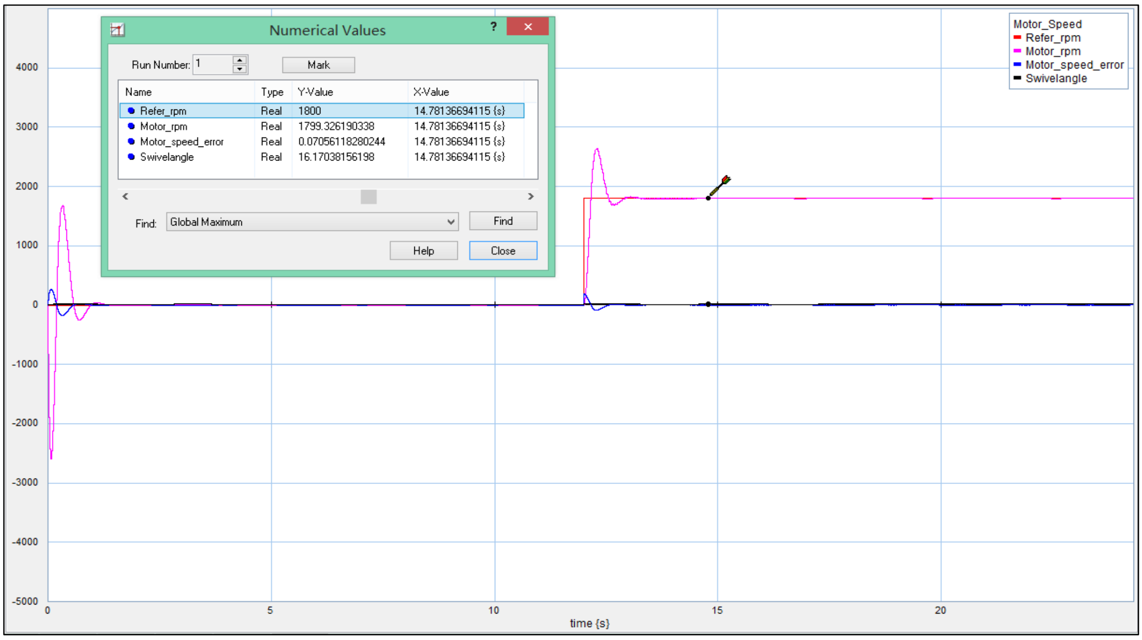


Figure 7.9 SWL = 10T Winch system step-input results-Motor speed

In Figure 7.9, at 10s the initiation steady state changed and settles down to the steady state after approximate 2s. As shown in the numerical values, actual motor speed follows the reference motor speed (step input) quite well, which means the system can response quickly and accurately to a sudden change of the control signal.

And when we check the payload performance in Figure 7.10, similar to the initiation process, the wire speed fluctuates moderately and approaching to the steady state after 8s. As for the position, we can see the payload position changes steadily. Therefore, the step input response tests prove the system is capable of “handling” a sudden change and can change into new steady state quickly, accurately and steadily.

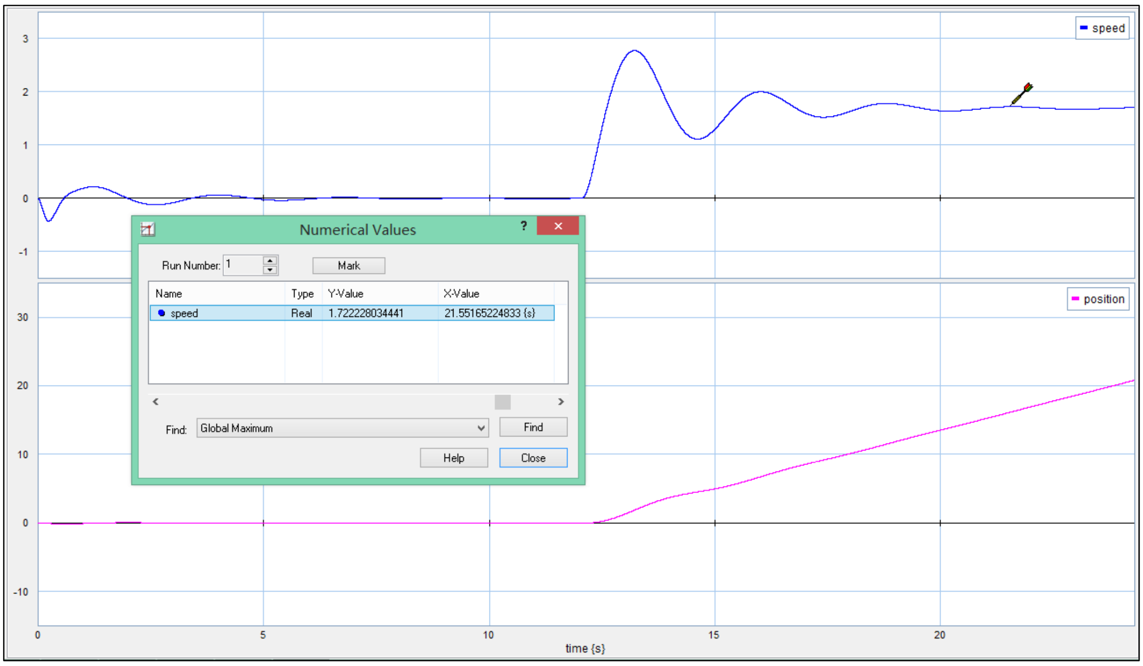


Figure 7.10 SWL = 10T Winch system step-input results-Payload

- Sinusoidal-input response

Now we can have a test of sinusoidal-input response for this system so as to check the response to time-varying control signal. The starting time of sinusoidal is also set 12s. The results are shown in Figure 7.11, Figure 7.12 and Figure 7.13.

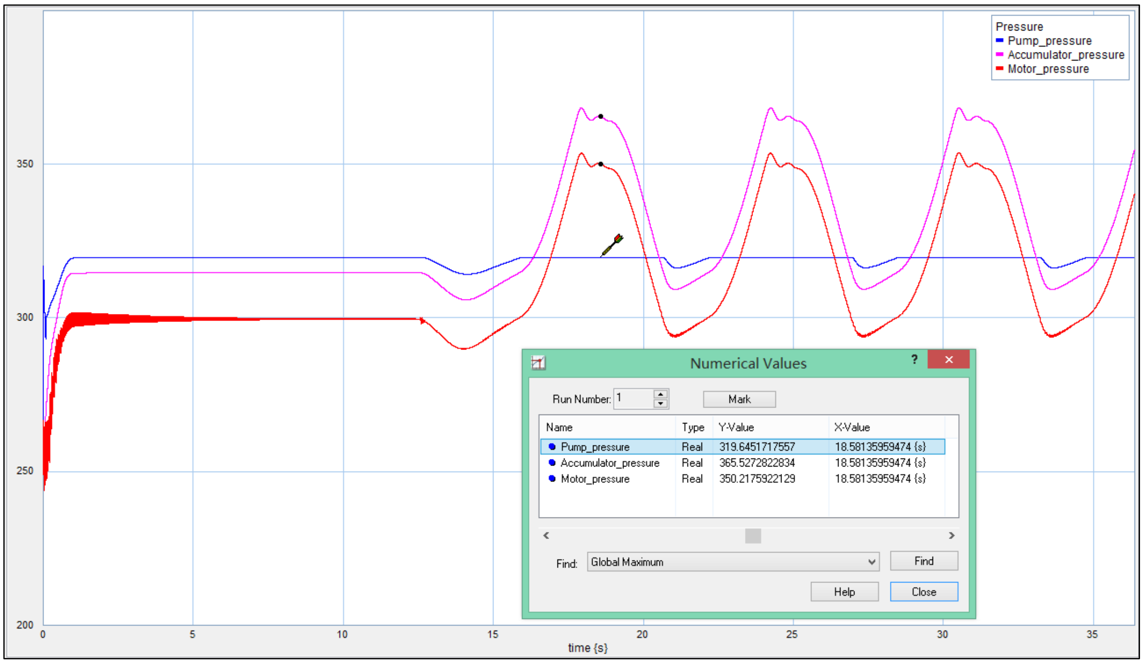


Figure 7.11 SWL = 10T Winch system sinusoidal-input results-Pressure

As can be seem from Figure 7.11, pump pressure (blue curve) tries to maintain the pre-set value (320 bar) and the motor pressure (red curve) changes in accordance with accumulator pressure changes. It is because in secondary control, the motor pressure is dependent on the loading condition of the accumulator and no longer be freely matched to the output torque of motor.

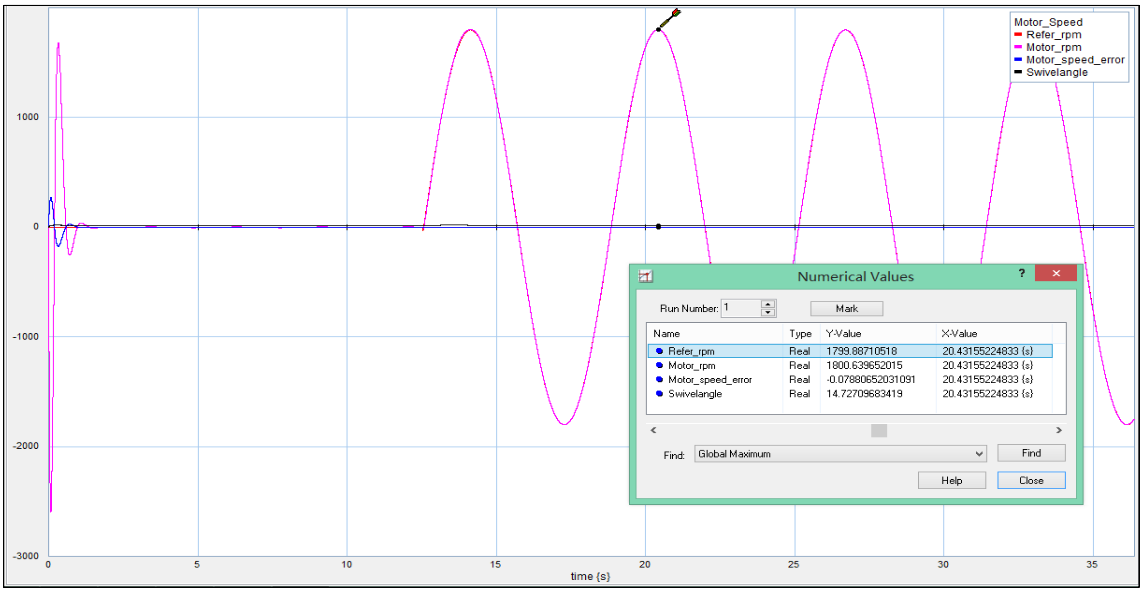


Figure 7.12 SWL = 10T Winch system sinusoidal-input results-Motor speed

In Figure 7.12, from graphic view we can see the actual motor speed (pink curve) and reference motor speed (red curve) are almost coincident. The numerical values also shows the speed error is as low as 0.07 rpm, which turns out that the system is able to “handle” time-varying control signal or continuous changes. That can be a good preparation for active heave compensation.

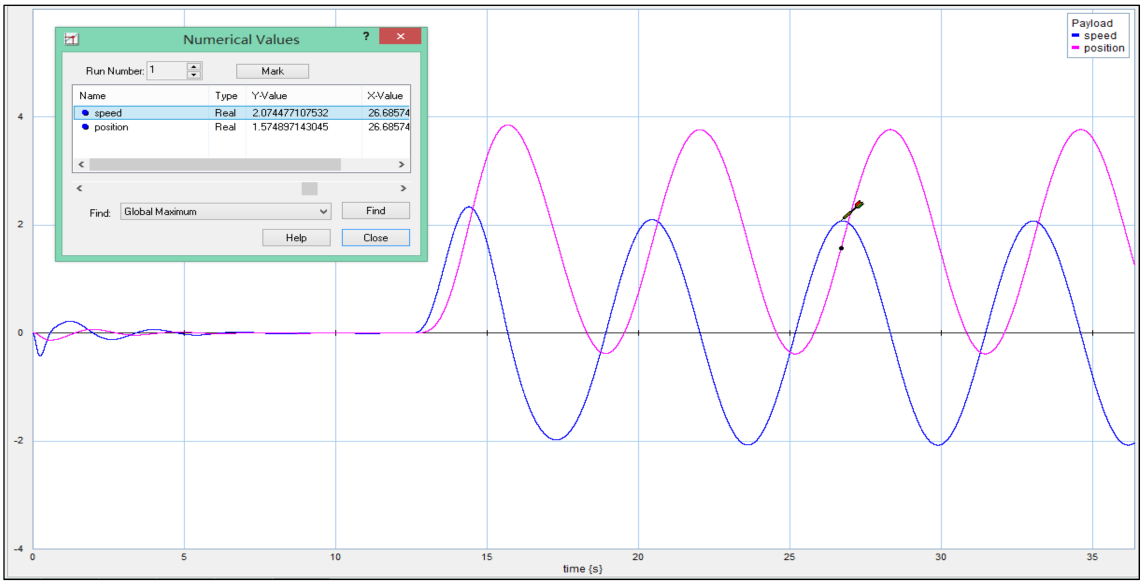


Figure 7.13 SWL = 10T Winch system sinusoidal-input results-Payload

In Figure 7.13, payload speed (blue curve) changes exactly as a sinusoidal curve, which also proves that the system’s ability to handle time-varying control signal.

After those testing described and analyzed above, the hydraulic system can fullfil the basic control requirements. Then we can check the AHC performance.

- Active heave compensation result

AHC capacity	wave amplitude = 2m wave period = 10s

In this testing, a sinusoidal curve with amplitude and period mentioned above will mimic the wave movements acquired by MRU. The wave movement starts at 12s and stops at 60s.

The AHC result is shown in Figure 7.14 and 7.15 below

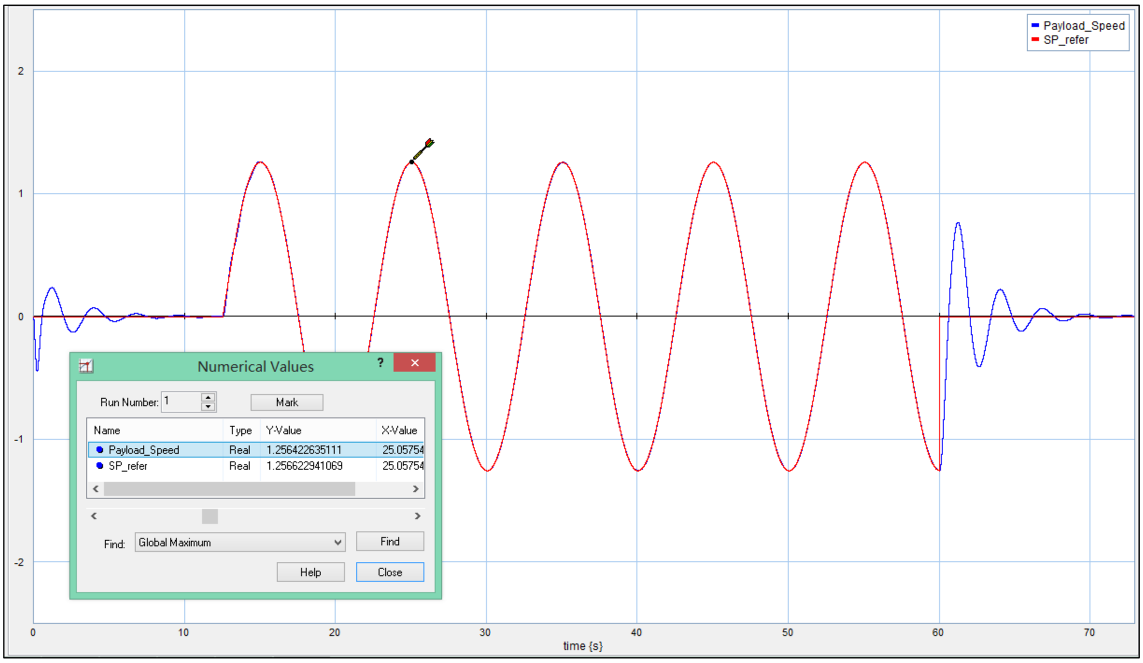


Figure 7.14 SWL = 10T Winch system AHC testing-Payload speed

As shown in Figure 7.14, payload speed (blue curve) can follow the reference speed signal (inverted signal of wave speed) quickly and accurately after the starting time (AHC activated). And after 60s, the refer signal jump to 0 (AHC deactivated). The payload speed experiences damped oscillation for 10s until come into a new steady state. This is the key performance to implement active heave compensation.

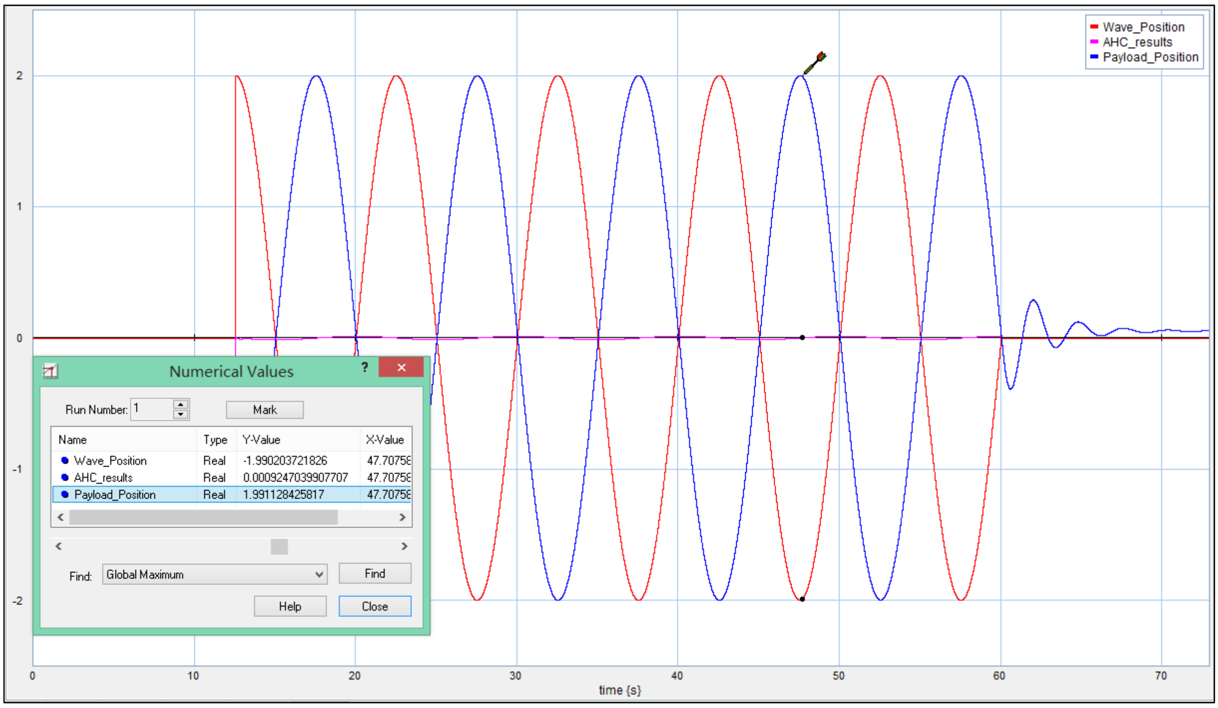


Figure 7.15 SWL = 10T Winch system AHC testing-Payload position

As we can see from Figure 7.15, the payload position changes exactly on the contrary direction of wave position, that is the payload move in advance to compensate the potential effects from the wave. From the numerical values, we can see the absolute minimum differences between wave position and payload position is as small as 1mm. The results prove this hydraulic system is good enough to perform the lowering, hoisting and AHC operations for this winch. Thus this SWL = 10T hydraulic model can be integrated into virtual prototyping framework for co-simulation.

7.2 Co-simulation of VWP

8. Conclusion

In this thesis, a virtual winch prototyping (VWP) system is proposed and implemented, we can conclude that it turns out to be a feasible prototyping solution for winch system during conceptual design phase. To be more specific:

1. VWP is suitable for fast and dynamic requirement changes. In mechanical design, no matter winch capacity estimator or parametric model on web page, they both provide a friendly user interaction solution that can be rapidly configured and properly visualized. In hydraulic design, the key components are developed as a component library, key parameters of them can be configured in global hydraulic parameter sub-model. The mapping between mechanical parameters and hydraulic parameters can be roughly performed inside the winch capacity estimator. As for the integration framework, java programming can provide quite good flexibility for co-simulation of domain-specific models.

2. VWP can decrease conceptual design mistakes or fails before physical prototyping. After checking the co-simulation results inside the physical engine, the key parameters performance can be easily visualized and plotted, which can be compared with the domain-specific simulation results. Therefore, design mistakes or fails can be identified before physical prototyping.

3. VWP largely decrease the potential costs during physical prototyping. One of the reason is mentioned in 2 above, it can decrease the mistakes or fails in conceptual design phase. Another reason is that virtual prototyping is implemented in totally by software, compared with physical prototyping, the costs can be largely decreased.

9. Alternative and future work

9.1 Alternative-Mechanical

Mechanical part can be modelled in many existing 3D modelling software such as NX, SolidWorks and CATIA etc. However, the shortcoming is the model cannot be rapidly parameterized, visualized and user interacted. Another alternative for mechanical model is using AgX. Similar to 3D software, user interaction cannot be well implemented.

9.2 Future work

9.2.1 Mechanical

1. More details shall be modelled in mechanical models, such as drum supports, gear, shaft and etc. Especially gear and shaft, these two parts are key components to drive the winch drum. If possible, the bearing part may be considered.
2. Winch capacity estimator is the pre-processing of key parameters of winch. If this part can be combined with parametric model on the web page, it will be more user-friendly and time-saving.

9.2.2 Hydraulic

1. Key hydraulic components can be modelled with more design details. Pump model is modelled with simplification without modelling the pressure control valve, control cylinder and swash plate inside the axial piston pump.
2. The closed loop speed controller in hydrostatic transmission can be modelled with double feedback loops (speed loop and swivel angle loop) to improve the control accuracy of swivel angle.

Bibliography

Peter McLeod. (November, 2001). *'Low-End' Virtual Prototyping. The Availability and Capabilities of 'Low-End' Virtual Modelling (Prototyping) Products to Enable Designers and Engineers to Prove Concept Early in the Design Cycle*. Page 3.

Patrick Chombart, Dassault System. MODELISAR Innovation report. *Multidisciplinary modelling and simulation speeds development of automotive systems and software*.

MODELISAR consortium (2008-2011) and Modelica Association Project "FMI". *Functional Mock-up Interface for Model Exchange and Co-simulation*. 2.0 Version.

http://functional-mockup-interface.org

www.modelisar.com

Dr. Arun K. Samantaray. (2006). *The System Modelling World-About Bond graphs*.

Eilif Pedersen, Hallvard Engja. (2014). *Mathematical Modelling and Simulation of Physical Systems*. Lecture Notes in TMR4275 *Modelling, Simulation and Analysis of Dynamic Systems.*

Tony Parisi. (2012). *WebGL Up and Running – Building 3D Graphics for the Web*.

Jos Dirksen. (2013). *Learning Three.js: The JavaScript 3D Library for WebGL - Create and animate stunning 3D graphics using the open source Three.js JavaScript library.*

Screaton and Associates. (2013). *Heave Compensation Briefing - Short Briefing Presentation on the basics of Heave Compensation.*

Algoryx Simulations AB. (2016). *AgX Dynamics User Manual.*

Raudi A. Lang. *Hydrostatic drives with control of the secondary unit – The hydraulic trainer, volume 6.*

Tor Eivind Jensen. (2007). The 12th North Sea Offshore Cranes & lifting Conference. *Secondary Control Technology.*

Mannesmann Rexroth. RE 90600/01.98. *Hydraulic Training –Axis Piston Units.*

DNV Standard for certificate No 2.22 (2011) *Lifting Applicances.*

Yingguang Chu, Yuxiang Deng, Birger Skogeng Pedersen, Houxiang Zhang. NTNU in Aalesund. (2016). *Parameterization and visualization of marine crane concept design.*

Yingguang Chu, Lars Ivar Hatledal, Filippo Sanfilippo, Vilmar Æsøy, Houxiang Zhang. NTNU in Aalesund. Virtual Prototyping System formaritime Crane Design and Operation Based on Fuunctional Mock-up Interface.

Appendix-1

Programming codes in JavaScript for parametric model based on WebGL+Three.js

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Winch</title>
    <script type="text/javascript" src=".idea/build/three.js"></script>
    <script type="text/javascript" src=".idea/examples/js/libs/dat.gui.min.js"></script>
    <script src=".idea/examples/js/controls/TrackballControls.js"></script>
    <script src=".idea/examples/js/exporters/OBJExporter.js"></script>
    <script src=".idea/src/jszip.min.js"></script>
    <script src=".idea/src/FileSaver.min.js"></script>
    <style>
        body { margin: 0; }
        canvas { width: 100%; height: 100% ;background: white;}
    </style>
</head>
<body>
<div id="Stats-output"></div>;
<script type="text/javascript">
    var stats, light, mesh, clock = new THREE.Clock(), renderers = [];
    function animate() {
        requestAnimationFrame( animate );//specify a function that is called at an interval
defined by the browser.
        var tim = clock.getElapsedTime() * 0.15;
        for ( var i = 0, l = renderers.length; i < l; i++ ) {
            var r = renderers[i];
            r.renderer.render( r.scene, r.camera );
            if (r.stats) { r.stats.update(); }
            if (r.callback) {
                r.callback();
            } else {
                r.camera.position.x = 20 * Math.cos( tim );
                r.camera.position.y = 20 * Math.cos( tim );
                r.camera.position.z = 20 * Math.sin( tim );
            }
            r.controls.update();
        }
    }
    animate();


    d = document.body;
    var renderer = new THREE.WebGLRenderer({antialias: true});
    renderer.setClearColor(0x99CCFF);
    renderer.setSize( window.innerWidth, window.innerHeight );
    renderer.shadowMapEnabled = true;
    renderer.shadowMapSoft = true;
    renderer.shadowMap.Type = THREE.PCFSoftShadowMap;
    renderer.domElement.style.border = '5px solid black';
    renderer.domElement.style.backgroundColor = '#000';
    d.appendChild( renderer.domElement );



    //set up scene
    var scene = new THREE.Scene();
```

```
    scene.fog = new THREE.Fog( 0x0099FF, 1, 5000 );
    scene.fog.color.setHSL( 0.6, 0, 1 );
    //set up perspective from camera
    var camera = new THREE.PerspectiveCamera( 45, window.innerWidth/window.innerHeight, 1,
1000 );
    //Set up camera position
    camera.position.set(10, 5, 5);
    camera.lookAt(new THREE.Vector3(0,0,0));
    //set up trackball
    var controls = new THREE.TrackballControls( camera, renderer.domElement );
    //show axes
    var axes = new THREE.AxisHelper( 20 );
    scene.add(axes);


    var spotLight = new THREE.SpotLight(0xffffff, 1);
    spotLight.position.set(10, 10, 10);
    spotLight.castShadow = true;
    spotLight.shadowCameraNear = 0.01;
    spotLight.shadowDarkness = 0.5;
    var ambient = new THREE.AmbientLight(0x444444);
    scene.add(ambient);
    scene.add(spotLight);





    //plane for shadow receive
    var planeGeometry = new THREE.PlaneGeometry(100,100);
    var planeMaterial = new THREE.MeshPhongMaterial({
        color: 0xCCFFFF,
        shininess: 150,
        specular: 0x888888,
        shading: THREE.SmoothShading
    });
    var plane = new THREE.Mesh(planeGeometry,planeMaterial);
    plane.rotation.x = -0.5*Math.PI;
    plane.position.x = 0;
    plane.position.y = -5;
    plane.position.z = 0;
    plane.receiveShadow = true;
    scene.add(plane);



    //Constructor Function to create the basic components
    CreateTube = function(geo)
    {
        //key parameters of tube
        this.innerR = 1;
        this.outerR = 1.2;
        this.width = 1.5;
        this.segments = 100;

        //Initial rotation and position of tube
        this.rot = [0,0,0];
        this.pos = [0,0,0];

        //Generate CreateTube function
```

```
    this.generate = function()
    {
        var geometry = new THREE.Geometry();
        var twopi = 2*Math.PI;

        //Generate vertices of tube(length = 4*segments)
        for (var i=0;i<this.segments;i++)
        {
            var RAD = i/this.segments;
            var XB = this.outerR * Math.cos(RAD * twopi);//outer Bottom Vertices X axis value
            var YB = this.outerR * Math.sin(RAD * twopi);//outer Bottom Vertices Y axis value
            var XT = this.outerR * Math.cos(RAD * twopi);//outer Top Vertices X axis value
            var YT = this.outerR * Math.sin(RAD * twopi);//outer Top Vertices Y axis value
            var xT = this.innerR * Math.cos(RAD * twopi);//Inner Top Vertices x axis value
            var yT = this.innerR * Math.sin(RAD * twopi);//Inner Top Vertices y axis value
            var xB = this.innerR * Math.cos(RAD * twopi);//Inner Bottom Vertices x axis value
            var yB = this.innerR * Math.sin(RAD * twopi);//Inner Bottom Vertices y axis value

            geometry.vertices.push(new THREE.Vector3(XB,YB,-0.5*this.width));
            geometry.vertices.push(new THREE.Vector3(XT,YT,0.5*this.width));
            geometry.vertices.push(new THREE.Vector3(xT,yT,0.5*this.width));
            geometry.vertices.push(new THREE.Vector3(xB,yB,-0.5*this.width));
        }

        //Generate closed faces of tube(length = 4*segments-8)
        for (i=0;i<this.segments-1;i++)
        {
            geometry.faces.push(new THREE.Face3(i*4,i*4+5,i*4+1));//outer face
            geometry.faces.push(new THREE.Face3(i*4,i*4+4,i*4+5));//outer face
            geometry.faces.push(new THREE.Face3(i*4+2,i*4+7,i*4+3));//inner face
            geometry.faces.push(new THREE.Face3(i*4+2,i*4+6,i*4+7));//inner face
            geometry.faces.push(new THREE.Face3(i*4+1,i*4+6,i*4+2));//upper face
            geometry.faces.push(new THREE.Face3(i*4+1,i*4+5,i*4+6));//upper face
            geometry.faces.push(new THREE.Face3(i*4,i*4+3,i*4+7));//bottom face
            geometry.faces.push(new THREE.Face3(i*4,i*4+7,i*4+4));//bottom face
        }

        //Fully closed all the faces
        geometry.faces.push(new THREE.Face3(1,4*this.segments-3,4*this.segments-4)); //outer face
        geometry.faces.push(new THREE.Face3(1,4*this.segments-4,0)); //outer face
        geometry.faces.push(new THREE.Face3(2,4*this.segments-1,4*this.segments-2)); //inner face
        geometry.faces.push(new THREE.Face3(2,3,4*this.segments-1)); //inner face
        geometry.faces.push(new THREE.Face3(2,4*this.segments-2,4*this.segments-3)); //upper face
        geometry.faces.push(new THREE.Face3(2,4*this.segments-3,1)); //upper face
        geometry.faces.push(new THREE.Face3(0,4*this.segments-1,3)); //bottom face
        geometry.faces.push(new THREE.Face3(0,4*this.segments-4,4*this.segments-1)); //bottom face

        //Define position and rotation of tube
        geometry.rotateX(this.rot[0]);
        geometry.rotateY(this.rot[1]);
        geometry.rotateZ(this.rot[2]);
        geometry.applyMatrix(new THREE.Matrix4().makeTranslation(this.pos[0],this.pos[1],this.pos[2]));

        //Merge new geometry into former geometry
        geo.merge(geometry,geometry.matrix);
```

```
            //Dispose the temporary geometry
            geometry.dispose();
        }
    };
    //Constructor Function to create the whole Winch
    Winch = function(mesh)
    {
        //key parameters of tube
        this.OuterREnd = 2.5;
        this.OuterRDrum = 1.5;
        this.InnerREnd = 0.3;
        this.ThickEnd = 0.3;
        this.ThickDrum = 0.2;
        this.WidthTotal = 5;

        //Initial rotation and position of tube
        this.rot = [0,0,0];
        this.pos = [0,0,0];

        //Generate Drum function
        this.generate = function()
        {
            mesh.geometry.dispose();
            mesh.geometry = new THREE.Geometry();

            var tube1 = new CreateTube(mesh.geometry);
            tube1.outerR = this.OuterRDrum;
            tube1.innerR = this.OuterRDrum - this.ThickDrum;
            tube1.width = this.WidthTotal -this.ThickEnd*2;
            tube1.generate();

            var tube2 = new CreateTube(mesh.geometry);
            tube2.outerR = this.OuterREnd;
            tube2.innerR = this.InnerREnd;
            tube2.width = this.ThickEnd;
            tube2.pos = [0,0,this.WidthTotal/2-this.ThickEnd/2];
            tube2.generate();

            var tube3 = new CreateTube(mesh.geometry);
            tube3.outerR = this.OuterREnd;
            tube3.innerR = this.InnerREnd;
            tube3.width = this.ThickEnd;
            tube3.pos = [0,0,-this.WidthTotal/2 + this.ThickEnd/2];
            tube3.generate();


        }
    };

    //Basic Winch Model Initiation
    var material = new
THREE.MeshPhongMaterial({color:0xFF0000,specular:0x555555,shininess:100,shading:
THREE.FlatShading});
    var winch_mesh= new THREE.Mesh(new THREE.Geometry(),material);
    winch_mesh.castShadow = true;
    scene.add(winch_mesh);
    var winch = new Winch(winch_mesh);
    winch.generate();
```

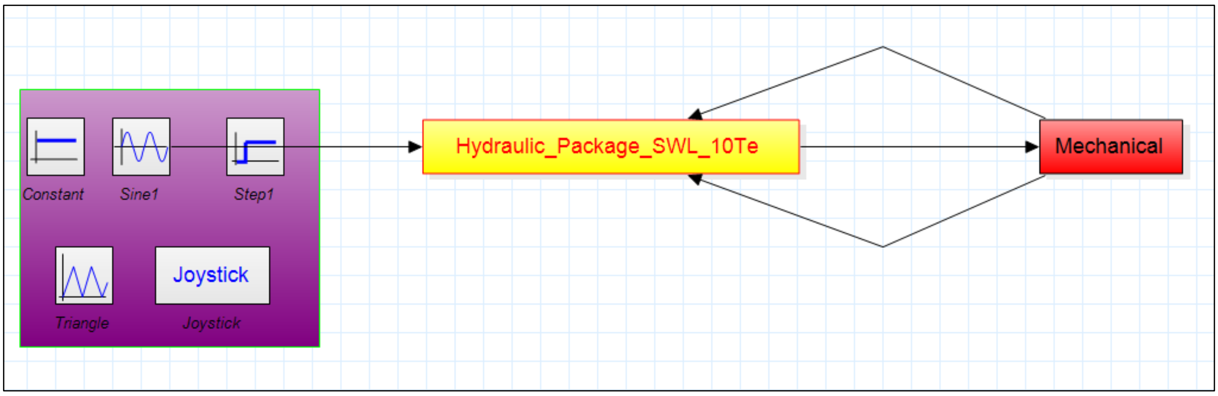
```
//Constructor function for control box
var boxConfig =
{
    OuterREnd : 0.7,
    OuterRDrum : 0.5,
    InnerREnd : 0.1,
    ThickEnd : 0.05,
    ThickDrum : 0.05,
    WidthTotal : 1.8,
    WireFrame : false,
    density : 7800
};


var save ={
saveAs :  function()
{
    var exporter = new THREE.OBJExporter();
    var obj = exporter.parse(winch_mesh );
    //var blob = new Blob([obj], {type:"text/plain;charset=utf-8"});
    //saveAs(blob,"Winch.obj");
    //console.log("hello world");

    var zip = new JSZip();
    zip.file("winch.obj",obj);
    zip.file("winch.json",JSON.stringify(boxConfig,null,'\t').replace(/[\n\t]+([\d\.e\-
\[\]]+)/g, '$1'));
    var zipfile = zip.generate({type:'blob'});
    saveAs(zipfile,"winch.zip");
}
};


//Control box for dimensioning changes
// var  boxConfig = new boxConfigData();
var boxGui = new dat.GUI();
var guiBox = boxGui.addFolder('Winch ~ Key Parameter List');
guiBox.open();

//Model Parametrization
guiBox.add( boxConfig, 'OuterREnd', 0.2, 1.2 ).step(.001).onChange( function()
{
    winch.OuterREnd = boxConfig.OuterREnd;
    winch.generate();
} );

guiBox.add( boxConfig, 'InnerREnd', 0.02, 0.3 ).step(.001).onChange( function()
{
    winch.InnerREnd = boxConfig.InnerREnd;
    winch.generate();
} );

guiBox.add( boxConfig, 'ThickEnd', 0.001, 0.08 ).step(.001).onChange( function()
{
    winch.ThickEnd = boxConfig.ThickEnd;
    winch.generate();
} );

guiBox.add( boxConfig, 'OuterRDrum', 0.1, 1 ).step(.001).onChange( function()
{
```

```
        winch.OuterRDrum = boxConfig.OuterRDrum;
        winch.generate();
    } );

    guiBox.add( boxConfig, 'ThickDrum', 0.001, 0.08 ).step(.001).onChange( function()
    {
        winch.ThickDrum = boxConfig.ThickDrum;
        winch.generate();
    } );

    guiBox.add( boxConfig, 'WidthTotal', 0.1, 1.5 ).step(.001).onChange( function()
    {
        winch.WidthTotal = boxConfig.WidthTotal;
        winch.generate();
    } );

    guiBox.add( save, 'saveAs' );


    var render = function ()//animation
    {
        requestAnimationFrame( render );
        renderer.render(scene, camera);
    };

    render();
    var callback = function() {
    };

    renderers.push( {renderer: renderer, scene: scene, camera: camera, controls: controls,
callback: callback} );


  //  var exporter = new THREE.OBJExporter();
   // var obj = exporter.parse(winch_mesh );

    //console.log(obj);
   // saveAs(obj,"Winch.obj");

</script>

</body>
</html>
```

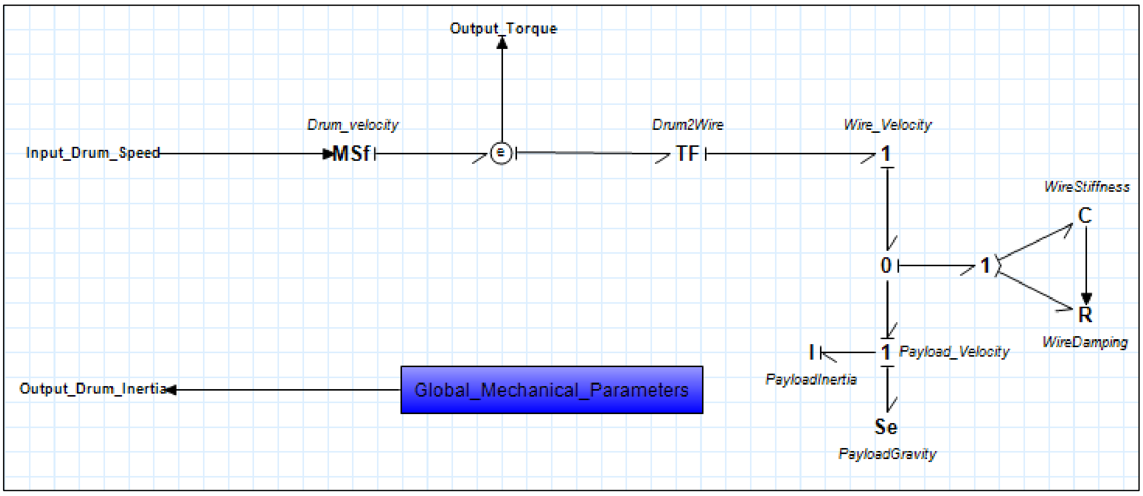
Appendix-2

Hydraulic 20Sim model

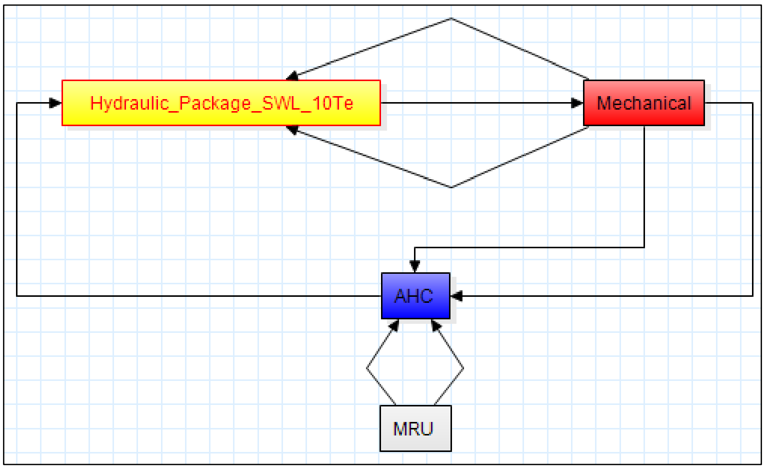
First layer without AHC



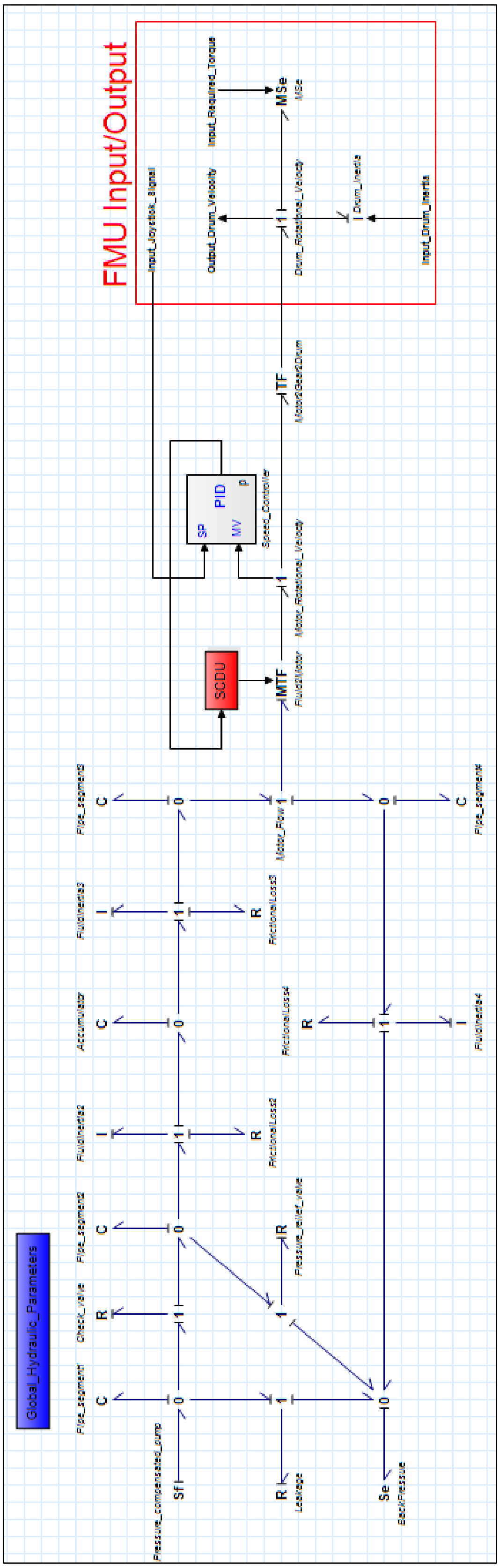
Mechanical



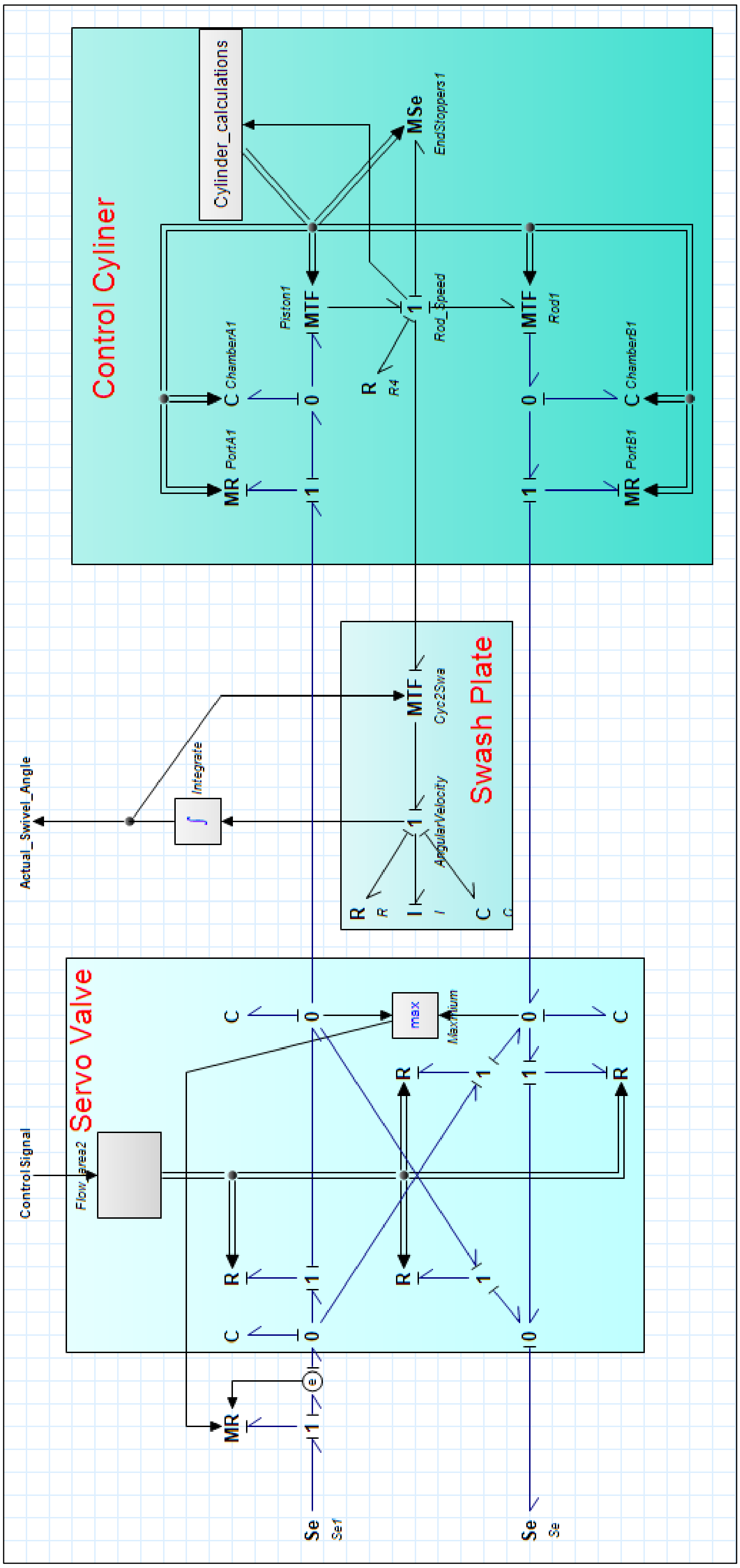
AHC



Hydraulic Package SWL 10Te



Hydraulic Package SWL 10Te-SCDU


Control Cyliner
Cylinder_calculations
MSe
EndStoppers1
MTF
Piston1
Rod_Speed
Rod1
R
R4
C
ChamberA1
ChamberB1
MR
PortA1
PortB1
Swash Plate
Cyc2Swa
AngularVelocity
Integrate
Actual_Swivel_Angle
Servo Valve
Control Signal
Flow_area2
max
Maximum
Se
Se1