University of Miskolc
Faculty of Earth Science and Engineering
Mikoviny Sámuel Doctoral School of Earth Sciences
Head of the Doctoral School: Prof. Dr. Péter Szűcs, PhD, Doctor of Science

TORQUE OPTIMIZATION OF SUCKER-ROD PUMPING UNITS

PhD Thesis

Department Defense Version

by

László Kis
Petroleum and natural gas engineer

Fluid production and transporting research section
Head of the section: Prof. Dr. László Tihanyi, Professor Emeritus

Integrated petroleum and natural gas production systems topic
Head of the topic: Prof. Dr. Gábor Takács, Professor Emeritus

Advisor
Prof. Dr. Gábor Takács, Professor Emeritus

2020

Table of Contents

1 Acknowledgements

2 Advisor's Foreword

The great majority of artificial lifted oil wells are placed on sucker-rod pumping all over the world. Due to the great importance of sucker-rod pumping the reduction of production costs is a major drive in operating those installations. Because the most significant element of production costs is related to the prime mover's energy requirement the improvement of power efficiency is a prime task of field personnel. The proper choice of the pumping unit's counterbalancing, the topic of this candidate's PhD Thesis, can substantially improve the power conditions of pumping and thus can increase the profits of oil production.

The candidate's choice of the topic of his PhD Thesis is especially appropriate today because of the great number of rod pumped wells worldwide as well as in Hungary. The results of the author's interesting and important research will surely help to increase pumping efficiency and, at the same time, increase the life of sucker-rod pumping installations.

The Thesis is properly constructed and clearly proves the candidate's skills in scientific research and publication. His treatment of the gearbox's torque loading under different kinds of counterbalancing conditions is correct. One of the best parts of the Thesis deals with unusual counterbalance arrangements that are very seldom used in the industry. As the author proves, the use of asymmetric counterweight arrangements, as compared to the traditional symmetric ones, can lead to definite operational advantages. The author, for the first time in the literature, introduces the use of Particle Swarm Optimization (PSO) method in the calculation of optimum counterbalancing conditions. The novel methods and calculation models developed by the author can be considered as new scientific achievements in the discipline of sucker-rod pumping of oil wells.

Budapest, October 13, 2020.

Prof. Dr. Gábor Takács
Petroleum Engineering Department
University of Miskolc

3 Introduction

The initial objective of the presented thesis was the investigation of the effect of using asymmetrical counterweight configuration in the counterbalancing of crank balanced sucker-rod pumping units. During the research it became clear, that this particular case of counterbalancing was not investigated in detail prior to this work. Since the number of sucker-rod pumping installations operating worldwide is approximately 750,000, it is important to ensure their optimal operation from both engineering and economical points of view. (SPE) The profitability of these oil producing units is increased by achieving the lowest power requirement possible for the desired liquid flow rate, which depends on mainly the torque loading of the unit's gearbox. Providing a sufficiently long lifetime for the installation by protecting the gearbox – its most expensive part – from overloading also improves the economic value of the sucker-rod pumping unit operation.

Before the optimization of the net gearbox torque can be carried out, the knowledge of all distinct torque components acting on the gearbox throughout the pumping cycle is necessary to accurately describe the actual torque conditions of the investigated pumping unit. The improved torque analysis presented in this work is based on the data provided by an electronic dynamometer, the routinely used measurement tool for sucker-rod pumping units. This measurement technique is the most widespread supervision type that has the required accuracy for a complete torque analysis and can be carried out with little effort. The refined procedure of the dynamometer survey evaluation is presented for four pumping unit geometries – Conventional, Air balanced, Mark II and Reverse Mark – but it can be modified to handle any special geometry type with little effort. An example problem is introduced, and the results of its evaluation are presented for every major calculation step to help the easier interpretation of the proposed calculation method.

After studying the API Spec 11E (API, 2008) – the recommendation by the American Petroleum Institute – it became apparent, that the evaluation method used in the industry lacks the capability to handle the proper description of those sucker-rod pumping units that have varying crank angular velocities throughout their pumping cycle. This condition occurs when either the pumping unit is operated in an unbalanced condition, or when a high slip, or ultra-high slip prime mover is used to drive the sucker-rod pumping unit. Therefore, the improvement of the interpretation of dynamometer surveys was the first crucial step after outlining the research goals.

Beside the literature research, the most widely used software in the petroleum industry for the evaluation of dynamometer surveys – the Total Well Management by Echometer – was inspected and its results were analyzed in detail. After identifying shortcomings in the results of the TWM software and in the relevant literature, the objectives of the research were expanded with the identification of the systematic errors. A comprehensive calculation procedure is proposed that determines the behavior of the sucker-rod pumping unit with higher accuracy than any already existing method; the findings of relevant publications in the topic are incorporated and new solutions are presented to address previously unresolved calculation steps.

Finding the crank angle values at the measured times with the highest accuracy possible is essential for the proper torque analysis because all torque components depend on the crank angle versus time function, which is not included in the dynamometer survey. The proposed method gives more accurate crank angle values than the programs currently used in the industry. The determination of the angular acceleration pattern of the crank arm and the walking beam are necessary for the calculation of the inertial torques acting on the gearbox. Several methods are presented and compared, providing the angular acceleration functions in time with the highest precision possible using the calculated crank angle values. After the evaluation of the kinematic behavior of the sucker-rod pumping unit, all torque components acting on the gearbox in time are found. The knowledge of these torque functions is the basis of any optimization procedure.

The second main objective of the research was to develop a calculation method to optimize the mechanical net gearbox torque and to determine the corresponding counterweight configuration for the investigated sucker-rod pumping unit. For this purpose, a particle swarm optimization (PSO) algorithm was used, due to the size of the solution space. By properly considering the effect of the asymmetrically placed counterweights, the number of independent variables increases from three – in the case of symmetrical counterweight configuration – to twelve; which makes the direct determination of the optimal arrangement of the counterweights impossible. The asymmetrically placed counterweights not only change the counterbalance torque by introducing a secondary phase angle but will alter both the rotary inertial torques as well. Hence the optimization procedure is more complex, but the resulting solution provides better torque loading of the gearbox for a given operating condition. Using this artificial intelligence technique, the resulting mechanical net gearbox torque function is superior to the output of the investigated TWM software. A novel optimization strategy was developed to maximize the cost savings of the operation of the sucker-rod pumping units while preventing the overloading of the gearbox. A computer program has been developed in C# to carry out the presented calculation steps.

4 Overview of Sucker-rod Pumping

Oil wells usually flow naturally in the early stages of their lives. At this point the pressure at the well bottom is enough to lift the reservoir liquid to the surface overcoming the pressure losses in the well. However, if the bottomhole pressure of a given oil well decreases due to the liquid and gas production, at some point an artificial production method has to be implemented to keep the wellhead pressure at the minimal level, so that the reservoir liquid is lifted to the surface. The artificial lifting method investigated in this thesis is sucker-rod pumping.

4.1 Relevance of Sucker-rod Pumping

The number of sucker-rod pumping installations can only be estimated, their exact number is unknown. According to recent estimates, there are approximately 2 million oil wells worldwide of which more than 50% are operated with some kind of artificial lift (Lea, 2007). The share of different artificial production methods is shown in **Figure 1** along with their respective production contribution based on the ALRDC (Artificial Lift Research and Development Council) estimates. (Takács, 2015)

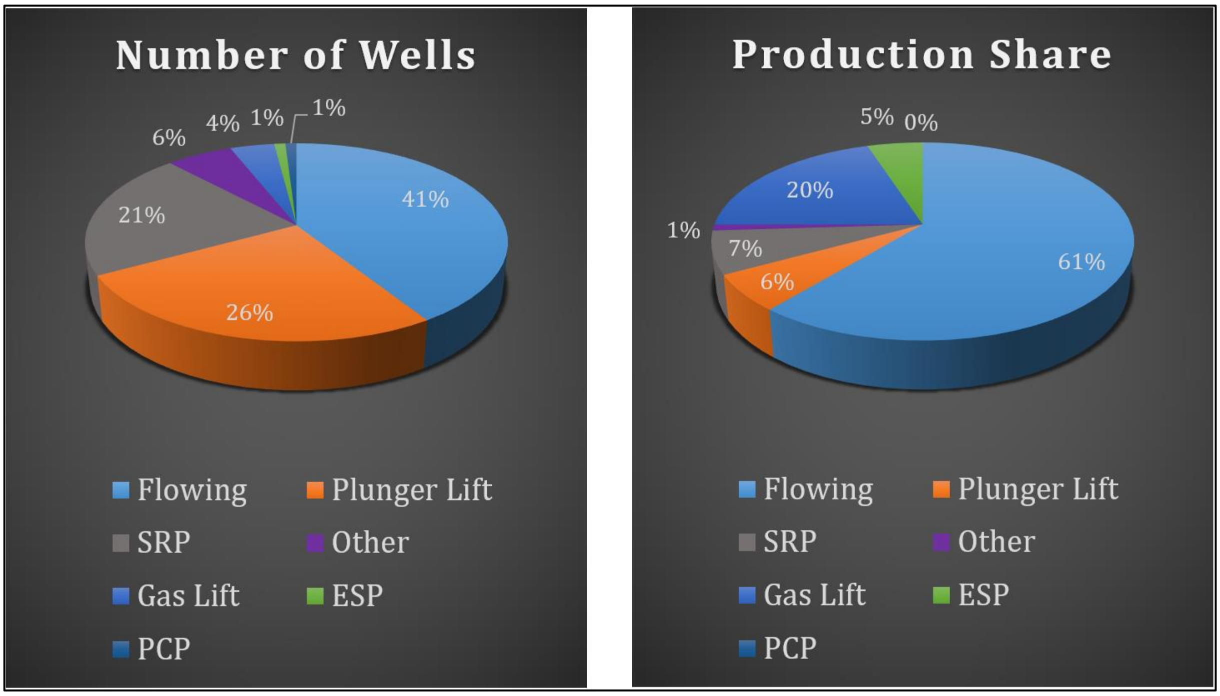


Figure 1 *The estimated number and production of different artificial lifting installations (Takács, 2015), own edit*

The current share of sucker-rod pumping is 21% globally, their production contribution is 7%, therefore it is crucial to maintain optimum operating conditions for such installations. The basic objective of production engineers is to safely operate wells using the least amount of operating cost to meet the required liquid regime.

Power costs in sucker-rod pumping operations are related to the surface power required to drive the pumping system. This power, in turn, depends mainly on the mechanical net torque required at the gearbox of the pumping unit. Thus, proper calculation of gearbox torque during the pumping cycle is essential to accurately determine the power requirements and operating costs of sucker-rod pumping. (Takács, 2003)

4.2 Operation of Sucker-Rod Pumps

Sucker-rod pumping was the first artificial lifting method used in the petroleum industry. In the early years, cable tool drilling was the dominant drilling method, in which the drilling bit was dropped and retrieved repeatedly by a connected cable. After the flowing state of the well stopped, a bottomhole plunger pump was placed in the bottom of the well and was operated by the walking beam. This was the ancestor of the later widely used sucker-rod pumping systems. The materials used changed from wood to steel, but the operational principles stayed the same ever since.

The schematic diagram of a typical sucker-rod pumping unit is shown in **Figure 2**. The objective of the surface equipment's design is to transform the rotational motion of the prime mover into an alternating motion of the polished rod at the wellhead. This reciprocating motion is used to operate a subsurface positive displacement pump situated below the static liquid level. The connection between the surface and the subsurface equipment is the polished rod with precisely manufactured surface that ensures the proper seal at the stuffing box while moving in it. To protect the polished rod from bending, it is only allowed to move vertically, this is ensured by the proper design of the horsehead.

The connection between the polished rod and the downhole pump is provided by the rod string. The rod string is tapered, having decreasing sizes towards the pump. The optimal rod shape is a downward pointing cone, this shape is approximated with the properly designed rod string to withstand the most common rod failure type, the fatigue break. The pump consists of a stationary cylinder – the pump barrel – with a standing valve, a travelling valve, and the plunger. The operation of the unit is powered by the prime mover, which is usually an electric motor. The rotational speed of the motor is decreased to operate the sucker-rod pumping system at a reasonable pumping speed. The gear reducer – or gearbox – is the unit responsible for the decrease of the rotational speed while simultaneously increasing the torque. During upstroke the prime mover lifts the rod string along with the liquid column above the pump. While lifting the fluid the travelling valve is closed and the standing valve is open. In downstroke however, the rod string falls in the liquid with open travelling valve and closed standing valve. The power requirement changes significantly during the pumping cycle. To achieve an improved power draw from the motor, counterweights, or other applicable counterbalancing methods are used. In the case of crank balanced units, the aim of the counterweights is to brake the rod string in the downstroke, when the rod string is falling in the liquid, and to help lift in the rod string and the produced liquid in the upstroke. In downstroke energy is stored in the counterweights by lifting them and the motor is prevented from functioning like a generator. The stored energy is released whilst upstroke, reducing the power requirement needed to lift the rod string.

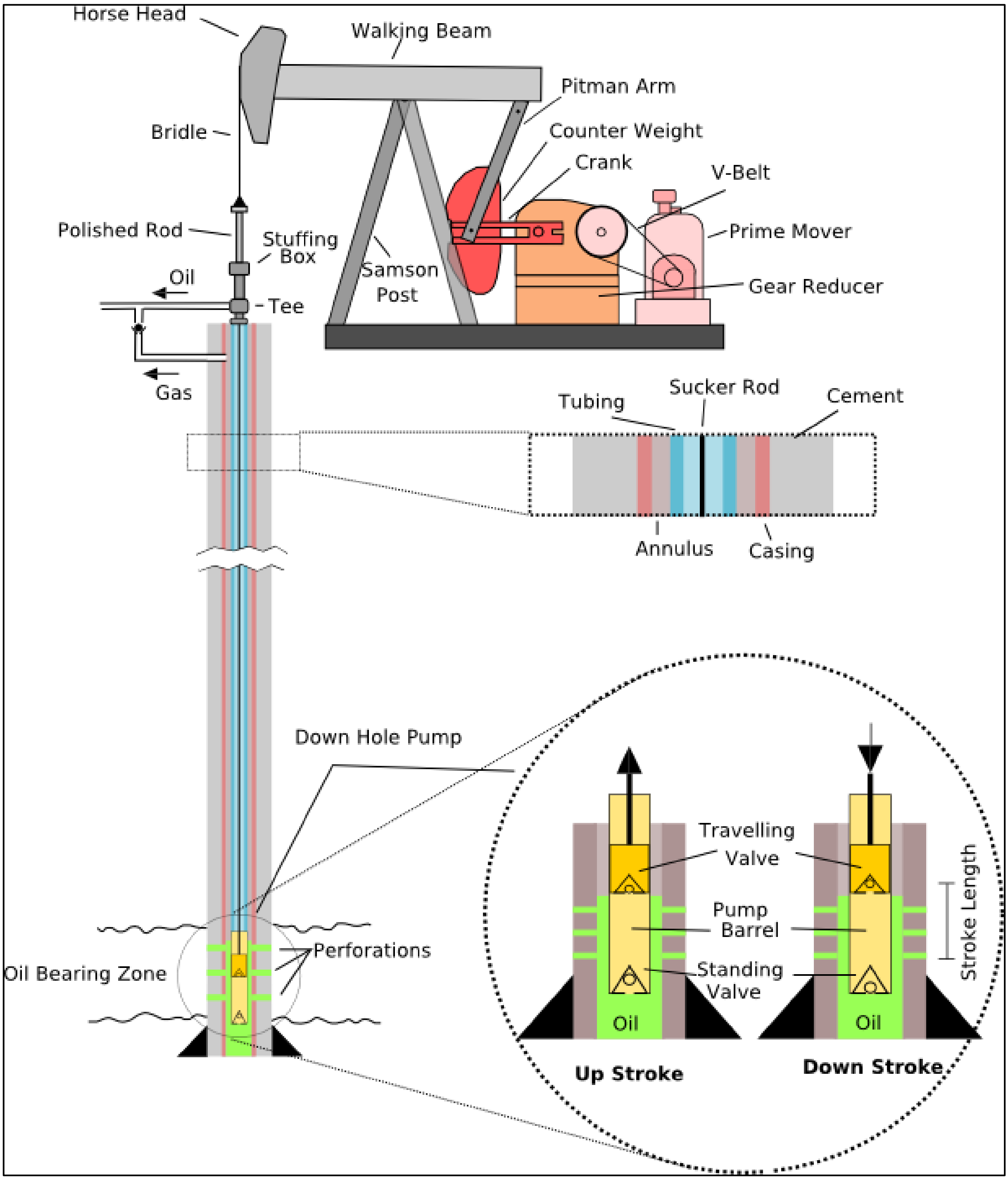


***Figure 2** The sucker-rod pumping system (Danel, 2015)*

4.2.1 Gearboxes

Since the prime mover – usually an electric motor – has extremely high rotational speed to turn the crank arm of a sucker-rod pumping unit directly, a gear reducer is used to slow down the speed to a desired value and to increase the output torque simultaneously. The gear reducers are the most expensive parts of the sucker-rod pumping units with around 50% Capex share. (Takács, 2015) API Spec. 11E (API, 2008) contains the relevant properties of the standardized gearboxes used in the petroleum industry. Most gearboxes include double-, or triple-reduction gearings, but chained reducers are used as well. The most widely used type is the double-reduction unit is presented in **Figure 3**, where the three shafts and two corresponding gear-pairs are shown. The prime mover drives the gearbox through a V-belt sheave, after the speed reduction the crank arm of the pumping unit is driven by the slow-speed shaft. (Takács, 2003) The most common tooth form is the herringbone due to their superior torque reversal tolerance, which usually happens in every pumping cycle. The gear reduction of gearboxes is around 30 to 1. The lubrication has key importance in protecting the moving parts of the gearbox, without a lubricant of the proper viscosity the lifetime of the gearbox significantly drops.

Figure 3 *A typical double-reduction gearbox used in sucker-rod pumping (Pidenergy, 2016)*

The most important parameter determining the lifetime of a gear reducer is the relationship between the torque rating of the unit and the torque loading during its operation. **Figure 4** illustrates the effect of overloading, showing that just a 10% increased torsional load compared to the rating can reduce the lifetime of the gearbox by half, a 20% overloading can result in only one-fifth of the lifetime specified by the manufacturer.

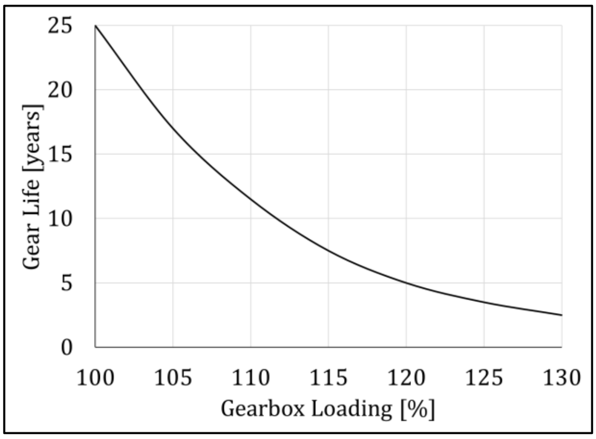


Figure 4 *The projected lifetime change of a gearbox due to overloading (Clegg, 2007), own edit*

A common problem due to overloading is pitting – a type of surface fatigue – when the stress on the surface of the gear tooth exceeds the limit of the material for periodic loading. These surface cavities can lead to gear tooth failures for overloaded gear reducers, according to the ANSI/AGMA 110.04, Nomenclature of Gear Tooth Failure Modes. (BakerHughes, 2018) Therefore, achieving optimal torque loading improves the lifetime of the most expensive part of the sucker-rod pumping installation. This can be achieved by using the appropriate counterbalancing as discussed in later chapters.

4.3 Pumping Unit Geometries

Different pumping unit geometries were developed to increase the efficiency of the petroleum production. In this chapter the four main geometries – Conventional, Air-Balanced, Mark II and Reverse Mark – are introduced in detail. Knowing the difference between the pumping unit geometries is essential to properly evaluate the dynamometer survey taken on one of these installations. For other geometry types the presented calculation method can be easily adapted. The dashed line – defining θ_p – in **Figure 5** through **Figure 8** is parallel with the link K. All figures representing the different geometries have the same scale (1:800) and are based on real API designations with 168 in nominal stroke length.

4.3.1 Conventional Pumping Unit

The conventional pumping unit – the oldest and most common sucker-rod pumping unit geometry – is based on the beam pumping unit first built in 1926 with the invention of crank counterbalance, which works with the same principle as the cable-tool drilling rig. The unit's popularity is based on its simple operation, low maintenance requirements and flexibility to cover a wide range of field applications. (Production Technology 1, 2018) The schematic layout is shown in **Figure 5**. The walking beam works like a double-arm lever that is driven at its rear end and drives the polished rod at its front. To counterbalance the unit, counterweights are placed on the crank arm to achieve a smoother torque loading of the gearbox. The unit can operate in both clockwise and counterclockwise direction of rotation.

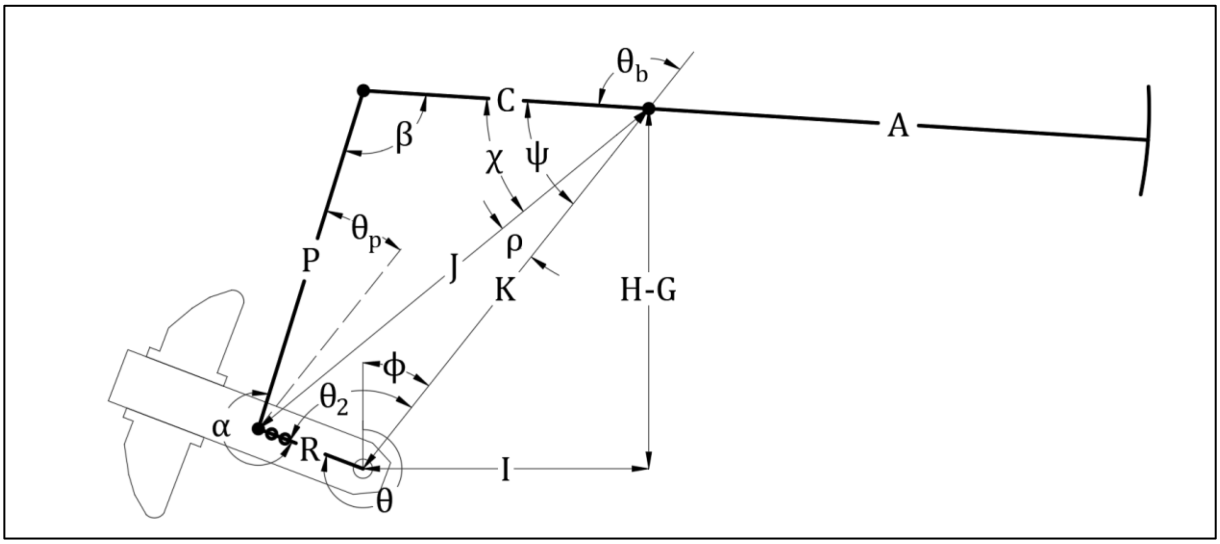


Figure 5 *The schematic layout of the conventional sucker-rod pumping unit*

4.3.2 Air Balanced Pumping Unit

The air balanced pumping units were developed in the 1920s. This configuration is similar to the Mark II in their linkage connections, but the crank arm is significantly smaller for the air balanced unit achieving the same stroke length, as seen in **Figure 6**.

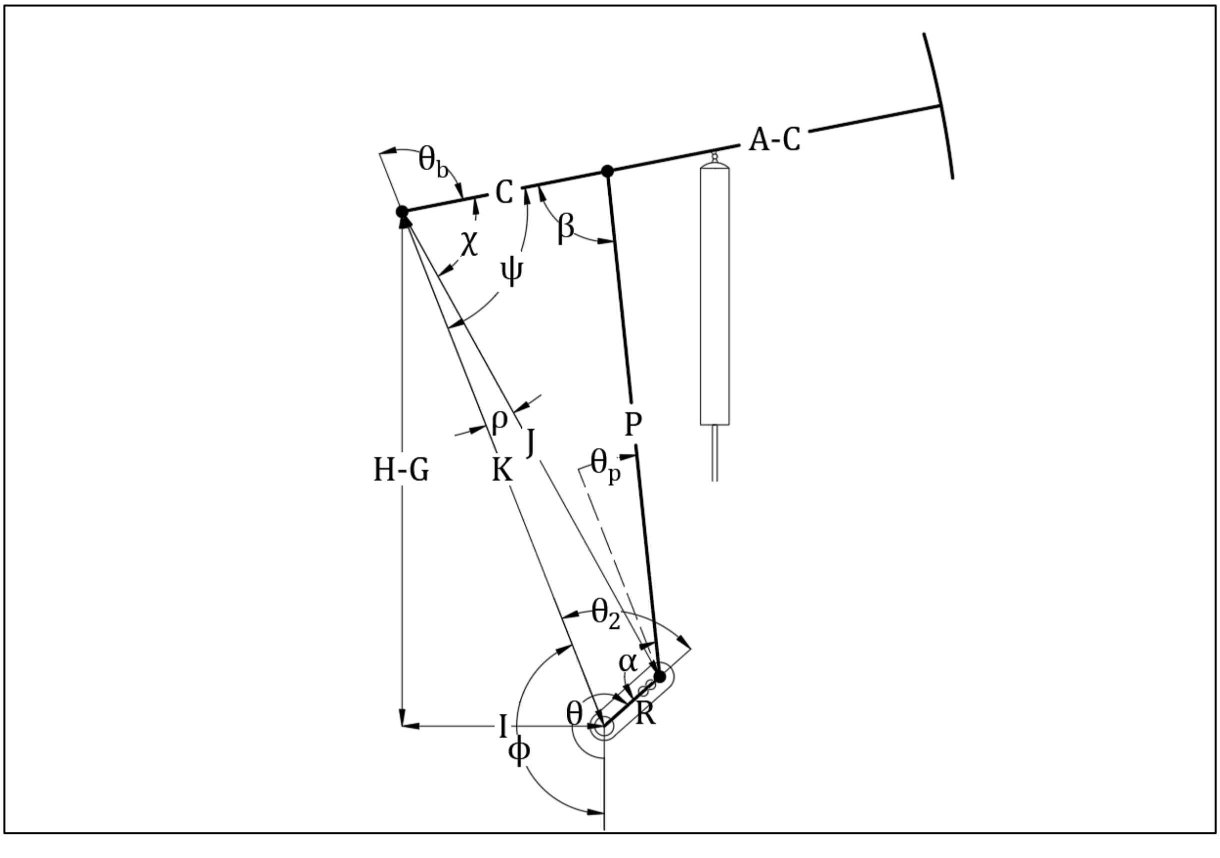


Figure 6 *The schematic layout of Air balanced sucker rod pumping unit*

The main difference between this and the other geometries is the counterbalancing method. The other investigated geometries use counterweights to even out the torque load on the gearbox, in this case a compressed-air cylinder is used to achieve the same. These units are way lighter due to the lack of heavy counterweights and are about 35% shorter than their conventional counterparts. (Takács, 2015) This sucker-rod pumping unit can be driven in both directions.

4.3.3 Mark II Pumping Unit

The Mark II sucker-rod pumping unit was invented by J. P. Byrd, it was patented in 1958 (Takács, 2015). The main objective of its development was to decrease the torque requirements, and consequently to decrease the power requirements of the operation compared to the conventional beam pumping units. Contrary to the conventional geometry, the walking beam works like a single-arm lever and it can only operate in the counterclockwise direction, shown in **Figure 7**.

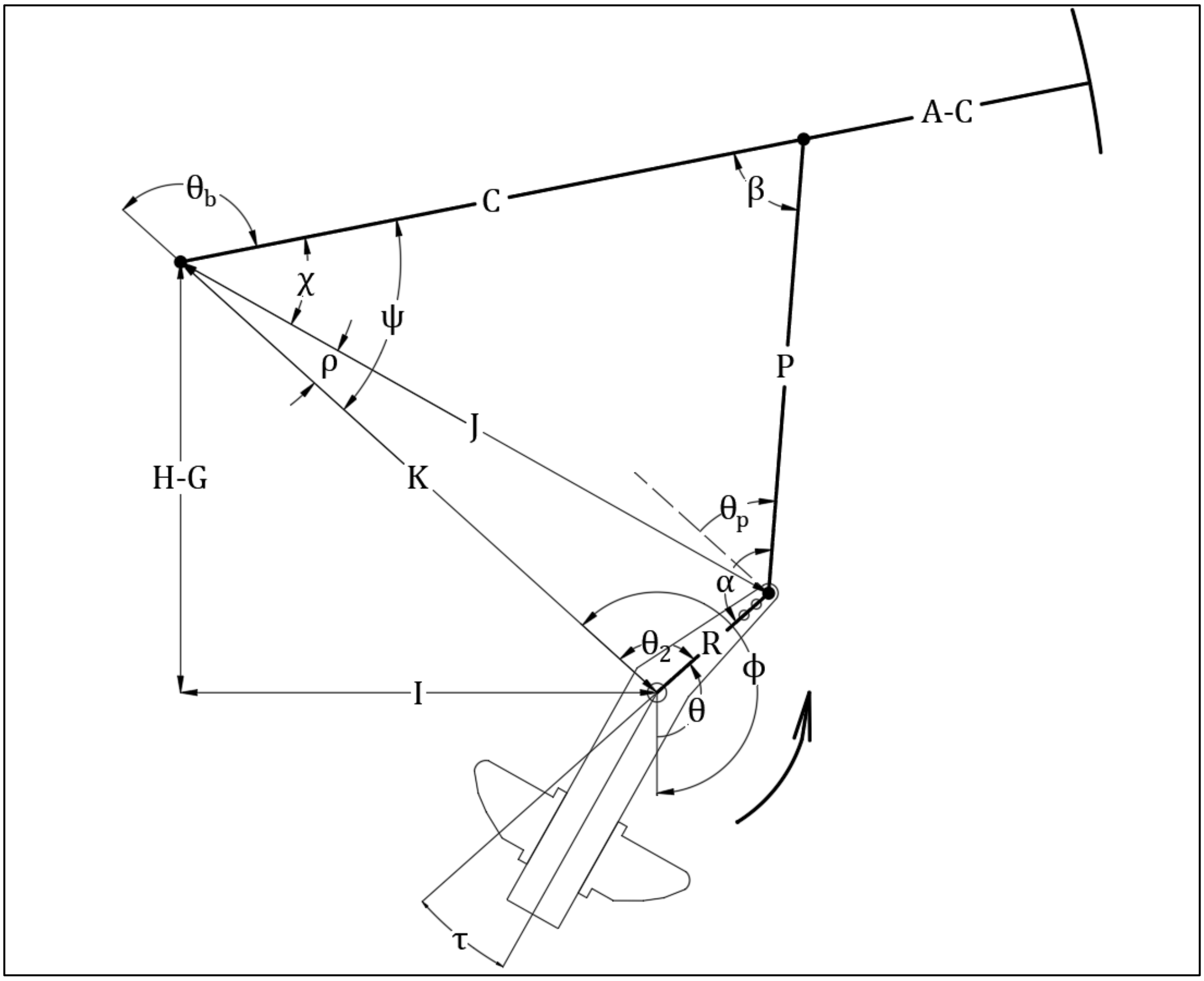


Figure 7 *The schematic layout of Mark II sucker rod pumping unit*

For the same pumping task, the Mark II unit will have a lower peak torque and a more uniform net gearbox torque distribution compared to an equivalent conventional pumping unit during the pumping cycle. (Production Technology 2, 2018) The rotary counterweights are placed on separate counterbalance arms that are directed opposite to the crank arm and are phased by τ, which is usually between 19° and 28°.

4.3.4 Reverse Mark Pumping Unit

The Reverse Mark – initially under the name TorqMaster – unit was developed in the 1980s by R. Gault, who analyzed the properties of already existing geometries, to combine all the good properties of the already existing geometries and to eliminate their disadvantages. (Takács, 2015) It was achieved by analyzing the previous geometries by computer and the results were incorporated in the design of the Reverse Mark unit. The schematic layout is shown in **Figure 8**.

At first, the Reverse Mark unit looks similar to the conventional geometry, the two main differences are the increased horizontal distance of the gearbox from the saddle bearing, and the phased counterweight placement on the crank arm. The maximum counterbalance moment is lagging behind the driven crank with a phase angle usually between 8-15°. By having a phase angle, the rotation of the unit is fixed in the clockwise direction, as shown in **Figure 8**. These modifications reduce the torque loading on the gearbox compared to the conventional unit while having the same operating conditions otherwise.

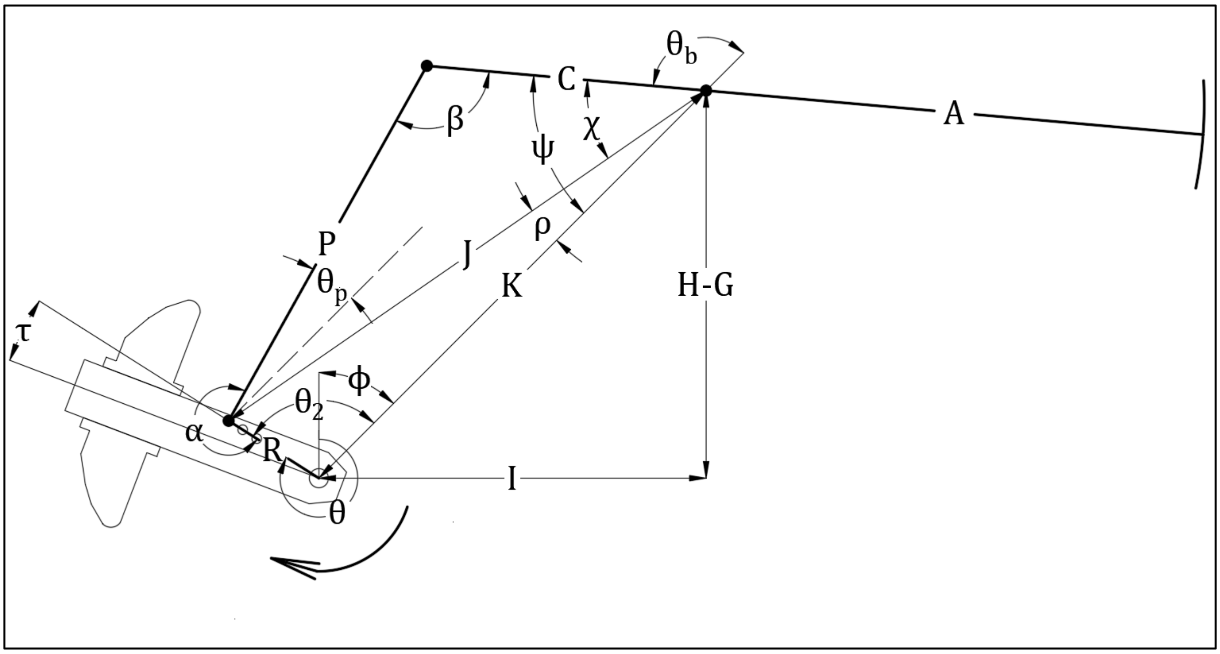


***Figure 8** The schematic layout of Reverse Mark sucker rod pumping unit*

5 Determination of the Net Gearbox Torque from Dynamometer Surveys

The complex interactions between the subsurface equipment, the produced liquid and the surface equipment during production make it nearly impossible to evaluate the operating condition of a sucker-rod pumping unit without measurement. The most widely used measurement technique is carried out by using an electronic dynamometer. The net mechanical gearbox torque can be determined by interpreting the dynamometer survey. The detailed solution of an example problem is presented in the thesis to illustrate the differences between the proposed evaluation method and the widely used TWM software; the relevant input data is given in **Table 1**. The variables used are consistent with the API Spec 11E (API, 2008).

Table 1 *Input data for the example problem*

Pumping unit designation	C-640D-365-168
Manufacturer	Lufkin
Geometry type	Conventional
Maximum torque loading of the gearbox	640,000 in lb
Maximum polished rod load	36,500 lb
Nominal stroke length	168 in
Structural unbalance	-1,500 lb
Crank type	94110CA
Gearbox mass moment of inertia	3,920 lb_m ft^2
Beam mass moment of inertia	1,047,183 lb_m ft^2
Rotation	Clockwise
Counterweights	4pcs. ORO, placed 10 in from long end of crank
Crank moment	470,810 in lb
Crank mass moment of inertia (2 cranks)	247,244 lb_m ft^2
Crank length	110 in
Crank half-width	11.5 in
Pumping speed	5.96 SPM

5.1 The Dynamometer Survey

Mechanical dynamometers were the first measurement equipment for sucker-rod pumping units. The mechanical dynamometers can only register the surface dynamometer card, which is a continuous plot of the polished rod load versus the polished rod displacement, whereas the new electronic devices measure both the polished rod load and polished rod position in time. **Figure 9** is the dynamometer card for the investigated pumping unit.

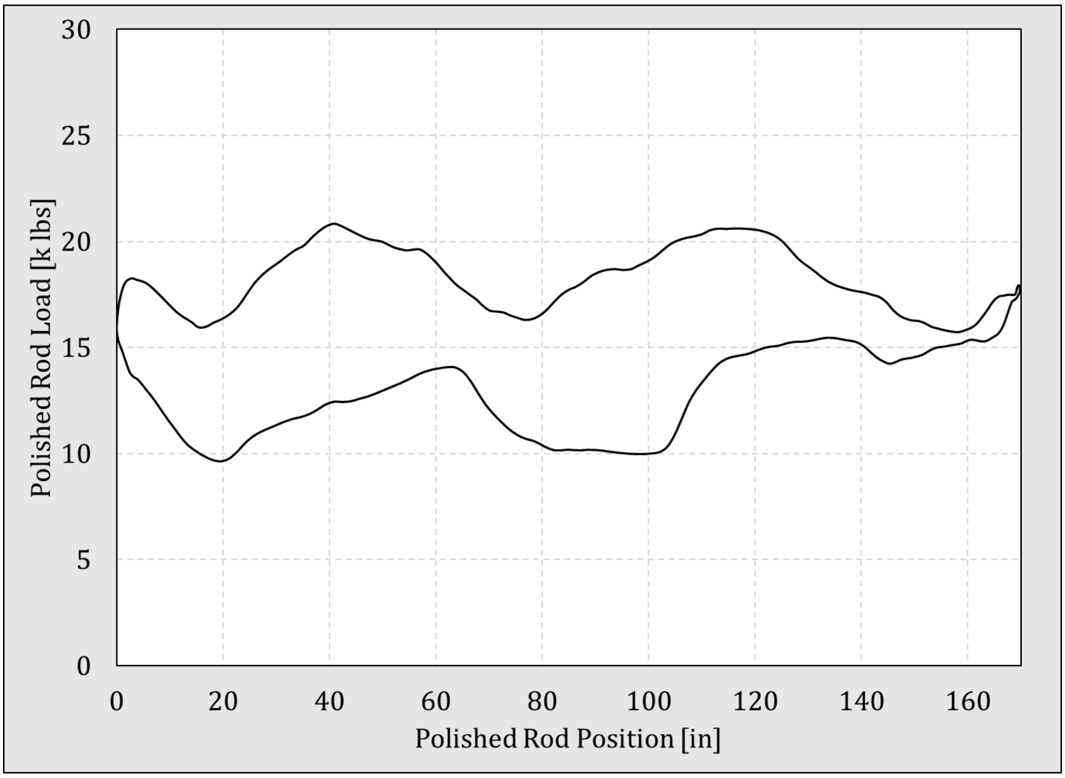


Figure 9 *The dynamometer card of the example problem*

The independent polished rod load and polished rod position functions in time are essential in an in-depth investigation of the pumping unit. Adequately determining the operating condition of a sucker-rod pumping unit can be carried out using a polished rod electronic dynamometer, or a polished rod transducer. **Figure 10** shows a horseshoe type electronic dynamometer and a rod transducer. The frequency of the data acquisition is usually greater than 20 Hz for modern electronic dynamometers; its value is 30 Hz for the example problem. 302 data points were registered in total for the investigated pumping cycle.

Figure 10 *A modern electronic horseshoe dynamometer and a polished rod transducer (Echometer, 2011)*

5.1.1 The Procedure of the Measurement

The dynamometer measurement is the easiest and most routinely used in the industry to obtain the required information for a complex torque analysis for sucker-rod pumping units. By installing the dynamometer between the polished rod clamp and the carrier bar it can record the load acting on the polished rod in time. During its normal operation, there is no space between the polished rod clamp and the carrier bar, see **Figure 11**.

The pumping unit must be stopped at the bottom of the stroke to begin the installation process by attaching a temporary rod clamp on the polished rod above the stuffing box. After restarting the pumping unit, a knock-off block is placed on the stuffing box, in downstroke the motor is shut down, and the brakes are activated when the unit reaches the bottom of the stroke. Due to this operation, the knock-off block will contact the previously installed temporary clamp releasing the load from the carrier bar. If the operation is carried out appropriately, there is enough space for the installation of the dynamometer, as seen in **Figure 11**.

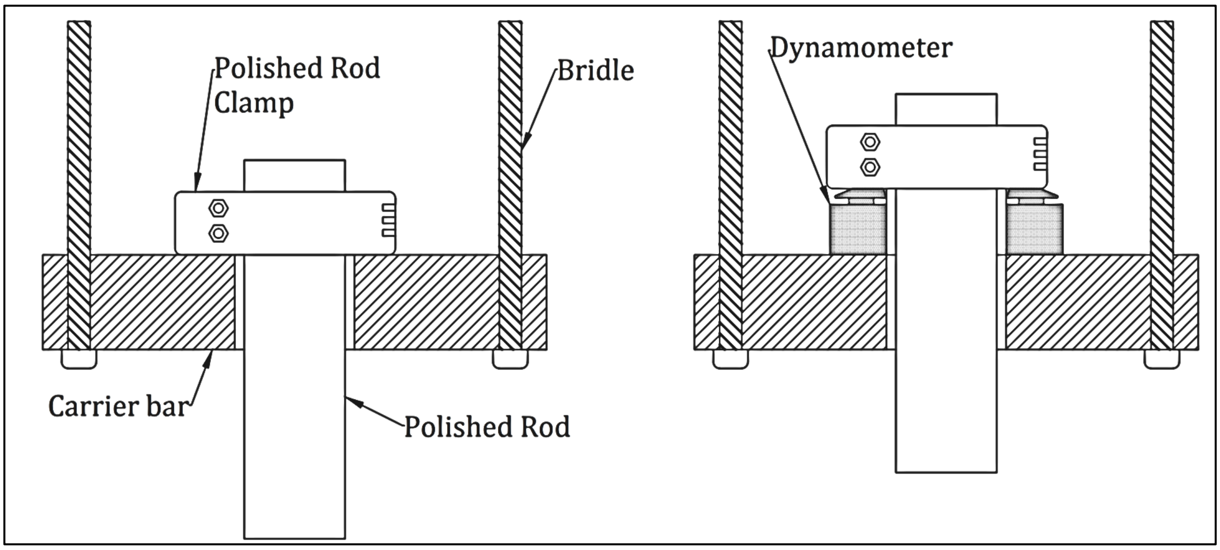


Figure 11 *Placement of the dynamometer (Echometer, 2011), own edit*

After restarting the unit and removing the knock-off, the loads in the polished rod will act on the dynamometer, making the measurement of the polished rod load possible. To measure the loads, the dynamometers usually use strain gauges. **Figure 12** shows the measured rod load variation in time for the example case, the measured polished rod positions are shown in **Figure 13**.

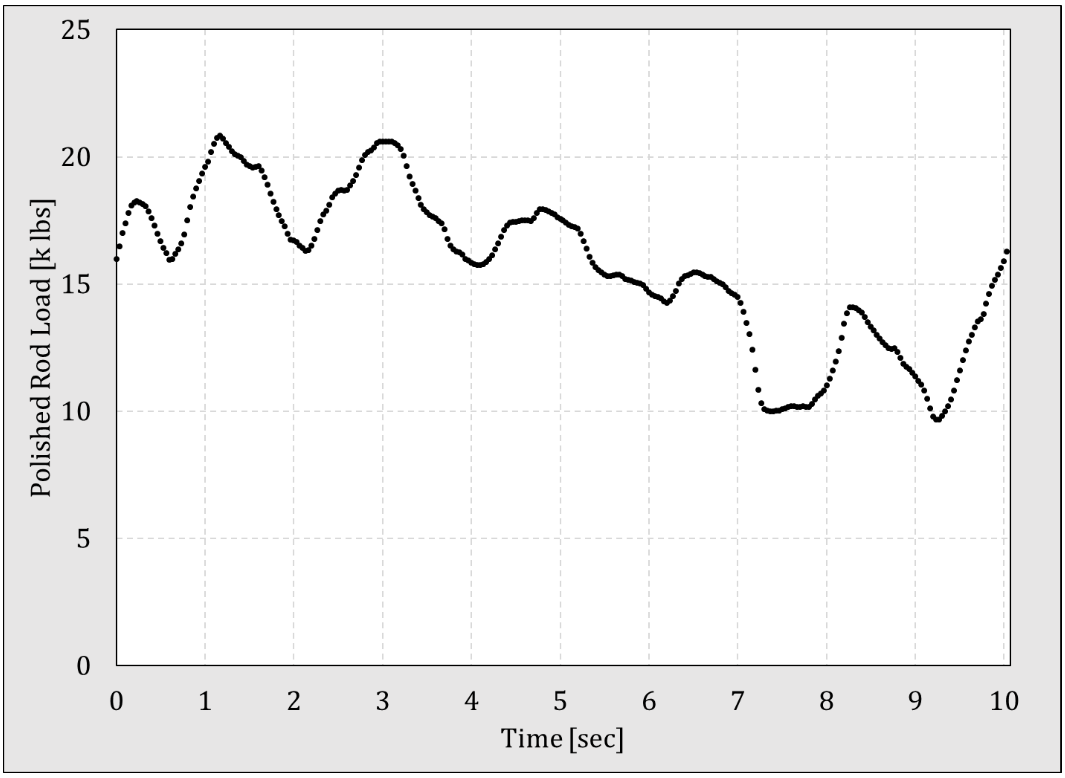


Figure 12 *Measured rod loads for the example problem*

For the position measurement usually data from a built-in accelerometer is used. The polished rod position values are determined by integrating the measured acceleration twice.

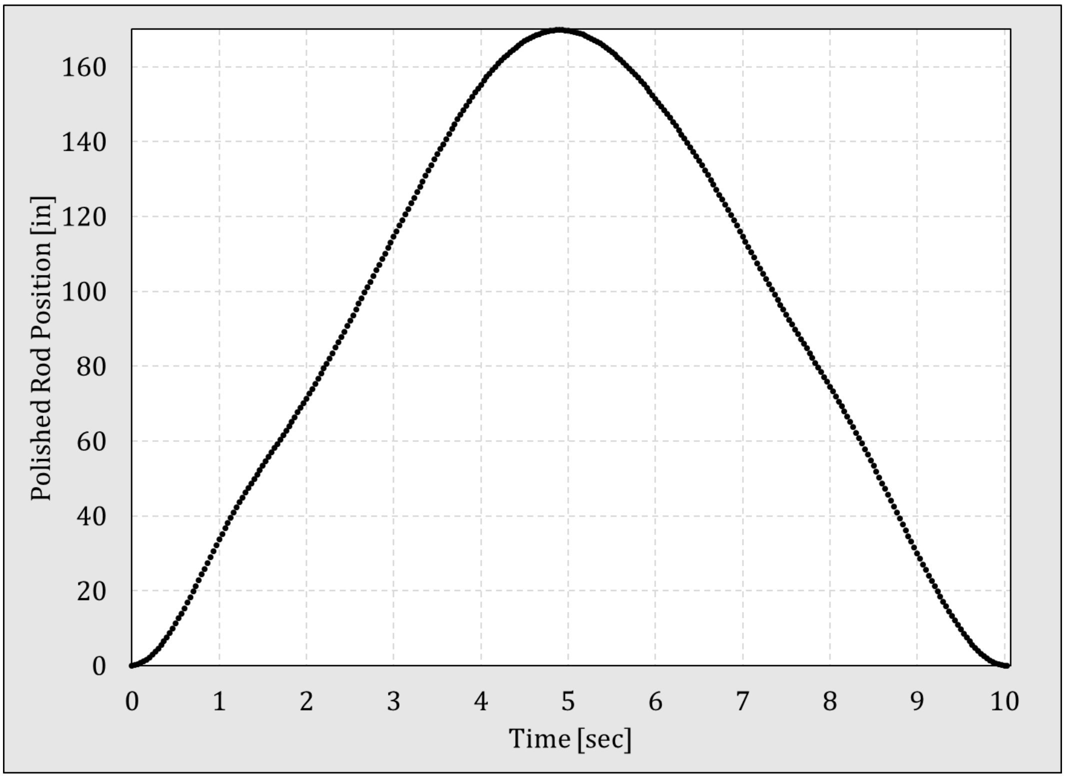


Figure 13 *Measured polished rod positions for the example problem*

At the start of the pumping the liquid level in the annulus will be at a higher position than the dynamic liquid level corresponding to the given pumping rate. No measurements must be done before the liquid level drops to its dynamic value. The time required to achieve the equilibrium liquid level depends on the inflow parameters of the well, the properties of the produced liquid, the configuration of the subsurface equipment and the type and operation of the surface elements of the sucker-rod pumping unit. The motion of the crank arm becomes periodic, when the operation of the pumping unit has been stabilized, so that the position of the dynamic liquid level is constant at the start of every upstroke.

The measurement with polished rod transducers is much simpler, it can be clamped under the carrier bar on the polished rod, but the provided accuracy is not sufficiently high for the complete torque analysis of the sucker-rod pumping unit.

5.2 Investigation of the Torque Loading of the Gearbox

There are two distinct cases in the calculation of gearbox torques based on the angular acceleration pattern of the crankshaft. The API Spec 11E (API, 2008) provides a calculation method for constant crankshaft velocities. But when the angular velocity of the crank changes more than 15% during the pumping cycle, the API method can lead to errors greater than 10%; this can result in operating decisions that overload the unit. As previously shown in **Figure 4**, the overloading drastically decreases the lifetime of the gearbox, therefore it is of paramount importance to determine the mechanical net gearbox torque adequately.

Having a non-zero crank angular acceleration is usually a consequence of using either a high-slip, or even an ultra-high-slip electric motor as the prime mover. In these cases, the crank angular velocity is a function of the torsional loading of the gearbox; at light loads the crank accelerates and achieves a higher speed, consequently at heavier loads it decelerates and slows down. This circumstance will produce a new torque component emerging in the calculation of the net gearbox torque calculations. In this case there are four different torque components acting on the gearbox of a sucker-rod pumping unit during its operating cycle. These torques are the rod torque, the counterbalance torque, the rotary moment of inertia and the articulating moment of inertia. The calculation of these torques requires the interpretation of a dynamometer survey. As a result of the analysis of the current operating condition, the net torque is determined throughout the pumping cycle by summing up the calculated torque components.

The basis of the torque analysis of sucker-rod pumping units is the knowledge of the crank angle variation in time throughout the pumping cycle. In this chapter the crank angles are assumed to be known, and the torque components acting on the slow-speed shaft are determined accordingly. The in-depth calculation of the crank angle function versus time is detailed in Chapter 5.3, the determination of the angular acceleration pattern of the crank arm and the walking beam are introduced in Chapter 5.4 and Chapter 5.5, respectively.

Unlike in previous works, the variation of every angle calculated from the measured polished rod positions are presented in time, not as a function of the crank angle. This is also true for the angular velocities and angular accelerations computed by the newly proposed methods. To determine the aforementioned angles, the knowledge of the crank angle is required, which is not necessarily changing linearly in time, as assumed in prior works.

5.2.1 Flowchart of the Torque Calculation Procedure

As previously discussed, four different torque components must be determined to find the mechanical net gearbox torque. All torque components can be calculated by interpreting the dynamometer survey. The simplified flowchart representing the calculation of the torque components from the dynamometer survey is shown in **Figure 14**. The in-depth determination of these torque functions in time is shown in Chapter 5.2.2 through Chapter 5.2.4.

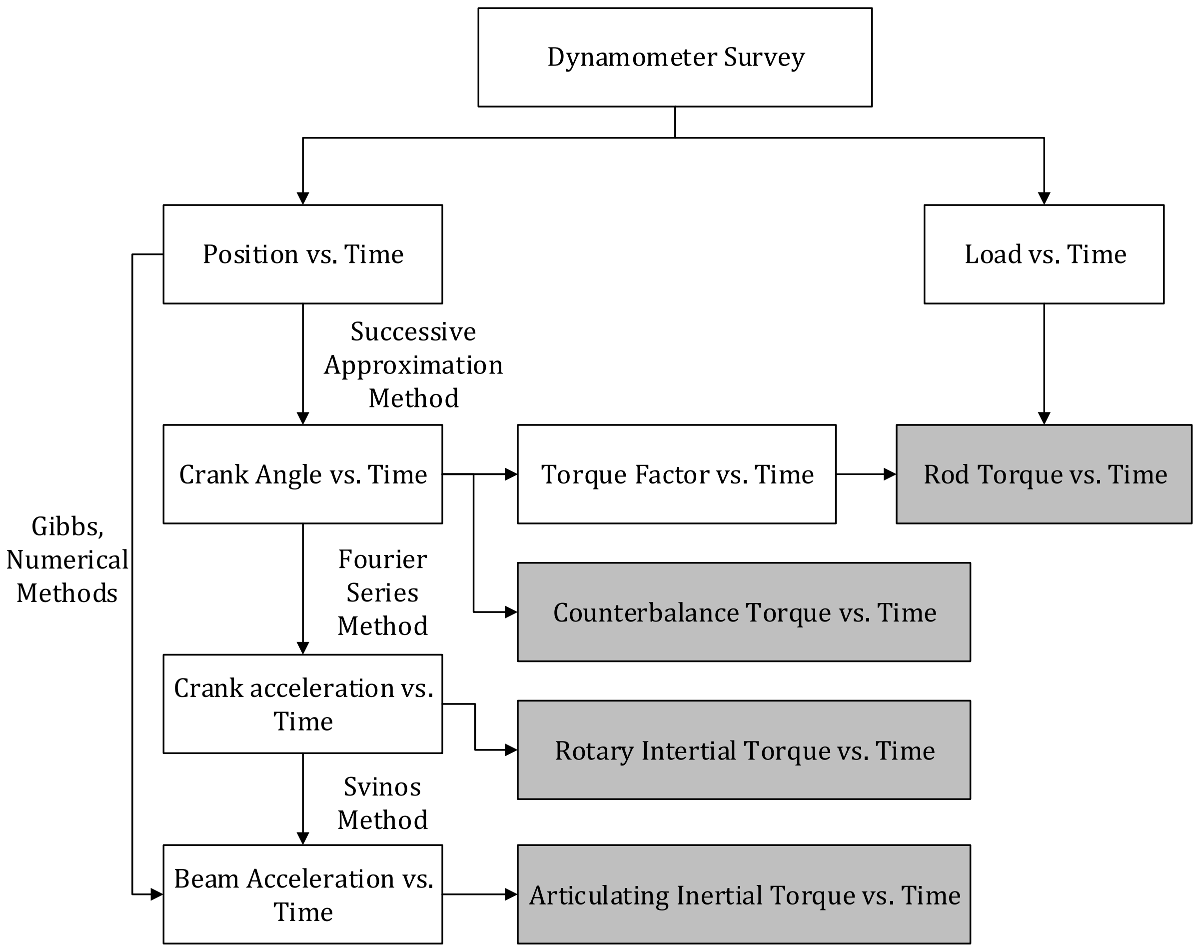


Figure 14 *Simplified flowchart of the determination of every torque component*

5.2.2 Rod Torque

The rod torque is required to overcome the sum of the weight of the rod string and the produced liquid, the frictional losses, and the dynamic losses during production. The formula which determines the rod torque is given in Equation 1. (Takács, 2015)

$$T_{Rod}(t) = TF(t) \cdot (F(t) - SU) \tag{1}$$

where:

$T_{Rod}(t)$	Rod torque in time [in lb],
$TF(t)$	Torque factor in time [in],
$F(t)$	Polished rod load in time [lb], and
SU	Structural unbalance [lb].

The structural unbalance is the force requirement to balance the walking beam horizontally with disconnected pitmans from the cranks. A sucker-rod pumping unit can be tail heavy – if a downward pointing force must be exerted on the horsehead side to maintain the balance – or horsehead heavy in the opposite case. (Takács, 2003) The rod torque calculated for the example problem is shown in **Figure 15** with the results from TWM.

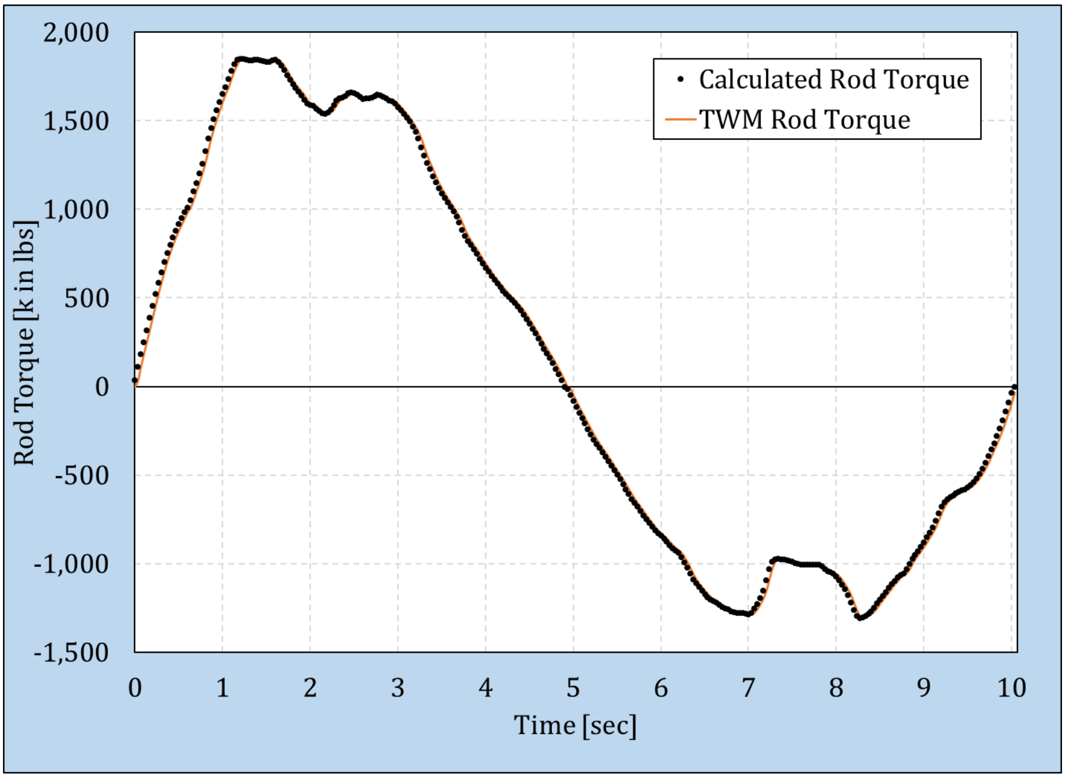


Figure 15 *Calculated rod torque for the example problem*

The value of the structural unbalance is considered positive when it is pointing downwards, therefore it depends on the rotation of the pumping unit. The structural unbalance is given for every pumping unit by the manufacturer.

5.2.2.1 Torque Factor

For the calculation of the rod torque the knowledge of the torque factor – the imaginary lever arm – throughout the pumping cycle is required, which is calculated from the crank angles using the geometry type and the linkage lengths of the pumping unit. In Equation 1 the polished rod loads are obtained directly from the dynamometer survey; the structural unbalance is provided by the manufacturer. The objective is to determine the torque factor as a function of time for the calculation of the rod torque, which is not included in the dynamometer measurement. The torque factor at a given time can be calculated using Equation 2. Both the torque factor and the auxiliary angles used depend on the crank angle, which was the basis of the previous torque analysis methods. If the crank angle variation in time is known, the change of these variables in time can be considered. **Figure 16** shows the calculated torque factor values for the example problem.

$$TF = \frac{R \cdot A}{C} \frac{sin(\alpha)}{sin(\beta)} \tag{2}$$

where:

TF Torque factor [in],

R, A, C Linkage dimensions [in], and

α, β Auxiliary angles defined in **Table 2** [rad].

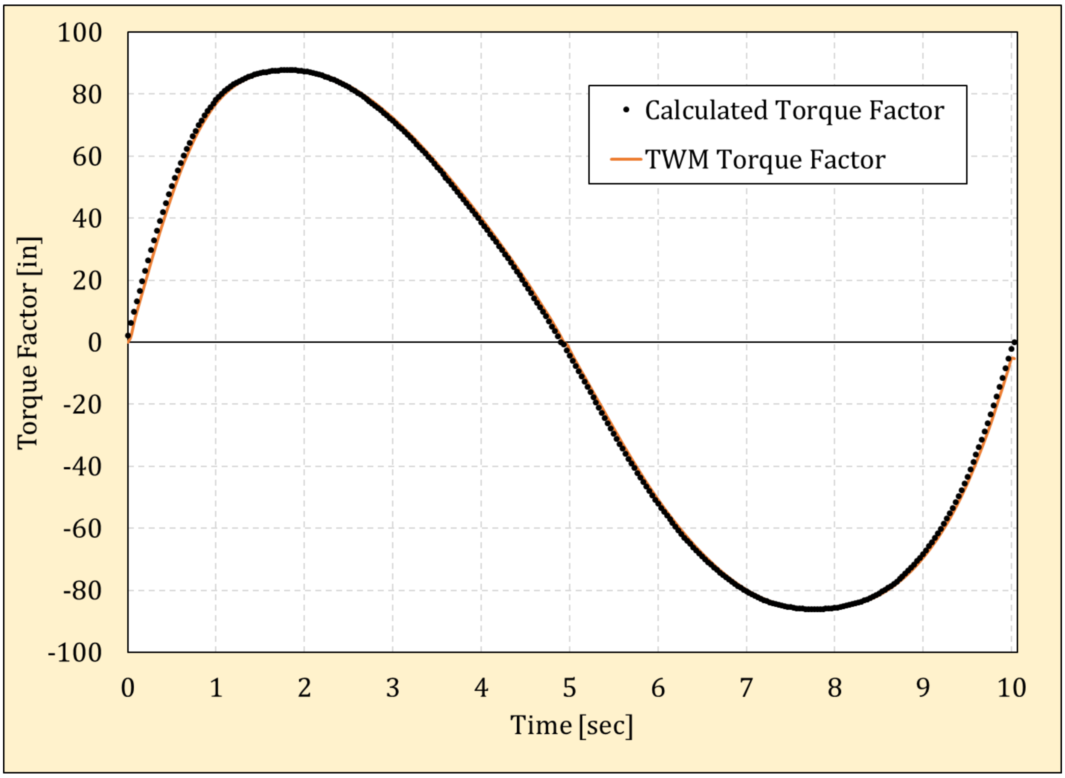


***Figure 16** Torque factors calculated for the example problem*

The angles on the right side of Equation 2 depend on the crank angle as seen in **Figure 5** through **Figure 8**; therefore, the crank angle has to be calculated first in order to determine the torque factor at a given position of rods. Once the crank angle, θ, is found, the corresponding α and β angles are found using the equations in **Table 2.** (API, 2008) (Takács, 2015)

***Table 2** Formulae used in the calculation of the torque factor*

<table>
<tr><th>Conventional and Reverse Mark</th><th>Mark II</th><th>Air Balanced</th></tr>
<tr><td colspan="3">$\theta_2 = 2\pi - \theta + \phi$</td></tr>
<tr><td colspan="3">$\beta = cos^{-1}\left(\frac{C^2 + P^2 - R^2 - K^2 + 2 \cdot K \cdot R \cdot \cos(\theta_2)}{2 \cdot C \cdot P}\right)$</td></tr>
<tr><td colspan="3">$J = \sqrt{R^2 + K^2 - 2 \cdot K \cdot R \cdot \cos(\theta_2)}$</td></tr>
<tr><td>$\rho = cos^{-1}\left(\frac{J^2 + K^2 - R^2}{2 \cdot J \cdot K}\right) \cdot b$</td><td colspan="2">$\rho = sin^{-1}\left(\frac{R}{J} \cdot sin(\theta_2)\right)$</td></tr>
<tr><td>$\chi = cos^{-1}\left(\frac{J^2 + C^2 - P^2}{2 \cdot J \cdot C}\right)$</td><td colspan="2">$\chi = sin^{-1}\left(\frac{P}{J} \cdot sin(\beta)\right)$</td></tr>
<tr><td>$\psi = \chi - \rho$</td><td colspan="2">$\psi = \chi + \rho$</td></tr>
</table>

$\alpha = \beta + \psi - (\theta - \phi)$	$\alpha = \theta - \phi - (\beta + \psi)$	$\alpha = \beta + \psi + (\theta - \phi)$

The parameter b in **Table 2** is defined by Equation 3. The calculated torque factors are shown in Figure 16 along with the data from the Total Well Management software.

$$b = \begin{cases} -1 \; if \; 0 < \theta_2 \leq \pi \\ 1 \; if \; \pi < \theta_2 \leq 2\pi \end{cases} \tag{3}$$

5.2.3 Counterbalance Torque

The load difference on the polished rod between the upstroke and the downstroke necessitates the utilization of counterbalancing, to achieve a possibly smooth torque loading during the pumping cycle. On crank balanced sucker-rod pumping units it is achieved by installing counterweights on the crank arms. On the main counterweights auxiliary weights can be placed. On beam balanced units the counterweights are placed on the end of the walking beam. On air balanced units the counterbalancing is achieved by installing a compressed air cylinder to the walking beam between the horsehead and the saddle bearing. Since the beam balanced units are generally much smaller and produce only a tiny fraction compared to a crank balanced one, the counterbalancing of these units is not detailed. The detailed description of counterbalancing of air balanced pumping units are omitted because it can be found in the literature in detail. (API, 2008)

5.2.3.1 Crank Balanced Pumping Units

The placement of the main counterweights on the crank arm is shown in **Figure 17.** The travel (T), the maximum distance (M), and the vertical component of the center of gravity (Y_{CW}) depend on the type of the counterweight used.

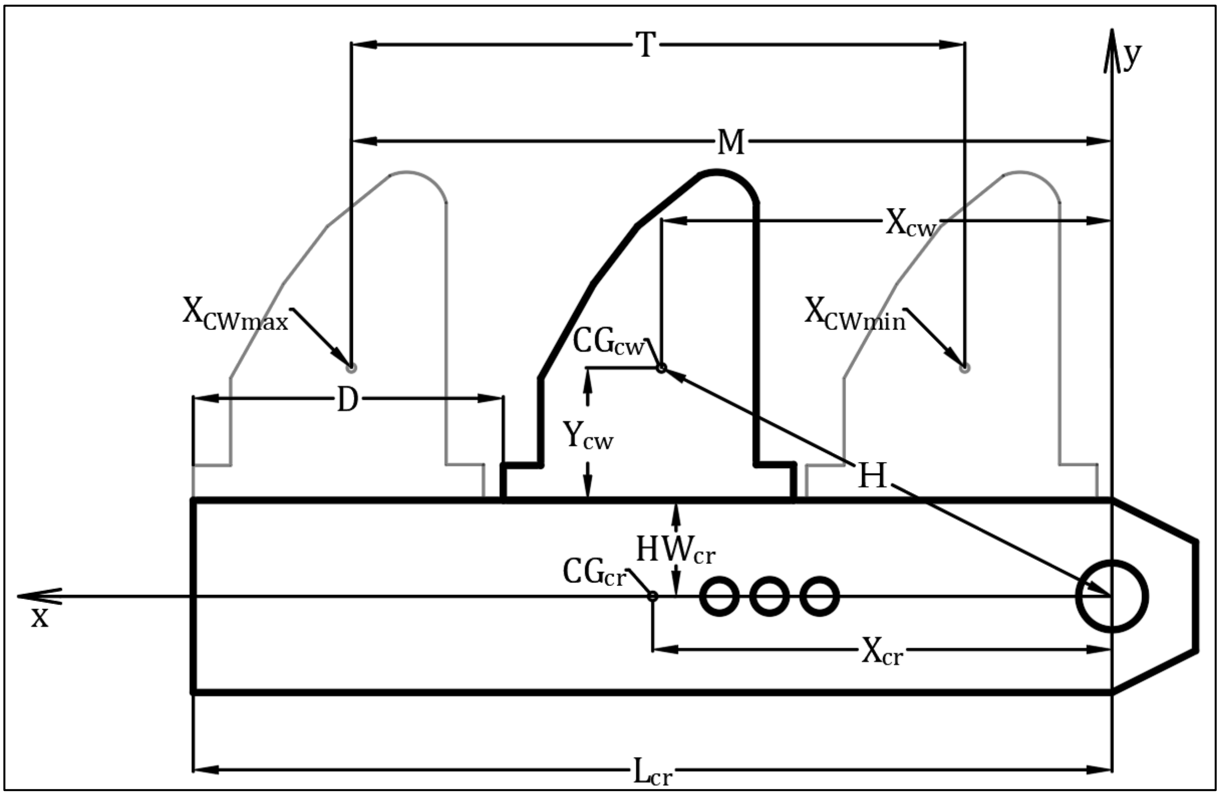


Figure 17 *Counterweight placement on the crank arm (Takács, 2015), own edit*

The list of applicable counterweights depends on the crank arm installed on the pumping unit. **Table 3** contains the compatible counterweights for the 94110CA crank arm of the investigated C-640D-365-168 pumping unit. The counterweights' masses and mass moments of inertia about their center of gravity is included. **Table 3** also includes the compatible auxiliary counterweights – highlighted with gray color – with their relevant properties. These parameters are usually given by the manufacturer.

Table 3 *The relevant properties of the compatible counterweights and auxiliary weights to crank 94110C (Lufkin, 1997)*

Index	CW. Type	Mass [lb]	I_{CG} [lb ft^2]	Y [in]	M [in]	T [in]
1	7RO	315	114	8.6	96.3	84.58
	7S	141	51			
2	6RO	504	229	9.9	94.65	81.58
	6S	190	83			
3	5CRO	662	430	11.8	93.1	78.84
	5CS	327	220			
4	5ARO	913	707	13.4	91.91	77.34
	5S	366	272			
5	3CRO	1,327	1,384	13.3	87.4	83.96
	3BS	572	562			
6	2RO	1,708	2,458	14.2	84.34	80.84
	2S	612	756			
7	1RO	2,075	3,478	15.4	83.4	79.84
	1S	638	1,222			
8	OARO	2,700	5,268	18.5	82.4	78.84
	OAS	836	1,505			
9	ORO	3,397	8,017	19	77.4	73.77
	OS	1,128	2,290			
10	OORO	3,894	9,960	20	77.4	63.77
	OOS	1,175	2,490			

In **Table 3** the maximum distance of the specific counterweight's center of gravity from the long end of the crank is provided with the maximum travel distance of the counterweight on the crank arm, the length T, see **Figure 17**. On the same crank the smaller counterweights' center of gravity can be placed further from the crankshaft, and they have a longer travel distance as well. Using the same counterweight on smaller cranks the maximum distance and the travel of the counterweight are shorter.

For cases when the mass moment of inertia is unknown for a specific counterweight, I have developed Equation 4 based on the data in **Table 3** to find an approximate value from its mass. To find the best parabolic function possible the least squares method was used; the equation proposed has 98.07% accuracy based on the input data listed in **Table 3**. Since only the counterweight masses are listed in (BakerHughes, 2018), this formula can be used in this case to provide reasonably good approximation.

$$I_{CG_a} = 4.052 \cdot 10^{-4} \cdot w^2 + 0.9734 \cdot w - 68.032 \quad (4)$$

where:

I_{CG_a}	Approximate counterweight mass moment of inertia about its center of gravity [lb_m ft^2], and
w	Mass of the counterweight [lb_m].

5.2.3.2 Using Identical Counterweights on the Crank Arms

The counterbalance torque calculation is based on the calculated crank angle variation in time. The counterbalance torque versus time function is described by Equation 5, if the same main and auxiliary counterweights are used on the opposing sides of the crank arms. The maximum counterbalance moment in Equation 5 can be determined from the moment of the crank arms of the sucker-rod pumping unit and the knowledge of the configuration of the applied counterweights on the cranks.

$$T_{CB}(t) = -T_{CB_{max}} \cdot sin(\theta(t) + \tau) \quad (5)$$

where:

$T_{CB}(t)$	Counterbalance torque in time [in lb],
$T_{CB_{max}}$	Maximum counterbalance moment [in lb],
$\theta(t)$	Crank angle variation in time [rad], and
τ	Phase angle [rad].

When two identical counterweights are used on a crank, the combined center of gravity for the crank arm and the counterweights – the only purely rotating components that create the counterbalance torque – is aligned on the symmetry line of the crank arm. The value of the phase shift – τ – is zero for Conventional pumping units; it is specified by the manufacturer for the Mark II and Reverse Mark pumping units. If the counterweights on both sides of the crank arms are of the same type and are placed at the same distance from the end of the crank, the maximum counterbalance moment is calculated using Equation 6. (Bommer & Podio, 2012)

$$T_{CB_{max}} = T_{crank} + (M - D) \cdot (n \cdot w + n_a \cdot w_a) \quad (6)$$

where:

T_{crank}	Crank moment [in lb],
M	Maximum lever arm of the counterweights [in],
D	Counterweight distance from the long end of the crank [in],
n	Total number of main counterweights [-],

w	Weight of the one main counterweight [lb],
n_a	Total number of auxiliary weights [-], and
w_a	Weight of one auxiliary weight [lb].

Since in the example problem all counterweights are the same, and their positions from the long end of the crank are also equal, Equation 6 can be used to find the maximum counterbalance moment, and Equation 5 produces the counterbalance torque function throughout the pumping cycle. The maximum counterbalance moment for the example case is found to be 1,386 k in lbs. The variation of the counterbalance torque for the example problem is shown in **Figure 18**. It does not have a perfectly sinusoidal shape because the crank angle values are not changing linearly with time, the crank does not turn at constant speed during the pumping cycle.

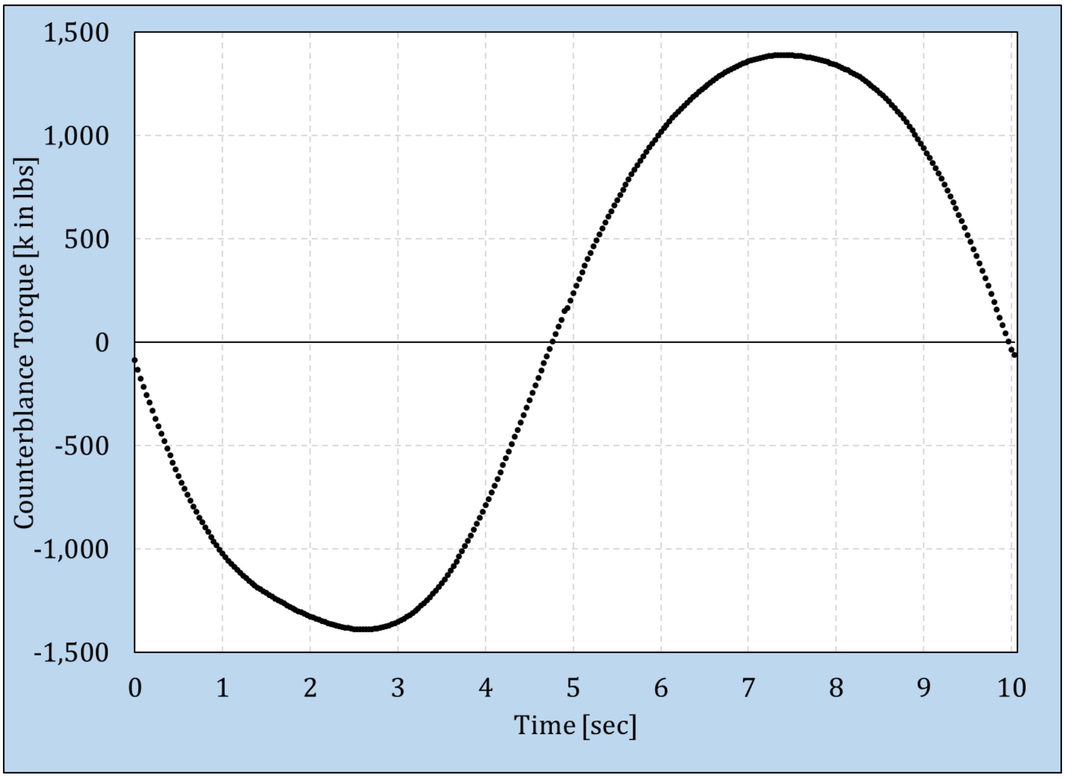


Figure 18 *Calculated counterbalance torque for the example problem*

The counterweights can be placed at different distances from the end of the crank arm, the vertical component of the center of gravity for the aforementioned system is unchanged, only the magnitude of the counterbalance torque will be different. This phenomenon is illustrated in **Figure 19**. T_{cbmax_1} refers to the topmost case illustrated in the right portion of the figure. Equation 7 is used to determine the maximum counterbalance moment accurately in the case of having identical counterweights at different positions on the cranks.

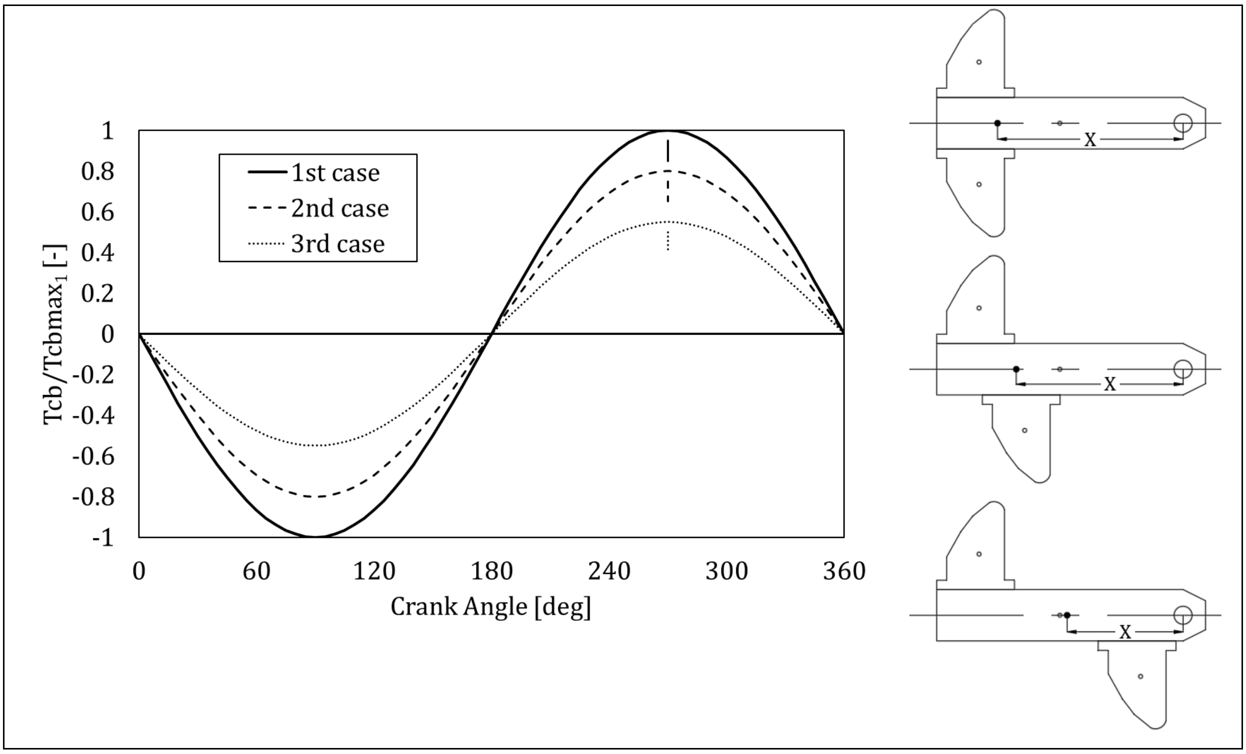


***Figure 19** Effect of differently positioned identical counterweights on the counterbalance torque function*

$$T_{CB_{max}} = T_{crank} + \sum_{i=1}^{n} \big((M - D_i) \cdot (w + n_a \cdot w_a)\big) \tag{7}$$

where:

T_{crank}	Crank moment [in lb],
n	Total number of main counterweights [-],
M	Maximum lever arm for the counterweights [in],
D_i	i^{th} counterweight distance from the long end of the crank [in],
w	Weight of one main counterweight [lb],
n_a	Number of auxiliary weights on one main counterweight [-], and
w_a	Weight of one auxiliary weight [lb].

5.2.3.3 Using Different Counterweights on the Crank Arms

The asymmetrical counterweight configuration means that on at least one crank arm different counterweights are used on its opposing sides. In the production practice the most common case for this type of counterbalancing occurs when only one main counterweight is used on one crank arm, but they are placed on different sides of the crank arm. In this case the counterbalance torque is exactly half compared to using four main counterweights.

(BakerHughes, 2018) specifically cautions the user to place only one counterweight on the same side of the cranks as shown in **Figure 20** if two counterweights are used. In this case the maximum counterbalance moment is in phase with the symmetry line of the crank arm, similarly to the symmetrical counterbalancing scenario. By having the counterweights on the same sides of the crank arms, a phase angle is introduced that shifts the counterbalance torque. It is important to state that this installation and operations manual was created in 2018 and it only refers to the possibility of overloading without an in-depth analysis or explanation. Note that for some pumping units, this phase angle can help to create a better net torque loading, but this must be determined strictly on case-by-case basis. The torque calculation model presented here can determine how this way of counterbalancing will act on the mechanical net gearbox torque function in time.

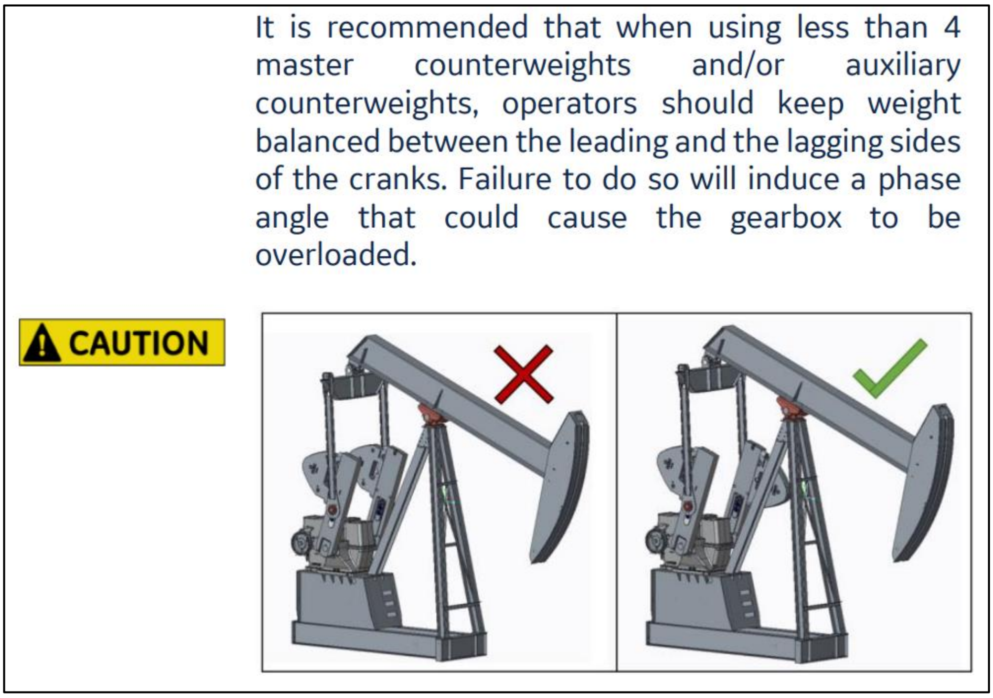


Figure 20 *Caution against placing the counterweights on the same side of the crank arms (BakerHughes, 2018)*

Asymmetrical counterbalancing occurs, when different main and auxiliary counterweights are used on one crank arm, or when only one counterweight is applied to the same side of the crank arm. These cases will not only change the amplitude of the counterbalance torque; an additional phase angle is introduced to the counterbalance torque versus time function. Equation 8 describes the calculation of counterbalance torque for the asymmetrically placed counterweights case.

$$T_{CB}(t) = -T_{CBMax} \cdot sin(\theta(t) + \tau + \tau') \tag{8}$$

where:

T_{CBMax}	Maximum counterbalance moment [in lb],
$\theta(t)$	Crank angle variation in time [rad],
τ	Phase angle [rad], and
τ'	Secondary phase angle [rad].

I have developed Equation 9, that defines the maximum counterbalance moment for asymmetrically placed counterweight configurations. This equation is the generalized form of Equation (7). With this new equation the maximum counterbalance moment can be determined for any counterweight configuration.

$$T_{CB_{max}} = T_{crank} + \sum_{i=1}^{n} \left((M_i - D_i) \cdot \left(w_i + \sum_{j=1}^{n_{a_i}} w_{a_{i_j}} \right) \right) \tag{9}$$

where:

T_{crank}	Crank moment [in lb],
n	Total number of counterweights [-],
M_i	Maximum lever arm for the i^{th} counterweight [in],
D_i	i^{th} counterweight distance from the long end of the crank [in],
w_i	Weight of the i^{th} counterweight [lb],
n_{a_i}	Number of auxiliary weights on the i^{th} counterweight [-], and
$w_{a_{i_j}}$	Weight of the j^{th} auxiliary weight on the i^{th} counterweight [lb].

Figure 21 illustrates the connection between the changes in the counterweight configurations and the resulting counterbalance torque functions for three sample cases. As shown, the combined center of gravity of the crank and counterweight system produces the evolution of a secondary phase angle.

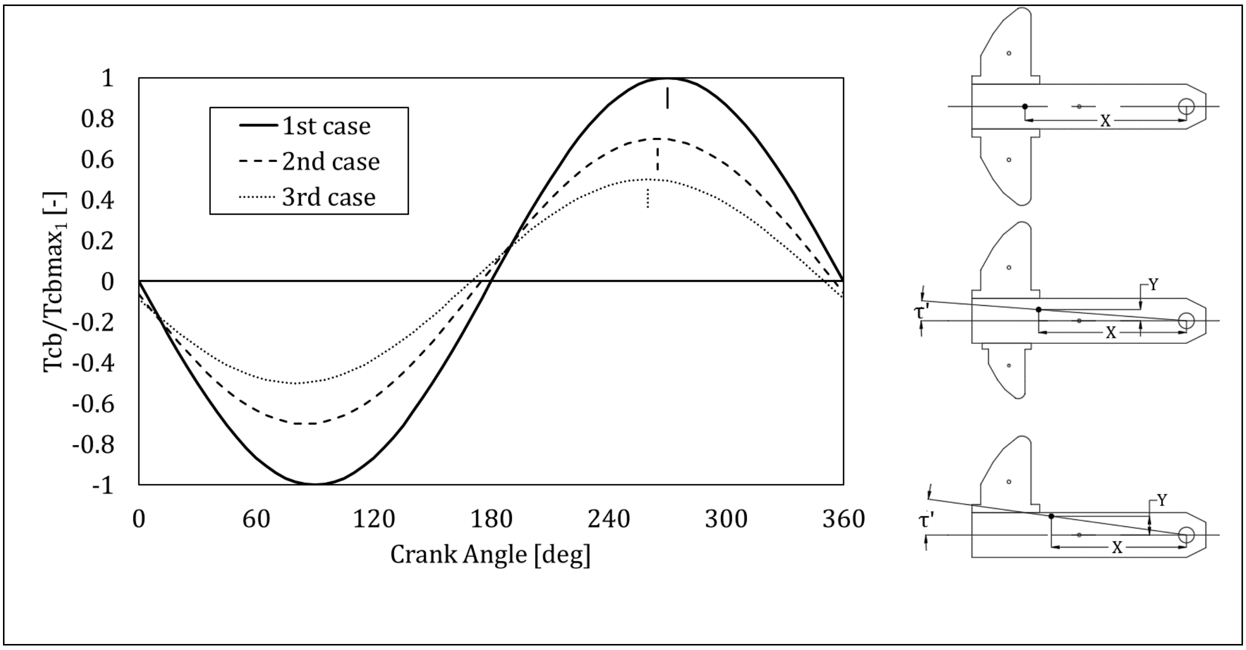


Figure 21 *Effect of different asymmetrical counterweight configurations on the counterbalance torque function*

The secondary phase angle – τ' – represents the lead or lag of the maximum counterbalance torque from the symmetry line of the crank arm, as shown in **Figure 21**. This value can be positive and negative, depending on the counterweight configuration and the direction of rotation. To calculate this angle, the center of gravity for the system containing the crank arm and the counterweights must be determined.

Knowing the vertical and horizontal distance of the center of gravity of the aforementioned system from the crankshaft, the secondary phase angle can be found using Equation 10, see **Figure 21**.

$$\tau' = tan^{-1}\left(\frac{Y}{X}\right) \tag{10}$$

where:

Y	Vertical distance of the center of gravity of the system containing the crank and the counterweights from the crankshaft [in], and
X	Horizontal distance of the center of gravity of the system containing the crank and the counterweights from the crankshaft [in].

To find the center of gravity of this system, the required data are the mass of the counterweights and the crank arm, the horizontal and vertical distance of their centers of gravity from the crankshaft, as defined by Equation 11 and Equation 12, respectively. The coordinate system used to describe the geometrical parameters used in these equations is illustrated in **Figure 17.** The value of Y_{cw_i} is positive if the counterweight precedes the crank arm in the direction of rotation, and is negative if it is on the opposite side of the crank arm. The auxiliary counterweights installed on the main counterweights are assumed to have the same center of gravity, as the main counterweight in Equation 11 and Equation 12.

$$X = \frac{X_{cr} \cdot m_{cr} + \sum_{i=1}^{n}\left(X_{cw_i} \cdot \left(m_{cw_i} + \sum_{j=1}^{n_{a_i}} m_{cw_{a_{ij}}}\right)\right)}{m_{cr} + \sum_{i=1}^{n}\left(m_{cw_i} + \sum_{j=1}^{n_{a_i}} m_{cw_{a_{ij}}}\right)} \tag{11}$$

where:

X_{cr}	Horizontal distance of the center of gravity of the crank from the crankshaft [in],
m_{cr}	Mass of the crank arm [lb_m],
X_{cw_i}	Horizontal distance of the center of gravity of the i^{th} counterweight from the crankshaft [in],
m_{cw_i}	Mass of the i^{th} counterweight [lb_m], and
$m_{cw_{a_{ij}}}$	Mass of the j^{th} auxiliary weight on the i^{th} counterweight [lb_m].

$$Y = \frac{\sum_{i=1}^{n}\left((Y_{cw_i} + HW_{cr}) \cdot \left(m_{cw_i} + \sum_{j=1}^{n_{a_i}} m_{cw_{a_{ij}}}\right)\right)}{m_{cr} + \sum_{i=1}^{n}\left(m_{cw_i} + \sum_{j=1}^{n_{a_i}} m_{cw_{a_{ij}}}\right)} \tag{12}$$

where:

Y_{cw_i}	Vertical distance of the center of gravity of the i^{th} counterweight from its base [in].

The mass for every counterweight is given by the manufacturer, but the mass of the crank arm is not always known. Some manufacturers publish the mass of the gearbox and the two cranks combined, helping the installation procedure of the pumping unit, but the individual mass of the crank is usually unspecified. (BakerHughes, 2018) If the mass of the crank must be approximated, I have developed Equation 13 to provide a reasonable value for the calculation based on the equation used in (Serway, 1986). Equation 13 assumes the crank arm to have a perfectly cuboid shape and its center of rotation is taken at the middle point of its shorter side closest to the crankshaft.

$$m_{crank_a} = \frac{12 \cdot \frac{I_{cr}}{2}}{\left(2 \cdot \frac{X_{cr}}{12}\right)^2 + 4 \cdot \left(\frac{HW_{cr}}{12}\right)^2} \tag{13}$$

where:

m_{crank_a}	Approximate mass of the crank [lb],
I_{cr}	Mass moment of inertia of the cranks [lb ft^2],
X_{cr}	Length of the crank arm [in], and
HW_{cr}	Half-width of the crank arm [in].

The approximate mass of one crank for the example problem is 4,366 lb, which is comparable with a value provided by a different manufacturer for a unit with the same designation. (Schlumberger, 2019) provides 4,699 lb crank mass for their C-640D-365-168 sucker-rod pumping unit. This comparison validates the applicability of Equation (13) for the example problem.

5.2.4 Inertial Torques

The inertial torques are results of the energy release and dissipation of the parts that are moving at varying speeds. Two different types of inertial torques are distinguished in the operation of sucker-rod pumping units: articulating moment of inertia and rotary moment of inertia. (Takács, 2015) These torques have a small magnitude compared to the rod torque and the counterbalance torque, and therefore are often omitted from the calculation of the mechanical net gearbox torque. But since the counterbalance torque tries to reduce the torque loading on the gearbox by counteracting the rod torque, the inertial torques can play a significant role on the value of the net gearbox torque, when the two main torques have a similar magnitude. By neglecting the inertial torques from the torque calculations, the resulting suggested counterweight configuration can in fact overload the pumping unit.

5.2.4.1 Articulating Inertial Torque

Since some parts of the pumping unit have an alternating movement during the pumping cycle – beam, horsehead, equalizer, pitmans etc. – the accelerations and decelerations introduce a new torque type, the articulating inertial torque. This torque component exists even at constant pumping speeds. (Gibbs, 1975) This torque component is directly proportional to the angular acceleration of the beam as seen in Equation 14. The value of I_b only depends on the pumping unit designation, its value is supplied by the manufacturer of the pumping unit.

$$T_{ia}(t) = \frac{12}{32.2} \cdot TF(t) \cdot \frac{I_b}{A} \cdot \frac{\mathrm{d}^2\theta_b}{\mathrm{d}t^2} \tag{14}$$

where:

$T_{ia}(t)$	Articulating inertial torque in time [in lb],
$TF(t)$	Torque factor in time [in],
I_b	Mass moment of inertia of the beam, horsehead, equalizer, and bearings referred to the saddle bearing [lb_m ft^2],
A	Linkage dimension [in], and
$\frac{\mathrm{d}^2\theta_b}{\mathrm{d}t^2}$	Angular acceleration of the walking beam [rad/sec^2].

The beam angular acceleration can be obtained using three different methods as seen in **Figure 14**. The first method involves the calculation of the crank angles as the first step, then using the calculation procedure proposed by (Svinos, 1983) to get the required beam acceleration versus time function. This method is exact, but cumbersome, it requires the calculation of angular velocities and accelerations of the cranks and the pitmans, using complex equations, as shown in Chapter 5.5.1.

The second calculation procedure is based on the work of (Gibbs, 1975) and is detailed in Chapter 5.5.2, that determines the beam acceleration by differentiating the measured polished rod displacements twice and then dividing them with the length of link A. Fourier series method is applied to the measured polished rod position points to make the differentiation simple and also to maintain a sufficient accuracy. The error of the method depends on the number of coefficients used in the truncated Fourier series, this behavior is investigated in detail in Chapter 5.5.2. Based on this evaluation, the proposed number of coefficients used in the Fourier series is 10, which provides nearly identical results to the exact calculation method proposed by (Svinos, 1983), see **Figure 22**.

Finally, a basic numerical method is used to validate the results of the previous two methods. This method is presented in detail in Chapter 5.5.3 in detail; its results contain a relatively high fluctuation, but it is helpful to validate the previous two methods, due to the exceptional fit shown in **Figure 22**. These calculation models were investigated in detail by (Takács & Kis, 2014). With increased pumping speed the magnitude of the articulating inertial torque increases, although the correlation is not linear. To find the articulating inertial torque function, the application of the second method proposed by (Gibbs, 1975) is recommended due to its high accuracy combined with little calculation effort.

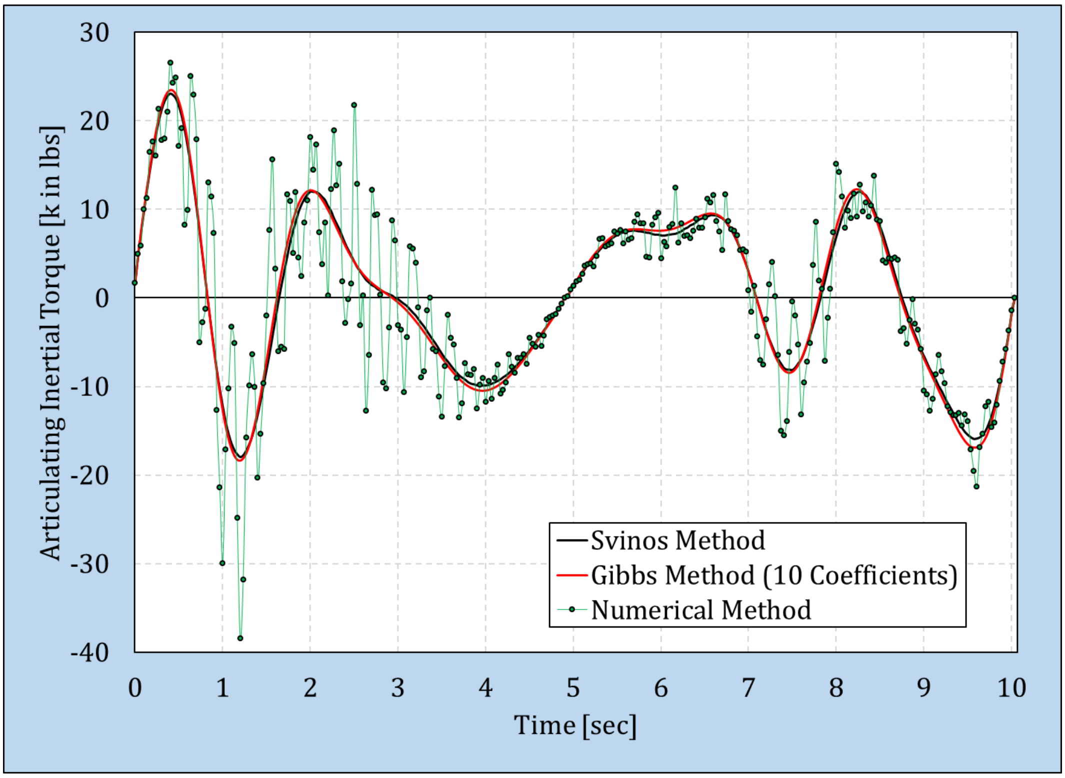


Figure 22 *Calculated articulating inertial torques for the example problem*

5.2.4.2 Rotary Inertial Torque

Unlike the articulating inertial torque, the rotary inertial torque only exists if the crank is turning at varying speeds during the pumping cycle, which is likely when a high slip or ultra-high slip prime mover drives the pumping unit. (Gibbs, 1975) This torque component is directly proportional to the crank angular acceleration, as shown in Equation 15.

$$T_{ir}(t) = \frac{12}{32.2} \cdot I_s \cdot \frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} \tag{15}$$

where:

$T_{ir}(t)$	Rotary inertial torque in time [in lb],
I_s	Mass moment of inertia of the counterweights, cranks and slow-speed gearing referred to the crankshaft [$\mathrm{lb_m\ ft^2}$], and
$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2}$	Angular acceleration of the crank arm [rad/sec^2].

The calculation of the crank angular acceleration in time is carried out in Chapter 5.4. Similarly to the determination of the beam angular acceleration, a simple numerical model is used for validation purposes. I_s is the sum of the mass moments of the listed purely rotating components of the sucker-rod pumping unit, see Equation 16.

$$I_s = \mathrm{I_{cr}} + I_g + I_{cw} \tag{16}$$

where:

I_s	Total mass moment of inertia of the rotating components [lb_m ft^2],
I_{cr}	Mass moment of inertia of the crank arms [lb_m ft^2],
I_g	Mass moment of inertia of the slow speed gearings [lb_m ft^2], and
I_{cw}	Mass moment of inertia of the counterweights [lb_m ft^2].

The value of the cranks' and the slow speed gearings' mass moment of inertia is provided by the manufacturer. Therefore, only the calculation of the counterweights' mass moment of inertia is required to find the value of I_s. Having a symmetrical counterweight configuration, Equation 17 should be used to find the mass moment of inertia of the counterweights.

$$I_{cw} = n \cdot I_{cg} + n_a \cdot I_{cg_a} + (n \cdot m_{cw} + n_a \cdot m_{cw_a}) \cdot \left(\frac{H}{12}\right)^2 \tag{17}$$

where:

I_{cg}	Mass moment of inertia of one main counterweight about its center of gravity [lb_m ft^2],
I_{cg_a}	Mass moment of inertia of one auxiliary counterweight about its center of gravity [lb_m ft^2],
n	Number of main counterweights [-],
n_a	Number of auxiliary counterweights [-],
m_{cw}	Mass of one main counterweight [lb_m],
m_{cw_a}	Mass of one auxiliary counterweight [lb_m], and
H	Distance between the crankshaft and the center of gravity of the main counterweight [in].

Equation 18 is used to find the distance between the crankshaft and the center of gravity of a main counterweight:

$$H = \sqrt{(M - D)^2 + (HW_{cr} + Y_{cw})^2} \tag{18}$$

where:

M	Maximum distance of the counterweight's center of gravity from the long end of the crank [in],
D	Distance of the counterweight from the long end of the crank [in],
HW_{cr}	Half-width of the crank [in], and
Y_{cw}	Vertical distance of the center of gravity of the counterweight from its base [in].

Since the counterweight configuration in the example case is symmetrical, Equation 17 can be used to find the missing mass moment of inertia from Equation 16. The mass moment of inertia for the counterweights in the case of the example problem is 614,466 lb_m ft^2, the resulting total mass moment of the purely rotating parts, I_s is 866,110 lb_m ft^2. The resulting rotating moment of inertia function in time for the example problem is shown in **Figure 23**.

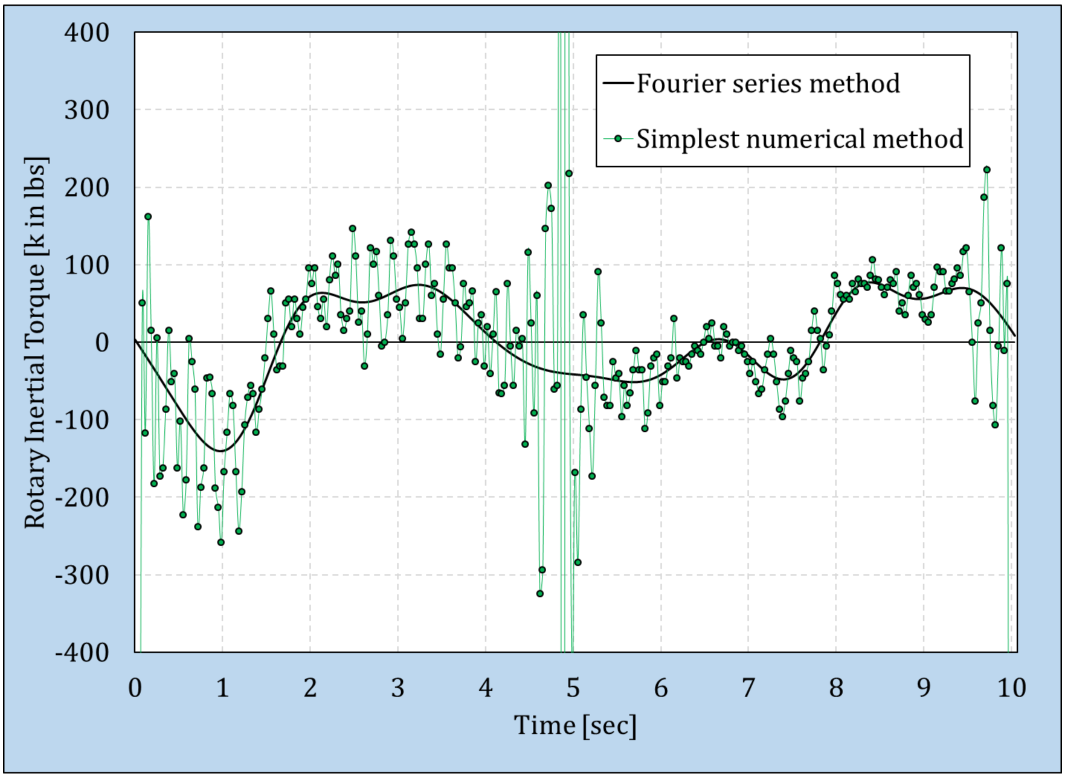


Figure 23 *Calculated rotary inertial torque for the example problem*

If identical counterweights are used, but their placement is different on the crank arm, I developed Equation 19 to properly provide the mass moment of inertia in this case.

$$I_{cw} = n \cdot I_{cg} + n_a \cdot I_{cg_a} + \sum_{i=1}^{n} \left(m_{cw_i} \cdot \left(\frac{H_i}{12} \right)^2 \right) + \sum_{i=1}^{n_a} \left(m_{cw_{a_i}} \cdot \left(\frac{H_i}{12} \right)^2 \right) \tag{19}$$

where:

I_{cg}	Mass moment of inertia of one main counterweight about its center of gravity [lb_m ft^2],
I_{cg_a}	Mass moment of inertia of one auxiliary counterweight about its center of gravity [lb_m ft^2],
n	Number of main counterweights [-],
n_a	Number of auxiliary counterweights [-],
m_{cw_i}	Mass of the i^{th} counterweight [lb_m],
$m_{cw_{a_i}}$	Mass of the i^{th} auxiliary weight [lb_m], and
H_i	Distance between the crankshaft and the center of gravity of the i^{th} main counterweight [in].

For asymmetrical counterweight configurations I created Equation 20, that defines the counterweights' mass moment of inertia for any counterbalancing scenario on crank balanced sucker-rod pumping units.

$$I_{cw} = \sum_{i=1}^{n}\left(I_{cg_i} + \sum_{j=1}^{n_{a_i}} I_{cga_{i_j}} + \left(m_{cw_i} + \sum_{j=1}^{n_{a_i}} m_{cwa_{i_j}}\right)\cdot\left(\frac{H_i}{12}\right)^2\right) \tag{20}$$

where:

n	Number of main counterweights [-],
I_{cg_i}	Mass moment of inertia of the ith main counterweight about its center of gravity [lb$_m$ ft^2],
n_{a_i}	Number of auxiliary weights on the ith main counterweight [-],
$I_{cga_{i_j}}$	Mass moment of inertia of the jth auxiliary weight on the ith main counterweight about its center of gravity [lb$_m$ ft^2],
m_{cw_i}	Mass of the ith main counterweight [lb$_m$], and
$m_{cwa_{i_j}}$	Mass of the jth auxiliary weight on the ith main counterweight [lb$_m$].

5.2.5 Net Gearbox Torque

The net gearbox torque is the sum of all torque components acting on the slow-speed shaft of the gearbox. Its variation throughout the pumping cycle is shown in **Figure 24** for the example problem along with the calculated individual torque components.

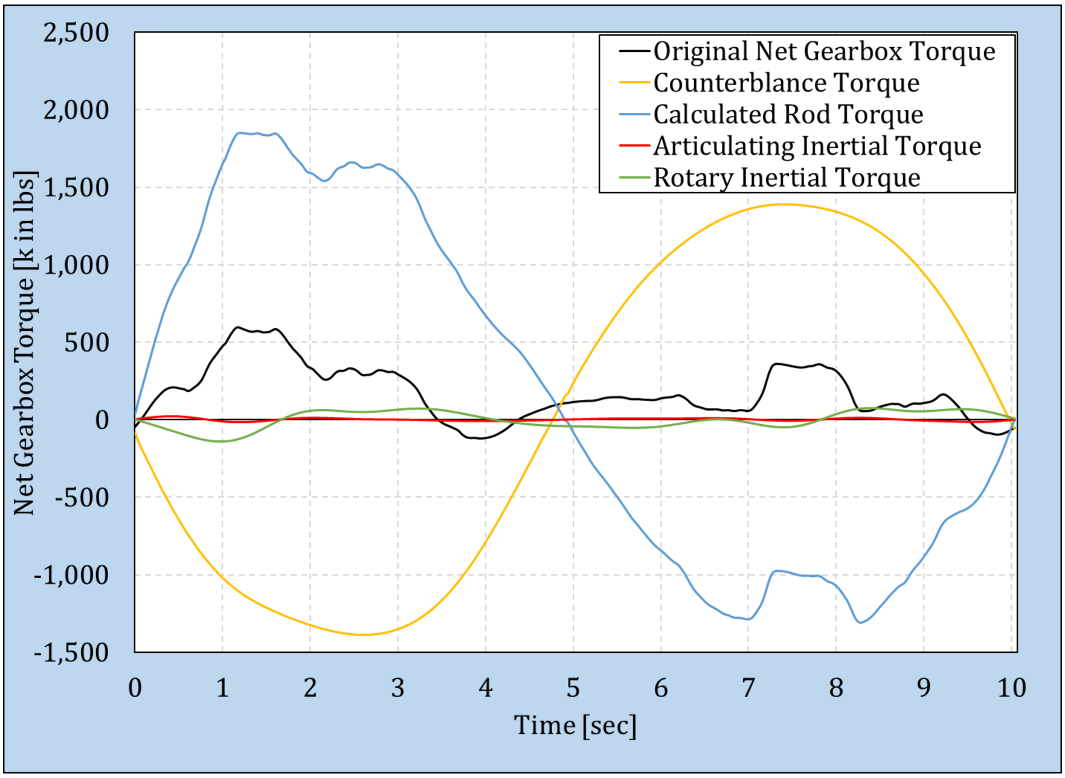


Figure 24 *Torque components acting on the gearbox for the example problem*

The inertial torques have smaller amplitude than the other two main torques, but their influence can be significant. The determination, whether the torsional loading of the gearbox exceeds the maximum allowed torque is essential to maintain a sufficiently long lifetime of the gear reducer, as illustrated previously in **Figure 4**.

For the example case, the comparison of the net gearbox torque found using the newly introduced method in the thesis to the result of the TWM software is shown in **Figure 25**. By neglecting the inertial torques, the TWM finds the pumping unit to be overloaded. In contrast, this conclusion is incorrect, based on the results of the complete torque analysis.

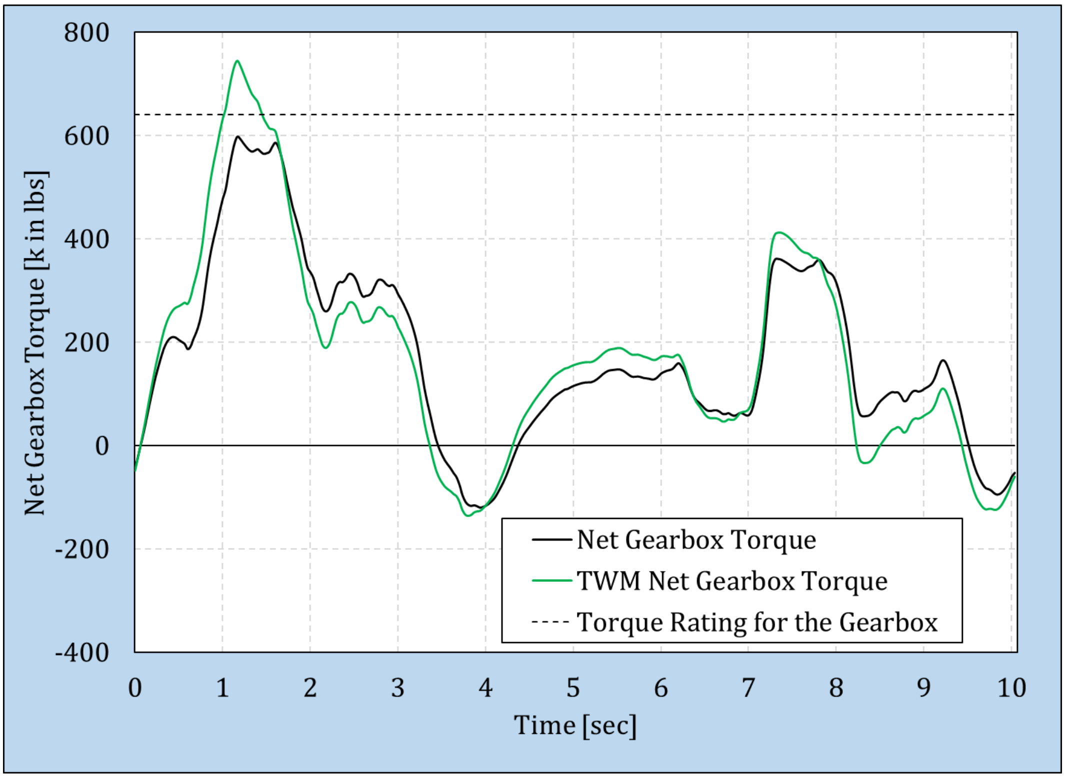


Figure 25 *Comparison of net gearbox torque variations*

5.3 Determination of the Crank Angle vs Time

Modern electronic dynamometers register polished rod displacements and loads in function of time at uniform time intervals throughout the measurement. But all four torque components acting on the gearbox are functions of the crank angle, not recorded in the dynamometer survey. This circumstance necessitates the determination of the crank angles in time from the measured polished rod displacements. To handle this problem, a successive approximation was introduced by (Takács, Kis, & Koncz, 2015). For this calculation, in addition to the measured data, only the rotation and the API designation of the sucker-rod pumping unit is required. The corresponding linkage lengths are found in the tables provided by the manufacturer of the pumping unit.

The determination of the crank angle variation in time is the cornerstone of a proper calculation of the mechanical net gearbox torque. The crank angle values produced by the proposed calculation method are compared to the Total Well Manager results. TWM has slight error in the determination of the crank angles, but it is important to find these values with the highest accuracy, because it is the first major calculation step in the evaluation of the dynamometer survey. Any error in this step will reduce the precision of every calculation based on the calculated crank angles.

5.3.1 Necessity of a Numerical Method

From a measured polished rod displacement, the direct calculation of the corresponding crank angles is impossible because for every polished rod position there is one corresponding crank angle on the up- and downstroke. Since an explicit relationship does not exist between the position of rods and the crank angle, a numerical calculation method must to be used in order to determine the crank angles corresponding to the measured polished rod positions.

To infer the crank angles, the pumping unit's kinematic parameters are used. This process is complete, when the measured polished rod position is equal to the position determined from the kinematic analysis of the pumping unit, see Equation 21. The crank angle that produces the appropriate dimensionless position of rods value corresponds to the measured time. (Takács, Kis, & Koncz, 2015)

$$s_i = S \cdot PR(\theta_{calc}) \tag{21}$$

where:

s_i	i[th] element of the measured polished rod position array [in],
S	Stroke length [in], and
$PR(\theta_{calc})$	Dimensionless position of rods at crank angle θ_{calc} [-].

This process is carried out for each measured polished rod position, the product of this procedure is the series of crank angle values valid at the measured times. (Takács, Kis, & Koncz, 2016) For this purpose, a successive approximation numerical method is proposed, it is presented in detail in the following subchapter. This calculation method can provide the crank angle values at the measured data points with any desired precision.

5.3.2 Successive Approximation Numerical Method

This method is used to determine the crank angles, θ, that produce the same PR (position of rod) values as the measured polished rod displacements, its flowchart is shown in **Figure 26**.

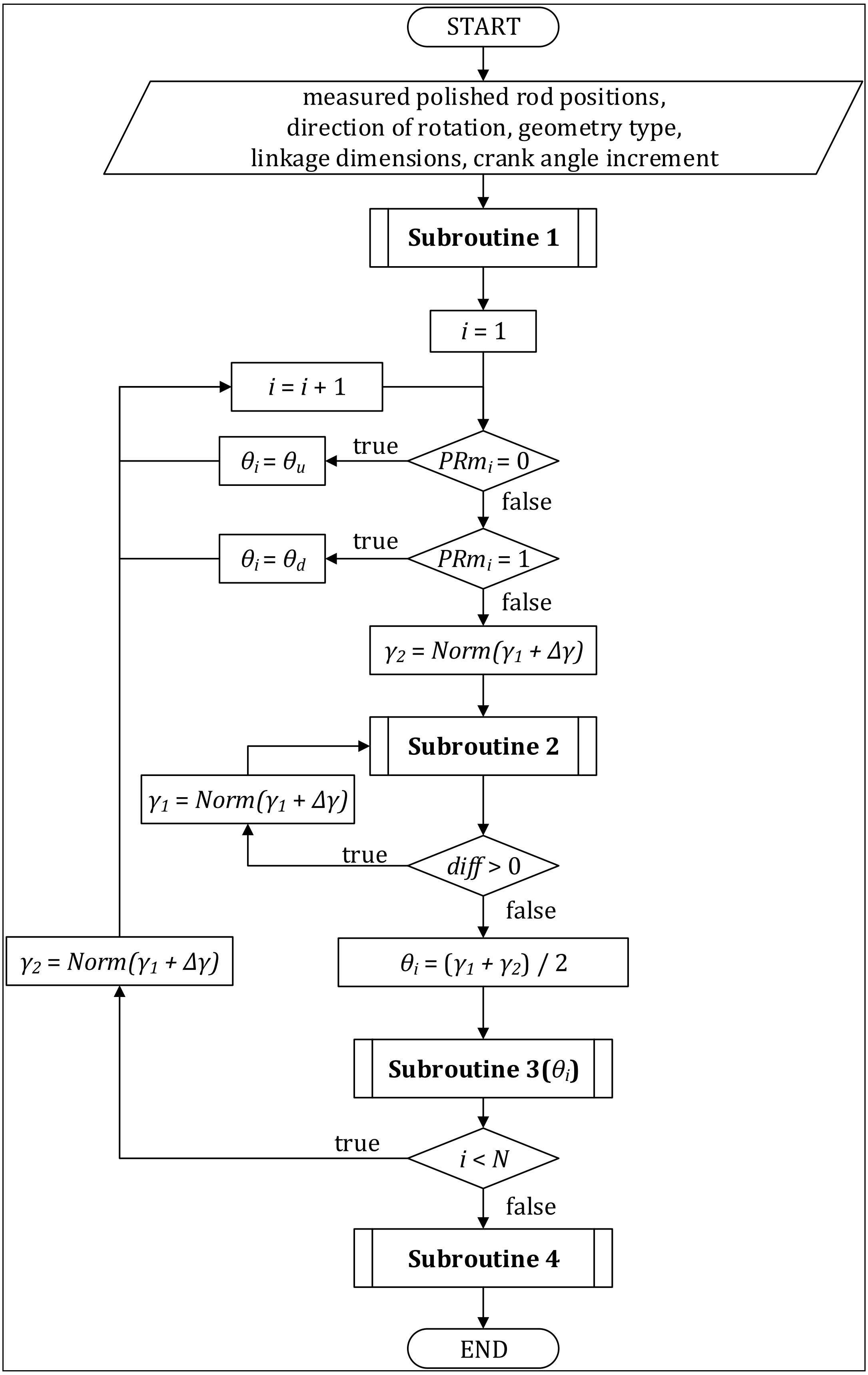


***Figure 26** Flowchart of the successive approximation numerical method that finds the crank angles corresponding to the measured polished rod positions*

This numerical method can be applied to any dynamometer survey carried out on Conventional, Reverse Mark, Mark II and Air Balanced units. The Conventional and Air Balanced units can operate with both clockwise and counter-clockwise direction of rotation. In their counter-clockwise rotational case the crank angles – also the γ_1 and γ_2 auxiliary crank angles – have to be recalculated with Equation 22.

$$\theta_{CCW} = 2\pi - \theta_{CW} \tag{22}$$

where:

$\theta_{CCW}, \theta_{CW}$ Crank angle in counter-clockwise and clockwise direction, respectively [rad].

The fundamental idea of the calculation method is to create a moving pair of auxiliary crank angles – $\gamma_1$and γ_2 – and to determine, when the crank angle corresponding to the measured position of rods is between those two. These two angles are always the same distance apart, namely the used crank angle increment, $\Delta\gamma$. At these angles the corresponding position of rods values – $PR(\gamma_1)$ and $PR(\gamma_2)$, respectively – are evaluated using the API kinematic model for sucker-rod pumping units API Spec. 11E (API, 2008).

The crank angle of the sucker-rod pumping unit is always non-negative and smaller than 2π. If the value of γ_1 or γ_2, reaches, or exceeds 2π during the numerical calculation, the $Norm$ function adjusts their value, so it will be in the $[0, 2\pi[$ interval. The output of this procedure, as discussed before, is the crank angle array valid at the measured polished rod positions. The calculated crank angle values vs time for the example problem are presented in **Figure 27**, along with the results of the TWM software.

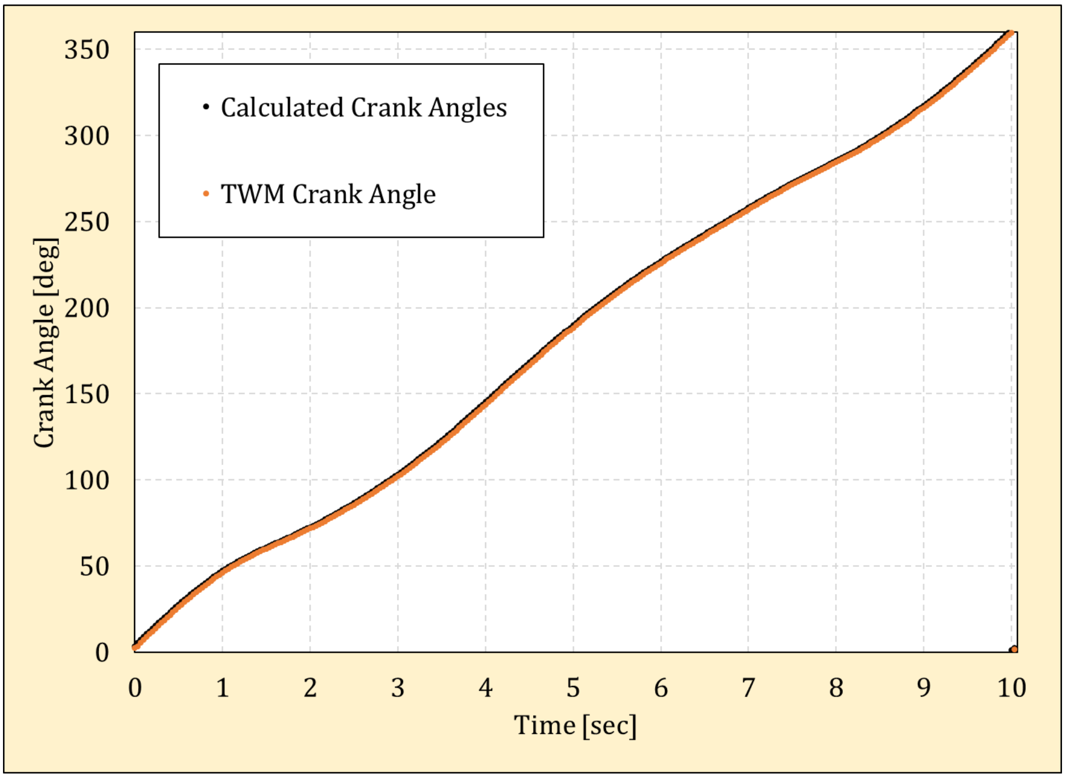


Figure 27 *Crank angles calculated for the example problem*

Figure 28 shows the difference between the calculated crank angles by the previously described method and the results of the TWM software, indicated with blue circles. The TWM software underestimates the crank angles at every data point, the difference between the results is between 0.5 deg and 2.2 deg with an average of 1.2 deg. The reason behind the outlier values at the beginning of the upstroke and downstroke is the fact that the difference of the measured positions by the dynamometer in these regions are comparable to the accuracy of the equipment. This difference in the crank angle calculation is magnified mainly in the inertial torque calculations.

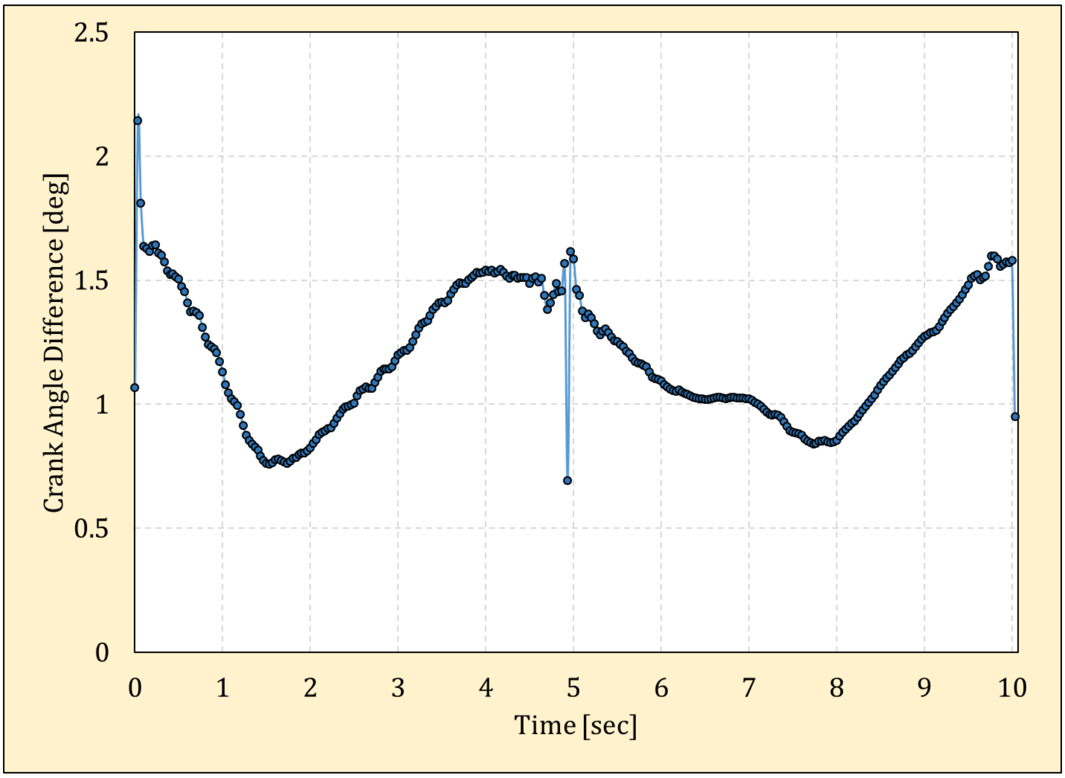


Figure 28 *Crank angle differences between the proposed method and the TWM results*

5.3.2.1 Subroutine 1 of the Successive Approximation Method

Subroutine 1 produces the $\theta$and ψ angles corresponding to the topmost and lowermost positions of the polished rod, determines the stroke length and creates the dimensionless position of rods array from the measured polished rod positions. Its flowchart is shown in **Figure 29** and the formulae for the four investigated sucker-rod pumping units are presented in **Table 4**. The formulae introduced in **Table 4** are in accordance with the API Spec. 11E (API, 2008).

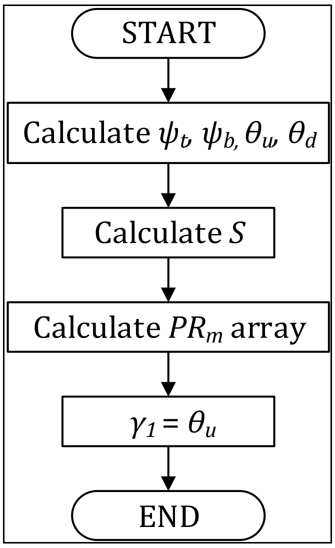


Figure 29 *The flowchart of Subroutine 1*

Table 4 *Formulae used in Subroutine 1*

<table>
<tr><th>Conventional and Reverse Mark</th><th>Mark II</th><th>Air Balanced</th></tr>
<tr><td>$\phi = sin^{-1}\left(\frac{I}{K}\right)$</td><td>$\phi = sin^{-1}\left(\frac{I}{K}\right) + \pi$</td><td>$\phi = \pi - sin^{-1}\left(\frac{I}{K}\right)$</td></tr>
<tr><td>$\psi_b = cos^{-1}\left(\frac{C^2 + K^2 - (P+R)^2}{2 \cdot C \cdot K}\right)$</td><td colspan="2">$\psi_b = cos^{-1}\left(\frac{C^2 + K^2 - (P-R)^2}{2 \cdot C \cdot K}\right)$</td></tr>
<tr><td>$\psi_t = cos^{-1}\left(\frac{C^2 + K^2 - (P-R)^2}{2 \cdot C \cdot K}\right)$</td><td colspan="2">$\psi_t = cos^{-1}\left(\frac{C^2 + K^2 - (P+R)^2}{2 \cdot C \cdot K}\right)$</td></tr>
<tr><td>$\theta_u = \phi - sin^{-1}(W_1)$</td><td>$\theta_u = \phi - sin^{-1}(W_2) + \pi$</td><td>$\theta_u = \phi + sin^{-1}(W_1) - \pi$</td></tr>
<tr><td>$\theta_d = \phi - sin^{-1}\left(\frac{C \cdot sin(\psi_t)}{P - R}\right) + \pi$</td><td colspan="2">$\theta_d = \phi + sin^{-1}\left(\frac{C \cdot sin(\psi_t)}{P + R}\right)$</td></tr>
<tr><td colspan="3">$W_1 = \frac{C \cdot sin(\psi_b)}{P + R}$ $W_2 = \frac{C \cdot sin(\psi_b)}{P - R}$</td></tr>
</table>

In the first step of Subroutine 1 the auxiliary angles corresponding to the start of the upstroke and downstroke of the unit are determined. Thereafter the stroke length of the pumping unit is calculated using Equation 23.

$$S = A \cdot (\psi_b - \psi_t) \tag{23}$$

where:

A	Linkage dimension [in], and
ψ_b, ψ_t	Auxiliary angle at the bottommost and topmost position of the polished rod, respectively [rad].

For a given sucker-rod pumping unit the stroke length can be changed by attaching the pitmans to a different wrist pin bearing, therefore modifying the length of link R. The calculated stroke length for the example problem is 169.82 in. Based on the measured polished rod positions, $s(i)$, the calculation of the appropriate dimensionless positions is possible using Equation (24).

$$PR_m(\theta)_i = \frac{s_i}{S} \tag{24}$$

where:

$PR_m(\theta)_i$	Dimensionless polished rod position for the i^{th} measured point [in],
s_i	i^{th} measured polished rod position [in], and

S Stroke length [in].

The start of the measured data points of the dynamometer survey should begin with the first point in the upstroke region to cover the whole pumping cycle. In this case the suggested starting value of γ_1 is equal to θ_u calculated by Subroutine 1. Otherwise, choosing a higher starting value for the auxiliary angle γ_1 can cause the faulty calculation of the crank angle in the downstroke corresponding to the position of rods. The next step of the calculation is to check whether the given PR value is equal to 0 or 1. In these cases the exact crank angles – θ_u and θ_d respectively – are previously calculated by Subroutine 1 and are added to the crank angle array.

5.3.2.2 Subroutine 2 of the Successive Approximation Method

The second subroutine determines the relative position of rods for the two auxiliary crank angles, by producing an indicative parameter, *diff*. The position of rods corresponding to a given crank angle is calculated using Equation 25. The flowchart of the second subroutine is shown in **Figure 30**.

$$PR(\theta) = \frac{(\psi_b - \psi)}{(\psi_b - \psi_t)} \tag{25}$$

where:

$PR(\theta)$ Position of rods [-],

ψ Auxiliary angle defined in **Figure 5** through **Figure 8** [rad], and

ψ_b, ψ_t Angle ψ at the start of the up- and downstroke, respectively [rad].

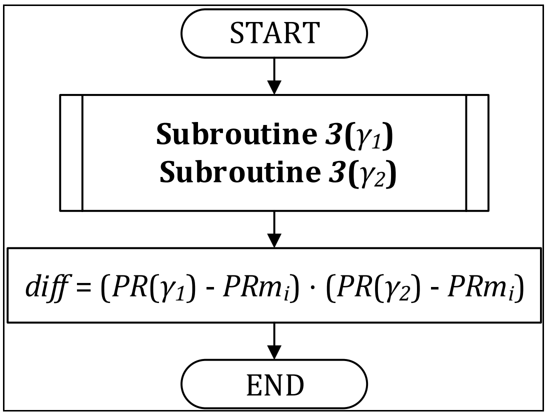


Figure 30 *The flowchart of Subroutine 2*

5.3.2.3 Subroutine 3 of the Successive Approximation Method

The calculation of the position of rods at the auxiliary crank angle pair is done by using Subroutine 3. The flowchart of this subroutine is shown in **Figure 31**, the governing equations are shown in **Table 2** for the investigated pumping unit geometries.

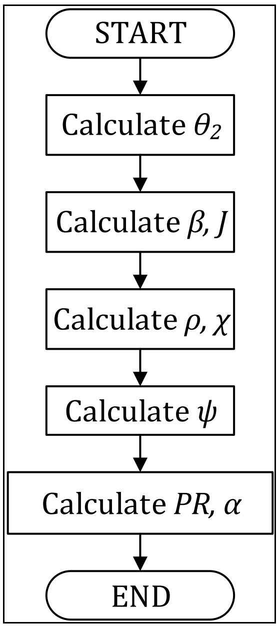


Figure 31 *The flowchart of Subroutine 3*

This calculation is straightforward if the direction of rotation, the geometry, and the length of the linkage dimensions of the investigated pumping unit are known. The input of this subroutine is a crank angle, the outputs are the necessary auxiliary angles listed in **Table 2** and the position of rods calculated by using Equation 25. The auxiliary angles used in this subroutine are defined for every pumping unit geometry in **Figure 5** through **Figure 8**. This calculation process is carried out in Subroutine 2 and in the main calculation of the successive approximation method as seen in **Figure 30** and **Figure 26**, respectively.

After finishing the calculations described in Subroutine 2, the calculated positions of rods are compared with the ith measured dimensionless position from the dynamometer survey. Their difference from the given PR_m value are multiplied, therefore the parameter *diff* has a negative value if the position of rods from the dynamometer survey is between the calculated $PR(\gamma_1)$ and $PR(\gamma_2)$, and has a positive value otherwise, see **Figure 26**. If the value of *diff* is positive, then both γ_1 and γ_2 are increased by $\Delta\gamma$, and Subroutine 2 is repeated with the updated auxiliary crank angle pair. When *diff* has a negative value the crank angle corresponding to the measured relative polished rod position is between the two auxiliary crank angles; its value is obtained averaging γ_1 and γ_2. Because of the sufficiently small crank angle increment used in the program ($\Delta\gamma = 0.1°$), a linear approximation is more than enough to find the crank angle that satisfies Equation 24. The maximum error of this procedure is half of the used increment, $\Delta\gamma$, which is sufficiently small for the purpose. To determine all crank angles corresponding to the measured relative polished rod positions, the previously detailed steps are repeated until the number of the measured polished rod positions in the dynamometer survey for the investigated pumping cycle is reached.

5.3.2.4 Subroutine 4 of the Successive Approximation Method

When the sampling rate is low compared to the pumping speed of the unit, the topmost and lowermost polished rod positions may be missing from the dynamometer survey. In such cases, for the proper crank angle calculation an additional validation step is required, as illustrated in **Figure 32.**

The black dots in **Figure 32** represent the data from the original dynamometer survey, the orange circles show the case when the sampling rate of the measurement is halved. The neighborhood of the crank angle at the start of the downstroke, θ_d is focused for better representation of the problem. As discussed previously, apart from the topmost and lowermost positions, there is one crank angle both in the upstroke and downstroke that corresponds to the measured position of rods.

An error emerges in the crank angle calculation, when the last measured position of rods in the upstroke is smaller than the first measured position in the downstroke, which is true in the illustrated scenario. In this particular case the calculation method presented gives the wrong crank angle as the solution. Instead of calculating the crank angle that corresponds to the position in the downstroke, the crank angle in the upstroke is calculated, which is shown with a green circle in the figure.

Since the dynamometer survey contains data measured at constant time intervals, this incorrect calculation will produce a smaller crank angle change in the upstroke, and to compensate this, a greater change in the beginning of the downstroke is introduced. These crank angle differences are visualized by the green horizontal lines. Even if the crank angular velocity is not constant, the variation of the crank angle is smooth, which is represented by the brown horizontal lines corresponding to the properly calculated crank angles.

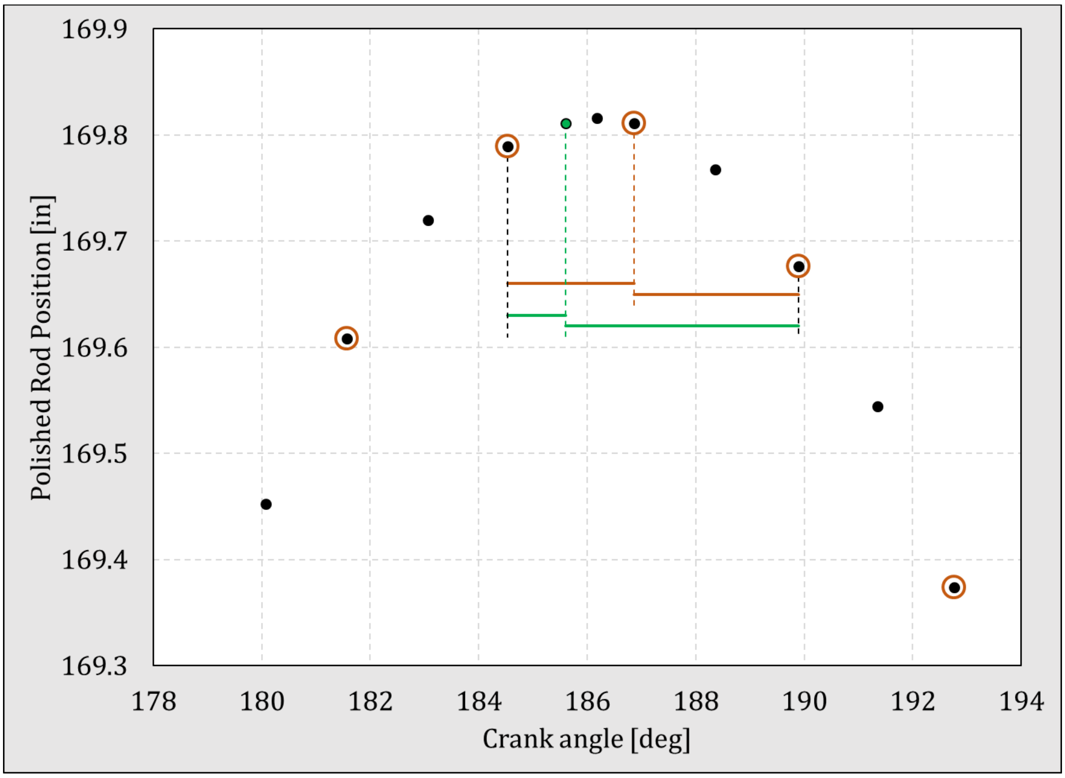


Figure 32 *Calculation of the incorrect crank angle without validation*

If this faulty calculation is not corrected and crank angles without verification are used, the crank angular velocity and crank angular acceleration functions can have extreme variations compared to the rest of the pumping cycle. This will consequently be transferred to the inertial torque calculations. Subroutine 4 tackles these calculation errors, its flowchart is shown in **Figure 33**.

First, it checks whether the dimensionless PR 0 and 1 are in the calculated position of rods array. If at least one of the two extremes is missing, Subroutine 4 determines, whether the measured positions create the possibility of the miscalculation, and corrects the crank angle if the relationship between the measured positions fulfills the condition. Usually the magnitude of the inertial torques are at least one order of magnitude smaller than the rod torque, or the counterbalance torque, but using this incorrectly calculated crank and beam angular acceleration functions, their value can fundamentally change the net torque variation.

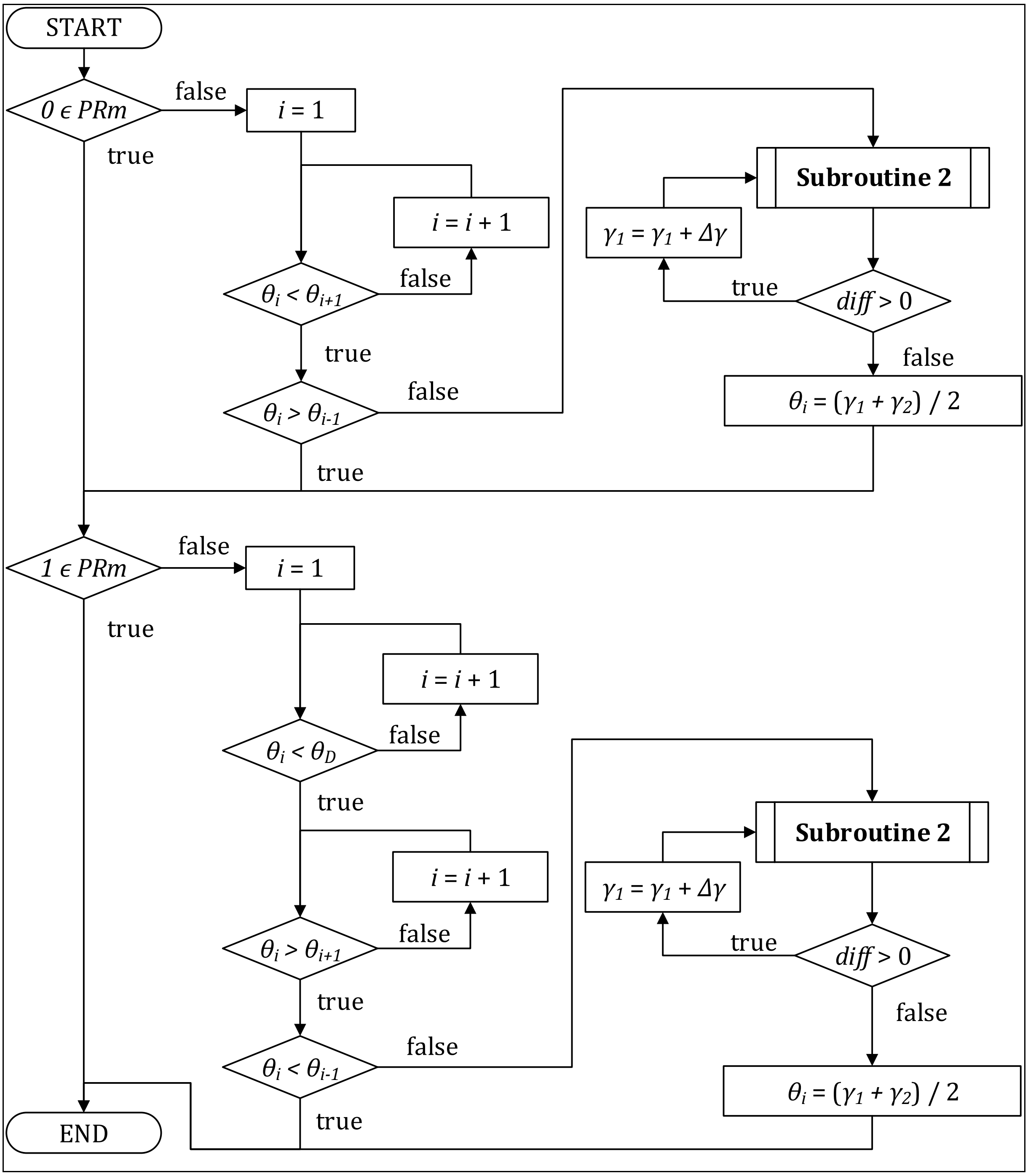


Figure 33 *Flowchart of Subroutine 4*

5.4 Calculation of the Crank's Angular Acceleration

To find the crank angular acceleration from the calculated crank angle values, first the angular velocity of the crank must be determined. Since the motion of the crank arm is periodic, every property, that describes the pumping unit has the same values at the start and end of the stroke. In the present chapter the determination of the crank angular velocity using multiple methods is presented. The first method is a basic numerical method, that is used for verification purposes. The second and third methods use Fourier series in different ways to describe the crank angular velocity function.

5.4.1 *Importance of Using a Simple Numerical Method*

The application of simple numerical methods is advantageous in the validation of more complex procedures. It is vitally important, that the results of any calculation should not have any methodical errors. The proposed numerical method produces the crank angular velocities by using Equation 26.

$$\frac{\Delta\theta}{\Delta \mathrm{t}}_{num_i} = \frac{Norm(\theta_{i+1} - \theta_i)}{t_{i+1} - t_i} \tag{26}$$

where:

$\frac{\Delta\theta}{\Delta \mathrm{t}}_{num_i}$	i^{th} element of the numerically calculated crank angular velocity array [rad/s],
θ_i	i^{th} element of the calculated crank angle array [rad], and
t_i	i^{th} element of the time array for the calculated crank angle array [sec].

This method approximates the tangent of the crank angle function in between the measured times with the secant created by the two neighboring crank angle points. This method creates a crank velocity array that contains one less element than the original crank angle array. The times at which the calculated crank angular velocities are valid can be determined using Equation 27. This process produces a rough estimate of the crank angular velocity variation throughout the pumping cycle.

$$t_{num_i} = \frac{t_i + t_{i+1}}{2} \tag{27}$$

where:

t_{num_i}	i^{th} element of the time array for the calculated crank angular velocities [sec], and
t_i	i^{th} element of the time array for the calculated crank angle array [sec].

5.4.2 *Using Fourier Series to Describe Periodic Behavior Based on Measured Data*

Generally, the best approach to describe complex periodic behavior is to use Fourier series. The general formula of the Fourier series is given in Equation 28. The function of the Fourier approximation requires the determination of the a and b coefficient arrays. In Equation 28, $a_0$is the constant coefficient, moving the function in the vertical direction, while the a and b arrays contain the information of the variation of the function over the investigated period.

$$F(x) = \frac{a_0}{2} + \sum_{k=1}^{N_F} a_k \cdot \cos\left(\frac{2 \cdot k \cdot \pi \cdot x}{P}\right) + b_k \cdot \sin\left(\frac{2 \cdot k \cdot \pi \cdot x}{P}\right) \tag{28}$$

where:

$F(x)$	Fourier series function [var.],
a_0	Constant coefficient of the Fourier series [-],
N_F	Number of coefficients in the Fourier series [-],
k	Index of the coefficients in the Fourier series [-],
a_k, b_k	k^{th} coefficients of the Fourier series [-], and
P	Period of the Fourier series [sec].

The advantage of the Fourier series is that it can create the best fitting function based on available points with user defined period times. The period time corresponding to the investigated stroke can be found from the calculated crank angle data. If the bottommost position of the polished rod is in the dynamometer survey in both the start and at the end of the stroke, the time required to complete a whole stroke is just the time difference of the last and first measured point in the dynamometer survey. However, if the bottommost position is not the recorded at the end of the stroke, the last data point is the last one that has a smaller crank angle value corresponding to it than θ_U. Using the calculated crank angle array, the time required to complete a whole pumping cycle is determined by Equation 29. The calculated period time for the example problem is 10.06 sec.

$$T = \frac{2\pi \cdot t_N}{(Norm(\theta_N - \theta_1) + 2\pi)} \tag{29}$$

where:

T	Period time [sec],
t_N	Time of the last measured point from the first one [sec], and
θ_1, θ_N	Crank angles at the first and last measured point, respectively [rad].

For the determination of the coefficients, a custom Fourier time array must be created over the previously calculated period. This is achieved by using Equation *30*

$$t_{F_i} = \frac{i \cdot T}{N} \tag{30}$$

where:

t_{F_i}	i^{th} element of the Fourier time array [sec],
i	Index that goes from *0* to *N-1* [-], and
N	Number of measured data points [-].

From the data points the values valid at the elements of the Fourier time array must be interpolated. Since the difference between the i^{th} element of the measured time array and the Fourier time array is relatively small (the maximum value is smaller than the time difference between the measured positions), a linear interpolation provides sufficiently precise values to find the input data for the Fourier series.

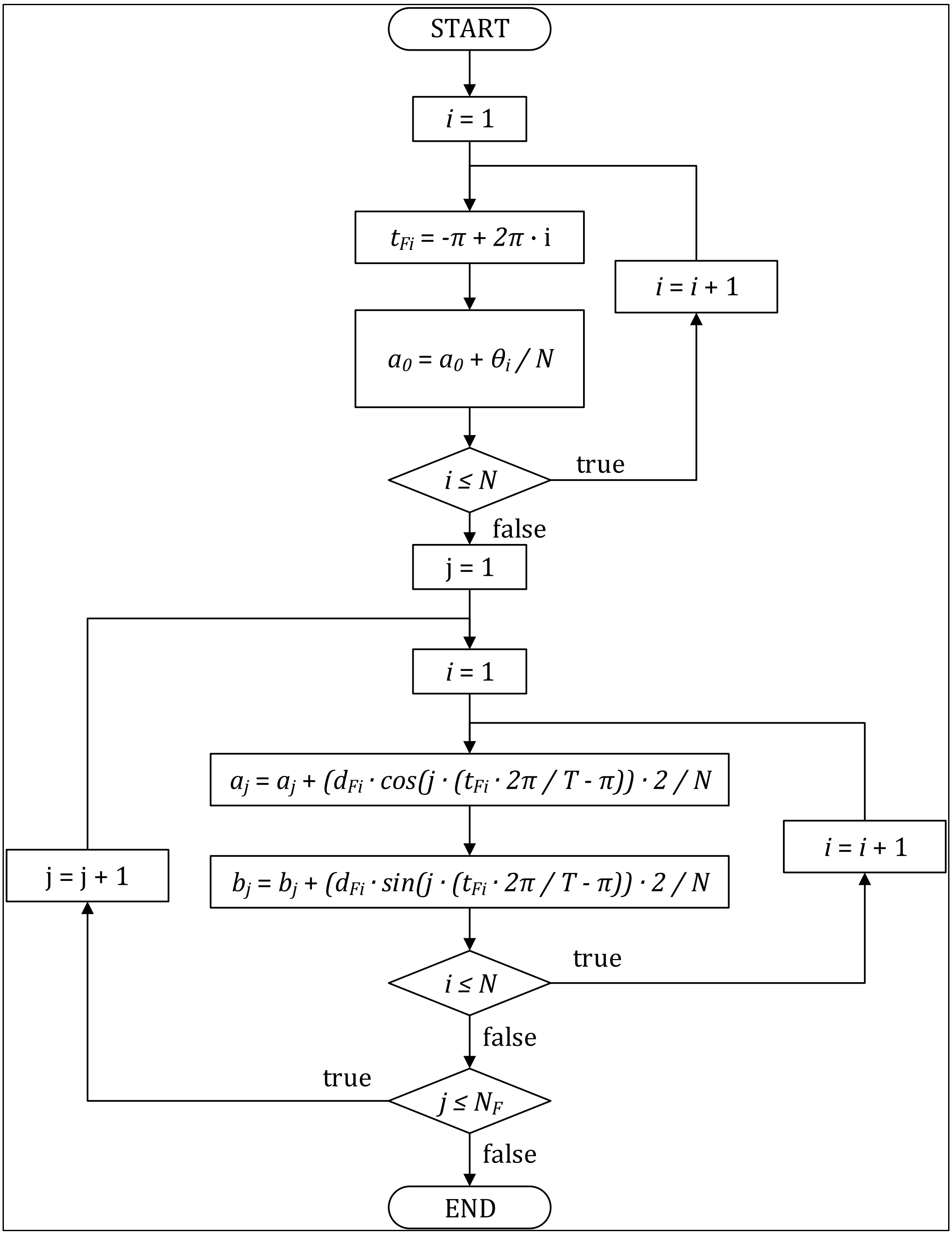


Figure 34 *Flowchart of determining the Fourier coefficients*

Equation 31 is used to find the data array suitable for the Fourier analysis. Once these new arrays are created, the determination of the Fourier coefficients is possible using the method described by **Figure 34**.

$$d_{F_i} = d_i + (d_i - d_{i-1}) \cdot \frac{t_{F_i} - t_i}{t_i - t_{i-1}} \tag{31}$$

where:

$d_{F\,i}$	i^{th} element of the Fourier input data array [var.],
d_i	i^{th} element of the data array [var.],
$t_{F\,i}$	i^{th} element of the Fourier time array [sec], and
t_i	i^{th} element of the measured time array [sec].

Using the calculated coefficients, Equation 32 provides the truncated Fourier series value at the measured times contained in the dynamometer survey.

$$F_i = a_0 + \sum_{k=1}^{N_F} a_k \cdot \cos\left(\left(\frac{2\pi \cdot t_i}{T} - \pi\right) \cdot k\right) + b_k \cdot \sin\left(\left(\frac{2\pi \cdot t_i}{T} - \pi\right) \cdot k\right) \tag{32}$$

where:

F_i	i^{th} solution of the Fourier series at the measured times [var.],
a_0	Constant coefficient of the Fourier series [-],
a_k, b_k	k^{th} coefficients of the Fourier series [-],
t_i	i^{th} element of the measured time array [sec],
T	Period time [sec],
k	Index of the coefficients in the Fourier series [-], and
N_F	Number of coefficients in the Fourier series [-].

Since the Fourier series is a sum of different sine and cosine functions, its differentiation is simple. After the values contained in the Fourier series for the original data are calculated, its time derivative can be determined using Equation 33, the second derivative is defined in Equation 34.

$$\frac{\mathrm{d}F}{\mathrm{d}t}_i = \frac{2 \cdot \mathrm{k}}{\pi} \cdot \sum_{k=1}^{N_F} -a_k \cdot \sin\left(\left(\frac{2\pi \cdot t_i}{T} - \pi\right) \cdot k\right) + b_k \cdot cos\left(\left(\frac{2\pi \cdot t_i}{T} - \pi\right) \cdot k\right) \tag{33}$$

where:

$\frac{\mathrm{d}F}{\mathrm{d}t}_i$	First derivative of the result of the Fourier series at the measured times [var.].

$$\frac{\mathrm{d}^2F}{\mathrm{d}t^2}_i = \frac{-4 \cdot \mathrm{k}^2}{\pi^2} \cdot \sum_{k=1}^{N_F} a_k \cdot \cos\left(\left(\frac{2\pi \cdot t_i}{T} - \pi\right) \cdot k\right) + b_k \cdot sin\left(\left(\frac{2\pi \cdot t_i}{T} - \pi\right) \cdot k\right) \tag{34}$$

where:

$\frac{\mathrm{d}^2F}{\mathrm{d}t^2}_i$	Second derivative of the result of the Fourier series at the measured times [var.].

5.4.3 Determination of the Crank Angular Velocity Using Fourier Series

5.4.3.1 Using Fourier Series on the Calculated Crank Angle Array

The most straightforward solution would be the application of Fourier series on the calculated crank angle values, then the crank angular velocity and angular acceleration can be derived using only differentiation. Since the movement of the crank is periodic, the function regressed on the measured points should produce the same values at the start and at the end of the interval. This statement is true, however, the crank angle function is a sawtooth-like function with a discontinuity at the bottom of the stroke. The reason for this behavior lies in the definition of the crank angle, it always falls between 0 and 2π.

Using the truncated Fourier series detailed in Chapter 5.4.2 on the crank angle array describes the data poorly, as seen in **Figure 35**. The black dots represent the calculated crank angle values; the blue curve shows the calculated truncated Fourier series using the crank angle values as input. Since the operation of any sucker-rod pumping unit is cyclical, all investigated variables are described by functions that have the same value at the start of the upstroke and at the end of the downstroke. Functions with discontinuity – like the crank angle function – cause oscillations of the used truncated Fourier series to ensure identical values at the ends of the investigated time interval.

As seen in **Figure 35**, the Fourier series provides even invalid crank angles, going below 0 deg, and above 360 deg. To find the crank angular acceleration, this function must be differentiated twice. The resulting acceleration pattern would surely be unusable due to the extreme oscillation resulting from the deviation from the crank angle data set. Therefore, this approach to find the acceleration pattern of the crank arm is rejected.

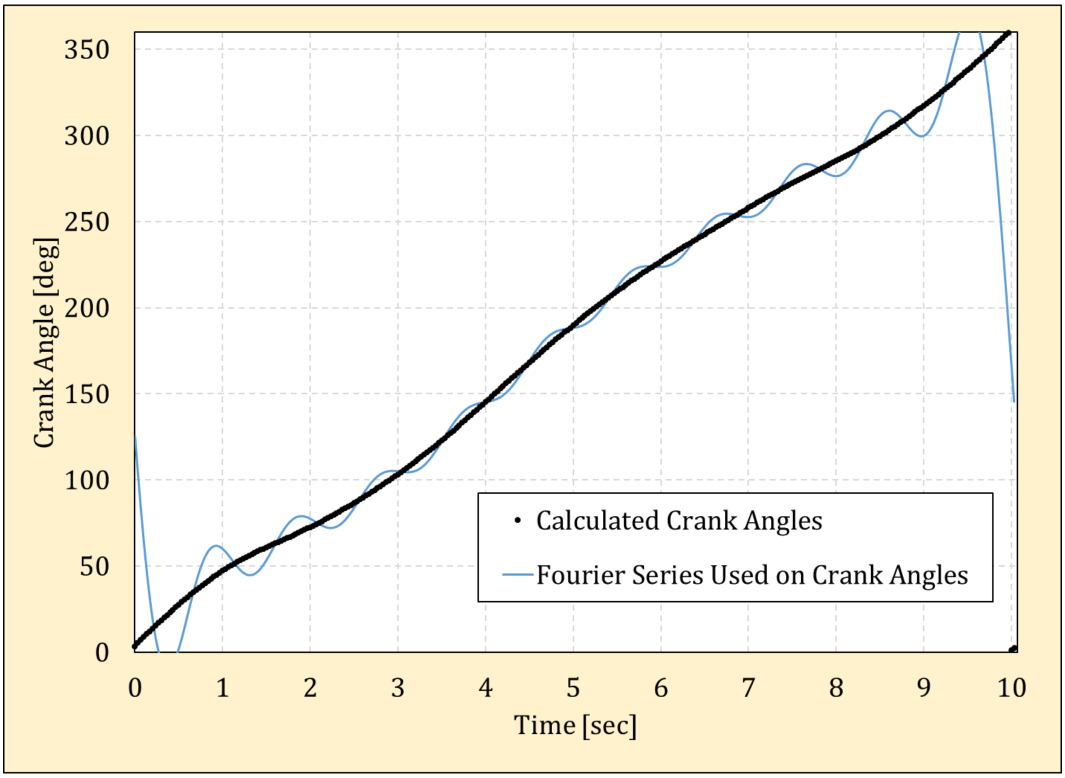


Figure 35 *Using Fourier series on the crank angle array*

5.4.3.2 Using Fourier Series on Numerically Calculated Crank Angular Velocity Arrays

Using Fourier series on data points with a discontinuity in the investigated interval provides unusable results, therefore, to apply the Fourier series properly, a data series has to be created without any discontinuity. By using the numerically calculated crank angular velocity array in Chapter 5.4.1 as the basis, the application of the Fourier series becomes possible. Along with this basic numerically calculated array an improved numerically calculated crank angular velocity array has been created using a five-step stencil method. In this case Equation 35 is used to generate the elements of this array of higher accuracy. This is a novel procedure that finds the crank angular velocity function with improved accuracy compared to the prior works and the results of the TWM software. **Figure 36** shows the comparison between the results of the TWM software and the presented calculation methods. The result of the TWM software has more extreme differences than the two introduced methods. The introduced calculation procedures produce similar crank angular velocities that correctly correlate with the TWM results.

$$\frac{\Delta\theta}{\Delta t}_{num2_i} = \frac{Norm(-\theta_{i+2}+8\cdot\theta_{i+1}-8\cdot\theta_{i-1}+\theta_{i-2})}{12\cdot(t_{i+1}-t_i)} \tag{35}$$

where:

$\frac{\Delta\theta}{\Delta t}_{num2_i}$ Numerically calculated crank angular velocity using the five-step stencil method [rad/sec],

θ_i i^{th} element of the calculated crank angle array [rad], and
t_i i^{th} element of the measured time array [sec].

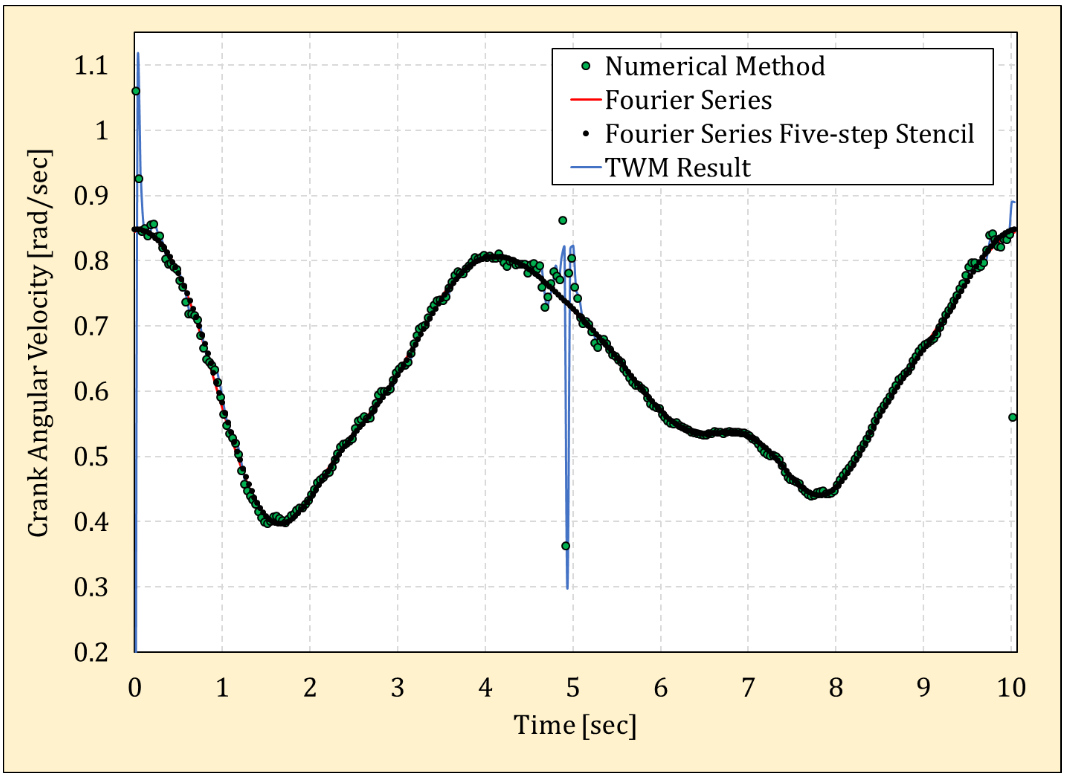


Figure 36 *The calculated crank angular velocity function*

The five-step stencil numerical method provides a smoother crank velocity array; however, it does not provide results for the first and last two measured times. At these times the crank angular velocity is approximated by the average of the first and last 4 calculated values, respectively. The increased precision of using the five-step stencil method becomes visible in **Figure 37**.

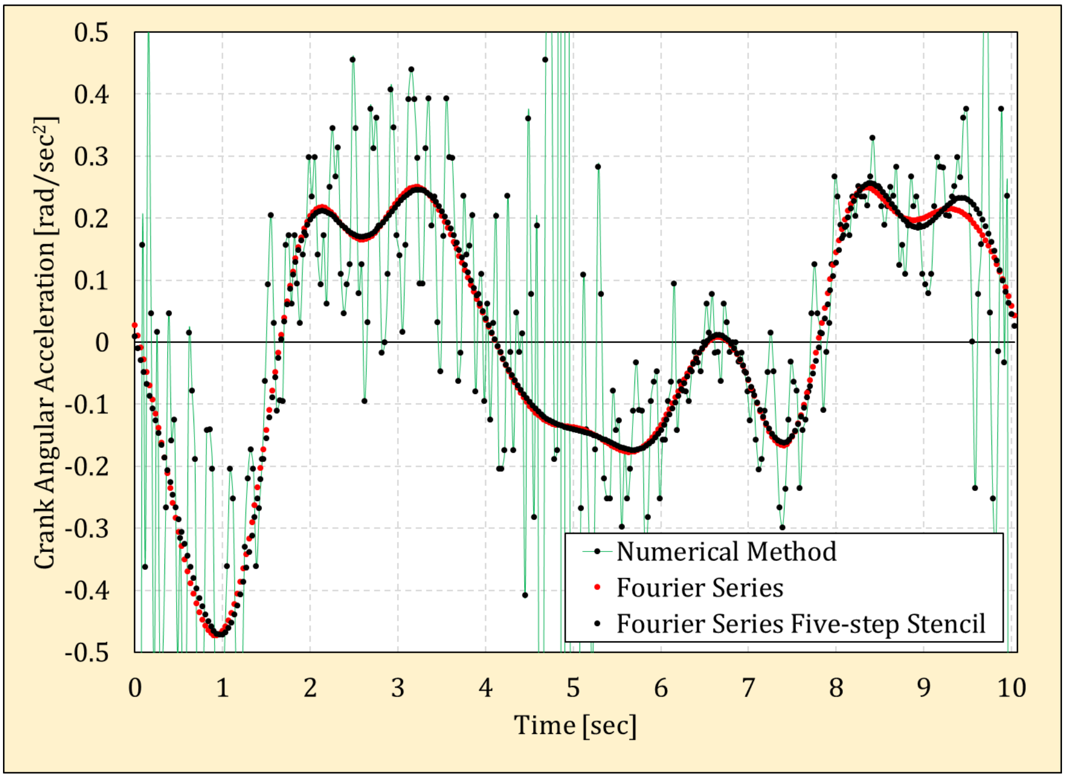


Figure 37 *The calculated crank angular acceleration function*

5.5 Determination of Beam Angular Acceleration

Knowledge of the angular acceleration pattern of the walking beam is necessary for the calculation of the articulating inertial torque, as shown in Chapter 5.2.4.1. Three different methods are presented in detail, and their results are compared to find the best procedure providing the required acceleration of the beam throughout the pumping cycle. The first method is based on the work of Svinos (Svinos, 1983) using vector analysis to describe the kinematic behavior of the pumping unit, the second procedure follows the proposal of Gibbs (Gibbs, 2012) to use Fourier series on the measured polished rod positions to derive the beam angular acceleration, and the third numerical method verifies the results of the two complex methods. (Takács, Kis, & Koncz, 2016)

5.5.1 Calculation of the Beam Acceleration Based on the Svinos Method

The method proposed by Svinos uses complex vectors to describe the exact kinematic behavior of the pumping unit and details a method to find the angular acceleration of the walking beam based on the movement of the crank arm. (Svinos, 1983) In the referred paper, the model is using an auxiliary angle, θ_2 instead of the crank angle, see **Figure 5** through **Figure 8** for its visual representation. Since θ_2 and θ have different orientations, their differentiated functions will have the same magnitude, but different signs, see Equation 36 and Equation 37. To find θ_2 corresponding to the crank angle, θ, use **Table 2**.

$$\frac{\mathrm{d}\theta_2}{\mathrm{d}t} = -\frac{\mathrm{d}\theta}{\mathrm{d}t} \tag{36}$$

$$\frac{\mathrm{d}^2\theta_2}{\mathrm{d}t^2} = -\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} \tag{37}$$

The vector equation of the position of the equalizer bearing from the crankshaft is defined in Equation 38. Both sides of the equation represent a vector pointing from the crankshaft to the equalizer bearing.

$$\vec{K} + \vec{C} = \vec{R} + \vec{P} \tag{38}$$

where:

$\vec{K}, \vec{C}, \vec{R}, \vec{P}$	Linkage vectors, oriented from the crankshaft along with their respective linkage [in].

Equation 39 is found by converting Equation ***38*** into exponential form with relative angles referred to linkage K.

$$K + C \cdot e^{i \cdot \theta_b} = R \cdot e^{i \cdot \theta_2} + P \cdot e^{i \cdot \theta_p} \tag{39}$$

where:

K, C, R, P	Linkage lengths [in], and
$\theta_b, \theta_2, \theta_p$	Auxiliary angles [rad].

The angles in Equation 39 are shown in **Figure 5** through **Figure 8** for the investigated pumping unit geometries and the governing equations calculating them are defined in **Table 5**. After rearranging Equation *39* to find θ_b, both sides of the equation are differentiated with respect to time to produce the time derivative of the beam angle, θ_b. Solving the system of equations received after differentiation (using the Cramer-rule) gives the angular velocity of links R, P and C. The angular velocity of the walking beam is defined by Equation 40.

Table 5 *Auxiliary angles for the Svinos method*

<table>
<tr><th>Conventional and Reverse Mark</th><th>Mark II</th><th>Air Balanced</th></tr>
<tr><td>$\theta_p = cos^{-1}\left(\frac{P^2 + J^2 - C^2}{2 \cdot P \cdot J}\right) + \rho$</td><td colspan="2">$\theta_p = cos^{-1}\left(\frac{P^2 + J^2 - C^2}{2 \cdot P \cdot J}\right) - \rho$</td></tr>
<tr><td colspan="3">$\theta_b = \pi - \psi$</td></tr>
</table>

$$\frac{\mathrm{d}\theta_\mathrm{b}}{\mathrm{d}t} = -\frac{R}{C} \cdot \frac{sin(\theta_p - \theta_2)}{sin(\theta_p - \theta_b)} \cdot \frac{\mathrm{d}\theta}{\mathrm{d}t} \tag{40}$$

where:

$\frac{\mathrm{d}\theta_\mathrm{b}}{\mathrm{d}t}$	Beam angular velocity [rad/sec],
R, C	Linkage lengths [in],
$\theta_b, \theta_2, \theta_p$	Auxiliary angles [rad], and

$\frac{\mathrm{d}\theta}{\mathrm{d}t}$ Crank angular velocity [rad/sec].

By differentiating Equation 40 with respect to time, the angular acceleration of the walking beam is defined by Equation 41.

$$\frac{\mathrm{d}^2\theta_\mathrm{b}}{\mathrm{d}t^2} = \frac{\mathrm{d}\theta_b}{\mathrm{d}t} \cdot \left(\frac{\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2}}{\frac{\mathrm{d}\theta}{\mathrm{d}t}} - \frac{\left(\frac{\mathrm{d}\theta_p}{\mathrm{d}t} - \frac{\mathrm{d}\theta_b}{\mathrm{d}t}\right)}{\tan(\theta_p - \theta_b)} - \frac{\left(\frac{\mathrm{d}\theta_p}{\mathrm{d}t} + \frac{\mathrm{d}\theta}{\mathrm{d}t}\right)}{\tan(\theta_2 - \theta_p)} \right) \tag{41}$$

where:

$\frac{\mathrm{d}^2\theta_\mathrm{b}}{\mathrm{d}t^2}$ Beam angular acceleration [rad/sec^2],

$\frac{\mathrm{d}\theta_\mathrm{b}}{\mathrm{d}t}$ Beam angular velocity [rad/sec],

$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2}$ Crank angular acceleration [rad/sec^2],

$\frac{\mathrm{d}\theta}{\mathrm{d}t}$ Crank angular velocity [rad/sec],

$\theta_b, \theta_2, \theta_p$ Auxiliary angles [rad], and

$\frac{\mathrm{d}\theta_p}{\mathrm{d}t}$ Pitman angular velocity [rad/sec].

The required crank angular velocity and angular acceleration arrays are already calculated in Chapter 5.4. Equation 41 needs the time derivative of the pitman auxiliary angle as an input for the calculation. It is calculated using the same method that produced the beam angular velocity defined in Equation 40. The pitman's angular velocity is found using Equation42.

$$\frac{\mathrm{d}\theta_\mathrm{p}}{\mathrm{d}t} = -\frac{R}{P} \cdot \frac{sin(\theta_b - \theta_2)}{sin(\theta_p - \theta_b)} \cdot \frac{\mathrm{d}\theta}{\mathrm{d}t} \tag{42}$$

where:

R, P Linkage lengths [in],

$\frac{\mathrm{d}\theta}{\mathrm{d}t}$ Crank angular velocity [rad/sec], and

$\theta_b, \theta_2, \theta_p$ Auxiliary angles [rad].

Equation 41 can be used in cases, when the crank angular velocity is not constant during the pumping cycle, as both the crank angular velocity and angular acceleration are taken into account. After following the calculation method of these variables throughout the pumping cycle introduced in Chapter 5.4, the beam angular acceleration variation can be determined.

5.5.2 *Calculation of the Beam Acceleration Based on the Method Proposed by Gibbs*

Gibbs introduced a different way to find the beam acceleration by using the fact, that the polished rod vertical displacement is equal to the length of the arc covered by the outer edge of link A, see Equation 43. (Gibbs, 2012)

$$s(t) = A \cdot (\theta_b(t) - \theta_{b_U}) \tag{43}$$

where:

$s(t)$	Measured polished rod position [in],
A	Linkage length [in],
θ_b	Auxiliary beam angle [rad], and
θ_{b_U}	Auxiliary beam angle at the start of the upstroke [rad].

By expressing the angle θ_b from Equation 43 and differentiating the resulting equation twice with respect to time, the beam angular acceleration is described by the resulting Equation 44.

$$\frac{\mathrm{d}^2\theta_b}{\mathrm{d}t^2} = \frac{\frac{\mathrm{d}^2 s(t)}{\mathrm{d}t^2}}{A} \tag{44}$$

where:

$\frac{\mathrm{d}^2 s(t)}{\mathrm{d}t^2}$	Polished rod acceleration [in/sec^2], and
A	Linkage length [in].

For this calculation only the polished rod positions registered in time –obtained from a dynamometer survey – are required in addition to the length of linkage A. Due to the complex nature of the operation of the sucker-rod pumping unit, the exact polished rod position function, $s(t)$, is not known. The easiest way to produce the required beam angular accelerations is to fit a function on the measured polished rod position points and differentiating it twice.

The best method to describe the polished rod position function is the application of Fourier series on the measured data, introduced in Chapter 5.4.2. Finding the proper coefficients to describe the variation of the polished rod position throughout the pumping cycle provides the required function by Equation 44. Since the measured polished rod position data describe a relatively smooth variation, as shown in **Figure 13**, the recommended number of coefficients required to produce a Fourier function, that properly fits the measured data is 10. (Gibbs, 2012)

To visualize the effect of the number of coefficients used in the truncated Fourier series, **Figure 38** is introduced. If 5 coefficients are used, the accuracy of the regression will not be at an acceptable level, as indicated with the purple curve. However, if the number of coefficients greatly exceeds 10, the resulting function will follow the systematic noise in the variation of the measured points, which is not desired. This is presented with the red curve that uses 30 coefficients for the calculation. The absolute error of the regression is decreased, but unwanted high frequency and low amplitude oscillations are produced due to the unnecessarily high number of coefficients.

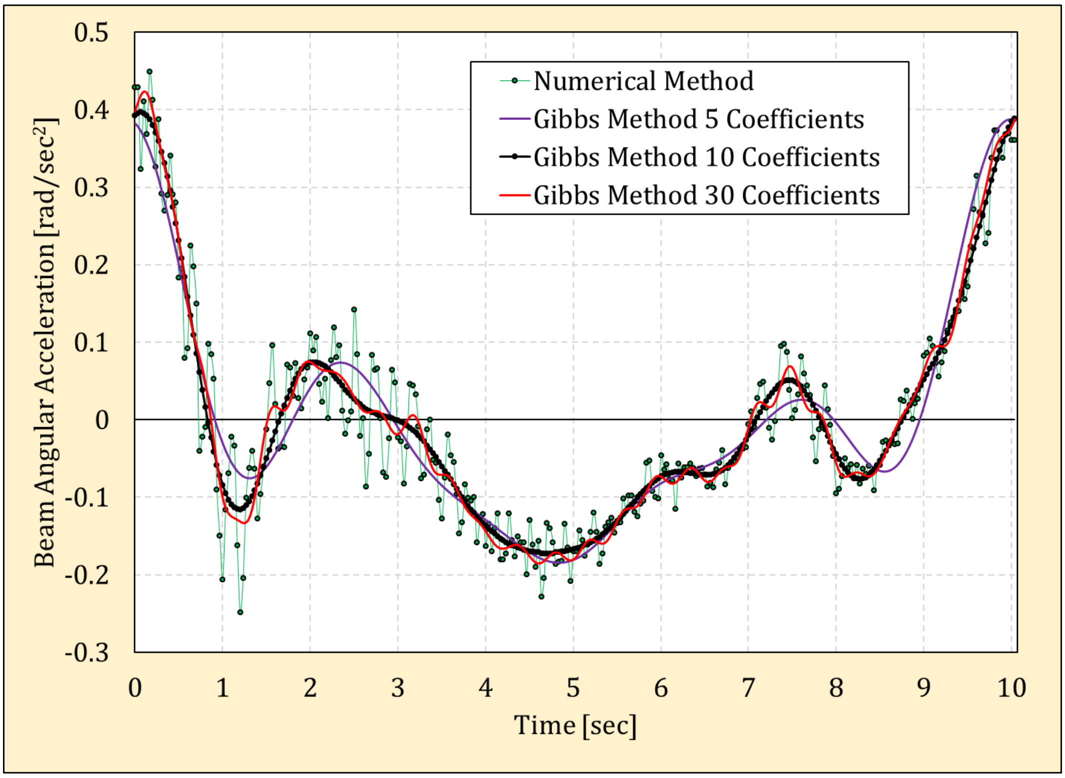


Figure 38 *Comparison of different number of coefficients used in the Gibbs method*

5.5.3 A Simple Numerical Method

For validating purposes, a simple numerical method should be used to make sure, that the more sophisticated methods produce correct results, as detailed in Chapter 5.4.1. A similar method is used in the Total Well Management, it is based on using Equation 44. (Echometer, 2007) The acceleration of the walking beam is found from the calculated polished rod acceleration. To find the polished rod acceleration pattern, first, the polished rod velocity has to be determined with Equation 45, which is done by numerical differentiation of the measured polished rod positions.

$$\frac{\Delta s(t)}{\Delta t}_i = \frac{s_{i+1} - s_i}{t_{i+1} - t_i} \qquad (45)$$

where:

$\frac{\Delta s(t)}{\Delta t}_i$	i^{th} element of the numerically calculated polished rod velocity array [in/sec],
s_i	i^{th} element of the measured polished rod position array [in], and
t_i	i^{th} element of the measured time array [sec].

These velocities are valid between the measured times, see Equation 27. Further differentiating the polished rod velocity array, the acceleration of the polished rod is determined, using Equation 46.

$$\frac{\Delta^2 s(t)}{\Delta t^2}_i = \frac{\frac{\Delta s(t)}{\Delta t}_{i+1} - \frac{\Delta s(t)}{\Delta t}_i}{t_{i+1} - t_i} \tag{46}$$

where:

$\frac{\Delta^2 s(t)}{\Delta t^2}_i$	i^{th} element of the numerically calculated polished rod acceleration array [in/sec],
$\frac{\Delta s(t)}{\Delta t}_i$	i^{th} element of the numerically calculated polished rod velocity array [in/sec], and
t_i	i^{th} element of the measured time array [sec].

Using Equation 44 the beam angular acceleration can be calculated from the polished rod acceleration values. The described method is the most basic numerical method, therefore it has larger error compared to the previously detailed methods in Chapter 5.5.1 and Chapter 5.5.2, but its most important advantage is the elimination of systematic errors. Due to the nature of the numerical differentiation, the resulting polished rod acceleration values are valid at the measured times registered in the dynamometer survey, except two missing values, one at the start and one at the end of the array. This is not a critical problem, since usually the measured points in one stroke are in the hundreds range, and the results are used for justifying the results of other, more complex – and therefore more accurate – methods, if their results show good correlation.

5.5.4 Comparison of the Calculation Methods

Figure 39 contains the beam acceleration data calculated using the three previously detailed methods along with the results of the TWM software. The strong correlation between the results of the simple numerical method and the two more sophisticated methods is clearly supported based on their visual representation, the correlation parameter is 0.9615 between the numerical data set and the results of the calculation based on the method proposed by Svinos. The correlation between the two more accurate methods is 0.9868, this means that the results of the methods are nearly identical. Based on this analysis, the result of the Svinos and Gibbs methods are accepted.

The number of Fourier coefficients used in the Gibbs method is sufficient based on the comparison with the exact calculation method developed by Svinos. Since the application of the Fourier series is much less cumbersome in the method proposed by Gibbs than the calculations required by the Svinos method, the usage of the former method is recommended to find the angular acceleration pattern of the walking beam.

The raw beam angular accelerations of the TWM software are acceptable, since they nearly coincide with the numerically calculated values, however, the filtered acceleration function is not properly calculated, as shown in **Figure 39**.

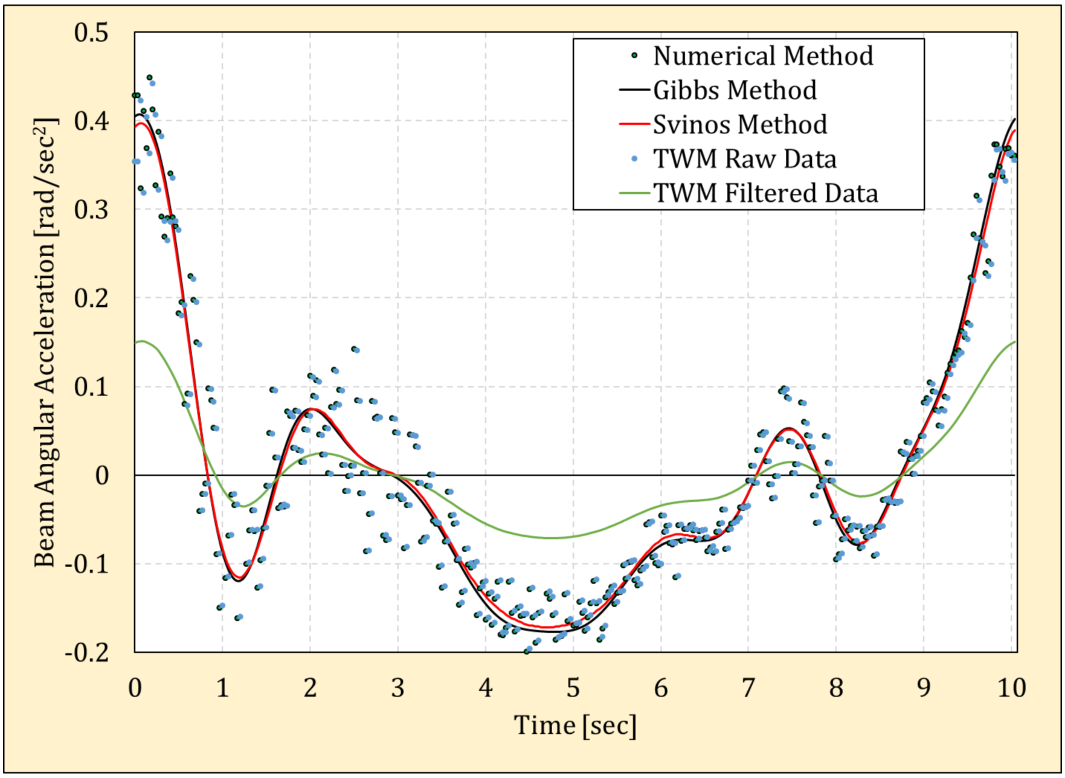


Figure 39 *Comparison of models calculating the beam angular acceleration*

6 Achieving Optimal Counterbalancing

Different theories on the optimal net gearbox torque are detailed in this chapter based on extensive literature research. After the discussion on the different optimization principles, the objective of this chapter is to provide the counterweight configuration corresponding to the best net gearbox torque variation throughout the pumping cycle for the investigated crank balanced sucker-rod pumping unit. By changing the counterweight configuration valid at the dynamometer measurement to the optimum arrangement the operation of the sucker-rod pumping unit can be improved significantly.

For this purpose, an artificial intelligence program was developed in C# programming language, due to the complexity of the emerging optimization problem. Screenshots of the program, the input and output files are included in Appendix A. Appendix B contains the most relevant parts of the source code.

The brute force method of checking every counterweight configuration is futile, since the total number of cases for the example problem is between $2 \cdot 10^{17}$ and $6 \cdot 10^{17}$ in the asymmetrical counterbalancing case. These boundaries were calculated based on the number of applicable counterweights on either side the crank arms, the travel of each counterweight on the crank arm, and the number of auxiliary weights on each main counterweight. There were 10 different applicable counterweight types, as shown in **Table 3**, resulting in 11 different cases in total by including the scenario without a main counterweight on the selected side of the crank arm. The travel of the main counterweights varies between 63.77 in and 84.58 in, the increment of the counterweight position was set to 0.1 in, resulting in 638 and 846 different positions, respectively. On each main counterweight maximum 2 auxiliary weights were allowed in the optimization procedure, resulting in 3 different cases for each counterweight.

6.1 Theoretical Background of Torque Optimization

The optimization of the net mechanical gearbox torque seems to be a well discussed problem due to the fact that the torque loading of the pumping unit determines the energy requirement, and therefore the cost of the oil production. However, some new achievements are shown in this chapter regarding the selection of the appropriate optimization procedure.

6.1.1 Optimization of the Maximum Net Gearbox Torque

The first optimization criterion was discussed as early as 1943 by (Kemler, 1943). The result of not having optimal counterbalancing results in energy being wasted and in some cases can lead to equipment damage due to overloading. The optimum counterbalancing means that the rod torque is offset in the greatest extent possible, resulting in the minimum net gearbox torque and therefore minimizing the peak torsional loading on the prime mover. (Richards, 1957) The corresponding counterweight configuration is found by selecting the counterbalance torque that equalizes the peaks of the net gearbox torque in the upstroke and downstroke. (Takács, 2015) As discussed in (Rowlan, McCoy, & Podio, 2005) in a balanced operation the peaks of the net gearbox torque function in the upstroke and downstroke are approximately equal.

During the optimization, the changes in the rotary inertial torque should be considered in addition to the changes of the counterbalance torque to improve accuracy. Previous works did not include the in-depth investigation of asymmetrical counterweight configurations in the torque optimization procedure. If identical counterweights are used to counterbalance the pumping unit, only the magnitude of the counterbalance torque is affected by their respective placements on the crank arms, as shown in **Figure 19**. By using an asymmetrical counterweight configuration, the secondary phase angle, τ' has to be considered, as shown in **Figure 21**. By having this new degree of freedom in the optimization, the net gearbox torque can have the same maximum value at three different times in one pumping cycle. This results in a smaller peak net gearbox torque compared to using identical counterweights.

6.1.2 Optimization of the Cyclic Load Factor

The calculation method presented by (Takács, 1990) focuses on introducing a more advanced calculation method to produce the optimal counterbalance torque than the one specified in the API Spec 11E (API, 2008). The objective of this optimization procedure is to achieve the smallest cyclic load factor (CLF) using an iterative method; CLF is defined by Equation *47*. The merit behind this optimum is that the lowest power requirement by the prime mover is obtained at the minimal CLF value. Using the least amount of energy to produce a given liquid regime increases the profitability of the oil production.

$$CLF = \frac{\sqrt{\frac{\int_{\theta=0}^{2\pi} \left(T_{net}(\theta)\right)^2 \mathrm{d}\theta}{2\pi}}}{\frac{\int_{\theta=0}^{2\pi} T_{net}(\theta) \mathrm{d}\theta}{2\pi}} \tag{47}$$

where:

CLF	Cyclic load factor [-], and
$T_{net}(\theta)$	Net gearbox torque versus crank angle function [in lb].

The cost-efficiency of the sucker-rod pumping can be greatly increased using the proper counterbalancing of the unit. In (Takács, 1990) the optimized result improved the CLF of the investigated unit from 1.594 to 1.400, and the overloading of the gearbox from 157.5% to 123.3%. By optimizing for a different objective – reducing the peak net gearbox torque – the overloading of the unit could have been reduced below 123.3%. This condition slightly increases the cost of pumping but improves the lifetime of the gearbox substantively, as shown in **Figure 4**.

6.1.3 *Introduction of the Modified Cyclic Load Factor*

A modified CLF parameter was developed, that generalizes Equation *47* by considering the varying crank angular acceleration in time. Using Equation 48, the torque optimization of sucker-rod pumping units with varying crank angular speeds is improved.

$$CLF_{mod} = \frac{\sqrt{\frac{\int_{t=0}^{T}\left(T_{net}(t)\right)^2 \mathrm{d}t}{T}}}{\frac{\int_{t=0}^{T} T_{net}(t)\mathrm{d}t}{T}} \tag{48}$$

where:

CLF_{mod}	Modified cyclic load factor [-],
$T_{net}(t)$	Net gearbox torque variation in time [in lb], and
T	Period time of the investigated pumping unit [sec].

In the past Equation (47) was used mainly because the crank angle was the basis of the torque analysis, every parameter was calculated at equally distributed crank angle values. In these cases, the constant increase of the crank angle function was assumed. The basis of Equation (48) is time, therefore this new equation is capable to consider the precisely calculated crank angle variation throughout the pumping cycle.

6.2 Change of Crank Acceleration due to Different Counterbalancing

By modifying the counterweight configuration, the acceleration pattern of the walking beam and the crank arm will change, however, this effect cannot be determined from only one dynamometer measurement. The operation of sucker-rod pumping systems is too complex for the exact determination of these variations. Based on two dynamometer measurements carried out on a M-640D-305-192 unit – its properties are shown in **Table 6** – before and after the counterweight modification, the acceleration patterns are compared.

Table 6 *Input data for the pumping unit in the investigation of the change in crank angular acceleration*

Pumping unit designation	M-640D-305-192
Manufacturer	Lufkin
Geometry type	Mark II
Maximum torque loading of the gearbox	640,000 in lb
Maximum polished rod load	30,500 lb
Nominal stroke length	192 in
Structural unbalance	-7,160 lb

Crank type	192130MRO
Gearbox mass moment of inertia	3,920 lb_m ft^2
Beam mass moment of inertia	4,621,470 lb_m ft^2
Rotation	Counterclockwise
Counterweights	4pcs. OARO, placed 0 in from long end of crank
Crank moment	905,690 in lb
Crank mass moment of inertia (2 cranks)	788,968 lb_m ft^2
Crank length	130 in
Crank half-width	16 in
Pumping speed	6.32 SPM

In the original case 4pcs. OARO counterweights were placed 0 in from the long end of the crank arm. After the counterbalance optimization of the TWM software the main counterweights were moved 3.25 in towards the crankshaft and 4pcs. OAS auxiliary counterweights were installed to increase the counterbalance torque. The net torque curves and the crank angular acceleration curves before and after the modification of the counterweight configuration are shown in **Figure 40**.

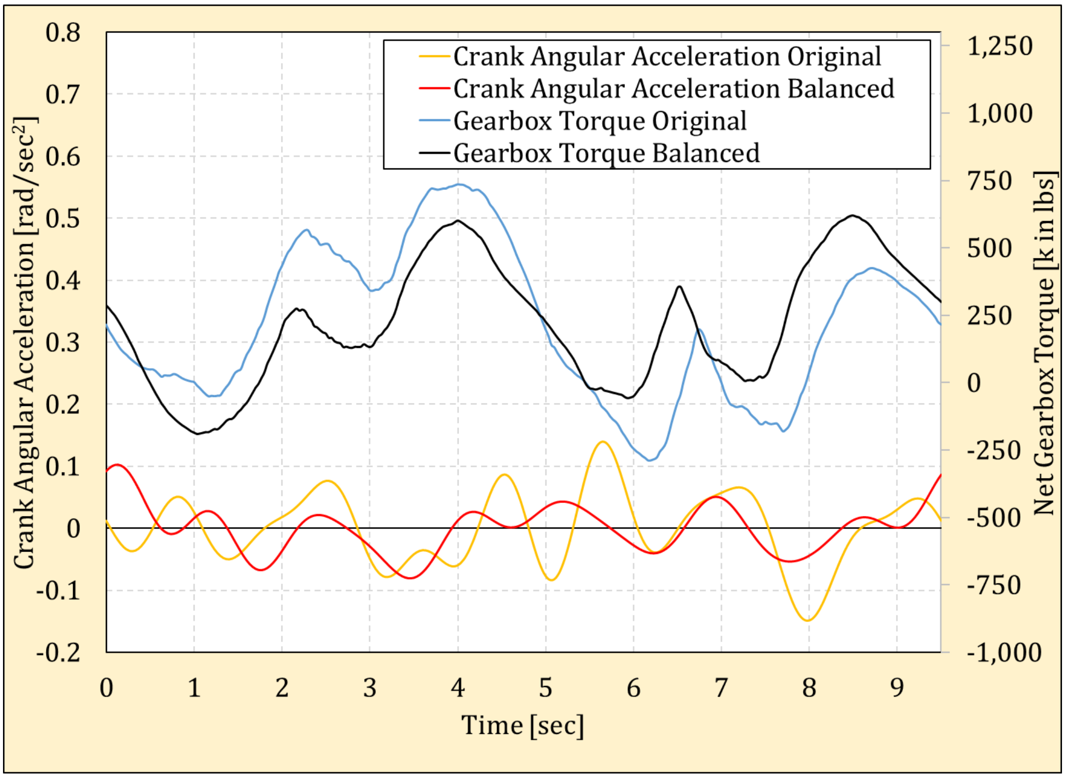


***Figure 40** Effect of different counterweight configuration on the crank acceleration*

When measuring correlation between two data series, Equation 49 is used to get a quantitative result. If the value is 1, there is a stochastic positive relationship between the two data series. At -1 correlation value, there is a negative and strong connection. As the correlation value approaches 0, it indicates a weak or no correlation between the two investigated data series. (Microsoft, 2019)

$$Correlation = \frac{\sum_{i=1}^{N}\left((x_i - \bar{x}) \cdot (y_i - \bar{y})\right)}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \cdot \sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{49}$$

where:

N	number of data points [-],
x_i, y_i	i^{th} element of the data series [var.], and
$\bar{x}, \bar{y}$	average of the respective data series [var.].

The correlation between the changes in the net gearbox torque and the differences in the crank and beam acceleration patterns are -0.16 and 0.085, respectively. These values represent a poor correlation. As the peak torque decreases in the balanced case, the resulting crank angular acceleration function also has lower peak values, but since the correlation is not strong enough, the new crank angular acceleration function cannot be approximated using the initial crank angle variation and the two net torque variations throughout the pumping cycle. Creating a calculation procedure capable of executing the aforementioned task would increase the accuracy of the net gearbox torque optimization procedures. Since such a method is not available, the crank angular acceleration values determined from the dynamometer survey are used to find the inertial torques under different counterbalancing conditions.

6.3 Particle Swarm Optimization Technique

The particle swarm optimization (PSO) algorithms are metaheuristic artificial intelligence techniques, that use an iterative process to find the optimum to a given problem. There are numerous different methods in this group, their different properties enable the engineers and mathematicians to solve a wide variety of optimization problems by selecting the proper type. (Engelbrecht, 2007)

6.3.1 General Properties of the PSO Method

The use of a PSO technique is preferred, when the direct calculation of the optimum condition is not possible, and when the other multi-dimensional algorithms fail to find the global optimum because of the high number of the local optima in the solution space. Another advantage of this method is its flexibility. The general optimization method can be customized with little effort to the solve the task at hand effectively by either modifying the calculation procedure, or changing the constants used in the method to create an improved optimization process. This algorithm provides a solution even if the specified fitness function is not continuously differentiable. Due to the nature of the method, the global optimum is not guaranteed to be the result of the calculation, but the results are better than any direct calculation method available. (Eberhart & Kennedy, 2001)

The method uses a given number of candidates, improving their position in the solution space in each calculation step. The determination of the new positions is carried out by minimizing the fitness – an error function value – of the candidates. The visualization of this step-by-step improvement shows remarkable resemblance to the movement of flock of birds, or school of fish. (Fernández-Martínez, 2012) **Figure 41** shows the simplified flowchart of the applied PSO algorithm.

As the first step of the optimization procedure the solution space is populated with particles, then the fitness value of every candidate solution is determined. The initialization of the particles is usually carried out by randomly generating their positions, independently from each other. Each particle is defined by a vector; its coordinates define the position of the particle in the respective dimensions. The dimension of the required vector is determined by the number of independent variables in the optimization procedure.

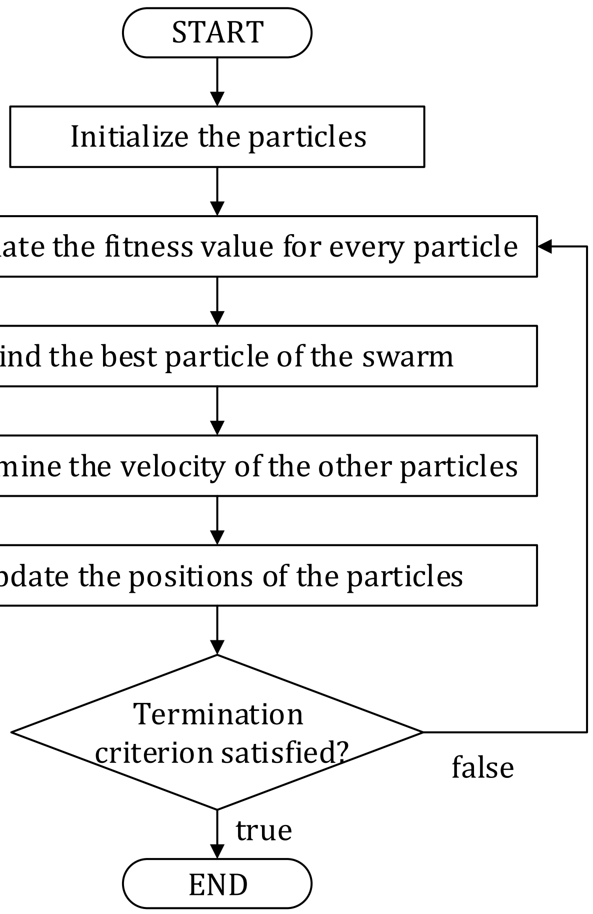


Figure 41 *General flowchart of the particle swarm optimization method*

The formula that defines the velocity vector for every particle is customizable to produce a robust optimization procedure for the selected task. The objective of the following iteration steps is to improve the global best fitness value. A termination criterion is specified to end the calculation process. This constraint is usually the number of iteration steps, but this condition can be tailored to handle the specific optimization task at hand.

The customizability is one of the main advantages of the PSO algorithm. The intervals from which the parameters can take value is not always constant. Another great benefit of the presented calculation method is the relatively easy modification of the optimization goal. Changing the procedure to produce the fitness value for every point using a different error function is straightforward.

6.3.2 Using the PSO Algorithm in the Net Gearbox Torque Optimization of Sucker-Rod Pumping Units

The previously introduced general PSO algorithm is customized to handle the necessary optimizations introduced in Chapter 6.1. The initialization of the particles is done by randomly generating their position in the investigated hyperdimensional space. Every component of their positions are generated independently using a uniform distribution within the boundaries of the respective dimension. Every component of the position vector must be non-negative, the upper limit is constant for the main counterweight type and for the number of auxiliary weights used. The upper boundary of the counterweight distance from the long end of the crank depends on the crank and the main counterweight used. When the main counterweight type is changed, the upper boundary of its position must be determined using data similar to **Table 3**.

The number of particles used in the optimization procedure mainly depends on the smoothness of the search space. For smooth surfaces smaller swarm sizes are sufficient, usually 30 particles provide the optimum solution in these cases. (Engelbrecht, 2007) However, in the optimization of the net gearbox torque the fitness function is discontinuous with numerous local optima. Based on the results of multiple test runs the swarm size was set to 500. A smaller number of particles provided inferior results even with increased number of iteration steps. Using more particles provided nearly identical results with increased simulation times.

The calculation of the fitness value for every particle is carried out using the criteria introduced in Chapter 6.1. The fitness functions are the maximum net gearbox torque in the pumping cycle, and the modified cyclic load factor, introduced in Chapter 6.1.1 and Chapter 6.1.3, respectively. If the pumping speed varies during the pumping cycle, the changes in the counterweights type and positions alters the value of I_S. Therefore, during the calculation of the new counterbalance torque the rotary inertial torque must be determined with the new mass moment of inertia. This circumstance makes the process more complex than the previous optimization methods, which neglected the inertial torques.

The global best solution is then selected, that will attract the other points to produce an improved counterweight configuration. The next step in the procedure is the determination of the velocity of every particle using Equation 50. The relevant parameters to find the velocity vector are the current position, the position corresponding to the lowest fitness value the selected candidate ever had, and the global best position in the current iteration step.

$$V_{i+1,j} = W \cdot V_{i,j} + C_1 \cdot Rnd_1 \cdot \left(BP_{i,j} - P_{i,j}\right) + C_2 \cdot Rnd_2 \cdot \left(GBP_i - P_{i,j}\right) \qquad (50)$$

where:

$V_{i,j}$	j^{th} velocity component of a particle in the i^{th} iteration step [-],
W	Damping factor [-],
C_1, C_2	Acceleration coefficients [-],
Rnd_1, Rnd_2	Random numbers from [0,1] [-],
$BP_{i,j}$	j^{th} component of the best position of a particle in the i^{th} iteration step [-],
$P_{i,j}$	j^{th} component of the position of a particle in the i^{th} iteration step [-], and
GBP_j	j^{th} component of the global best position [-].

The damping factor decreases the maximum vector length at every iteration, ensuring the convergence of the optimization. For the investigated torque optimization problems, a damping factor of 0.99 provided a good convergence; if a smaller number is used, the particles initially distant from the global best position cannot travel through the solution space, therefore the optimization procedure can end prematurely.

The acceleration coefficients control the behavior of the particles, C_1 considers the particles attraction to its own best position, C_2 determines the effect of the global best position on the particle. Their ideal absolute and relative values depend on the optimization task, usually a similar pair of values provide a robust and efficient calculation procedure. (Engelbrecht, 2007) Both of these parameters were set to 2 after series of testing, with these values the maximum velocity component was ideal. With greater acceleration values the particles would have greater velocities and therefore could miss optimum solutions on their trajectories. If smaller numbers were used, the required number of iteration steps had to be increased to achieve similar accuracy.

The random numbers – Rnd_1 and Rnd_2 – included in Equation 50 create a more robust and versatile optimization procedure by adding uncertainty to the stochastic nature of the equation. These variables are chosen randomly and independently from the $[0, 1]$ interval.

Maximum and minimum values can be specified for every component of the calculated velocity vector, and the resulting position coordinates. While solving the example problem, the upper limit for the velocity vector was set to 10. Using a hard limit ensures that the distant particles from the current best position will not immediately move to its local vicinity and therefore possibly missing better solutions in the process.

The termination criterion was specified by the allowed number of iteration steps. During extensive testing of the introduced PSO program, 30 iteration steps proved to be sufficient to find the optimal counterweight configuration considering the constraints of the optimization process.

The optimization of the net gearbox torque is a complex task if all the relevant torque components are considered. Even in the symmetrical counterweight configuration case there are three independent variables: the weight of the counterweights, the number of the used auxiliary counterweights and the counterweight placement from the long end of the crank. In the asymmetrically placed counterweight case however, the number of independent parameters rises to twelve: the type of the main counterweight, the number of the used auxiliary counterweights and the distance of the main counterweight from the end of the crank for both sides for both cranks independently. A twelve-dimensional vector contains these data; therefore, the optimization of the mechanical net gearbox torque has to be carried out in a twelve-dimensional solution space. Every combination of the coefficients in the vector will alter the resulting net torque variation during the production cycle.

Depending on the type of the main counterweight, the maximum travel distance on the crank arm is defined in **Table 3**. This is the upper boundary of the counterweight displacement used in the optimization, it must be reconsidered at each calculation step. Additional constraints – like only allowing counterweights from the same type with different positions on the crank arm – are implemented with little effort.

6.3.3 Investigating a Particle in the PSO Algorithm of the Example Problem

The calculation procedure defined in **Figure 41** and detailed in Chapter 6.3.2 is illustrated with the investigation of a selected particle in asymmetrical counterbalancing case using the peak net gearbox torque as the optimization criterion. The selected particle is randomly generated in the twelve-dimensional solution space considering the proper upper and lower boundaries for each component of its position.

The first four elements of the vector determine the types of the main counterweights used. Therefore, the fitness function is not continuously differentiable, it has discontinuities at every main counterweight type change in these four dimensions. To find the type of the main counterweight from the corresponding vector component, its value is rounded to the nearest integer. The following four vector coordinates define the counterweights' distance from the long end of the crank (D in **Figure 17**). The last four coordinates give the number of auxiliary weights used on the main counterweights, limited to 2 pcs. The fitness function has discontinuities in these four dimensions as well.

The procedure of the PSO optimization to find the optimal net gearbox torque is illustrated in **Table 7**. After randomly generating the first candidate – shown in the first column – its fitness value is determined using the torque determination process detailed in Chapter 5. In total, 500 particles are generated randomly, the best one in the first iteration step is introduced in the second column. The candidate with the global best position will attract every other particle based on their corresponding distance. The velocity vector calculated using Equation 50 is included in the third column of **Table 7**.

Even though the first four vector components represent the counterweight types, numerical values must be used as vector coordinates in the calculation procedure. The numerical values in these cases were rounded to the nearest integer and the counterweight types were selected. In this case 0 meant that no counterweight was used on the specific side of the crank arm. The index of the counterweights in **Table 3** was used to convert the numerical values into the counterweight types.

The position of the investigated particle in the second iteration step is determined using the calculated velocity vector and its initial position. The fourth column of **Table 7** contains the new position, resulting in a smaller fitness value compared to its initial state. The improvement of the fitness value is not necessarily true for every particle at every calculation step, but due to the robust nature of the algorithm, both the global best fitness value and the average fitness value tends to decrease with every successful iteration step.

During this investigation no additional constraints were used for the position coordinates of the investigated particle. The implementation of such a limiting factor e.g. specifying the usage of identical counterweights is added easily to the optimization procedure.

Table 7 *Detailed solution of the Example Problem with the PSO Algorithm*

	Initial Position of the Investigated Particle		Best Particle in the First Iteration Step		Velocity of the Investigated Particle	Position of the Investigated Particle After the First Iteration Step	
	Numerical value	Used value	Numerical value	Used value	Numerical value	Numerical value	Used value
1st CW. Type	4.78	5 (3CRO)	7.82	8 (OARO)	1.91	6.69	7 (1RO)
2nd CW. Type	5.21	5 (3CRO)	8.13	8 (OARO)	0.21	5.42	5 (3CRO)
3rd CW. Type	8.34	8 (OARO)	7.62	8 (OARO)	-0.14	8.2	8 (OARO)
4th CW. Type	3.89	4 (5ARO)	6.94	7 (1RO)	6.94	10.83	10 (OORO)
1st CW. Distance	50.3	50.3 in	14.26	14.3 in	-10	40.3	40.3 in
2nd CW. Distance	61.11	61.1 in	11.58	11.6 in	-10	51.11	51.1 in
3rd CW. Distance	4.06	4.1 in	10.16	10.2 in	9.91	13.97	14 in
4th CW. Distance	72.68	72.7 in	4.92	4.9 in	-10	62.68	62.7 in
1st CW. No. Aux Weights	0.32	-	1.45	1 pcs. OAS	0.51	0.83	1 pcs. 1S
2nd CW. No. Aux Weights	1.73	2 pcs. 3BS	1.03	1 pcs. OAS	-0.68	1.05	1 pcs. 3BS
3rd CW. No. Aux Weights	1.12	1 pcs. OAS	0.67	1 pcs. OAS	0.32	1.44	1 pcs. OAS
4th CW. No. Aux Weights	1.37	1 pcs. 5CS	1.98	2 pcs. 1S	0.02	1.39	1 pcs. OOS
Peak Net Torque [k in lbs] (Fitness Value)	1022.03	**1022**	504.95	**505**		988.03	**988**

At the end of every iteration step the best position is determined and is compared to the global best position in the previous calculation step. The global best position is replaced, when a new position is found with smaller fitness value. **Figure 42** shows the evolution of the peak net gearbox torque with the iteration steps. In total 30 iterations were carried out, the solution was achieved after the 23th calculation step.

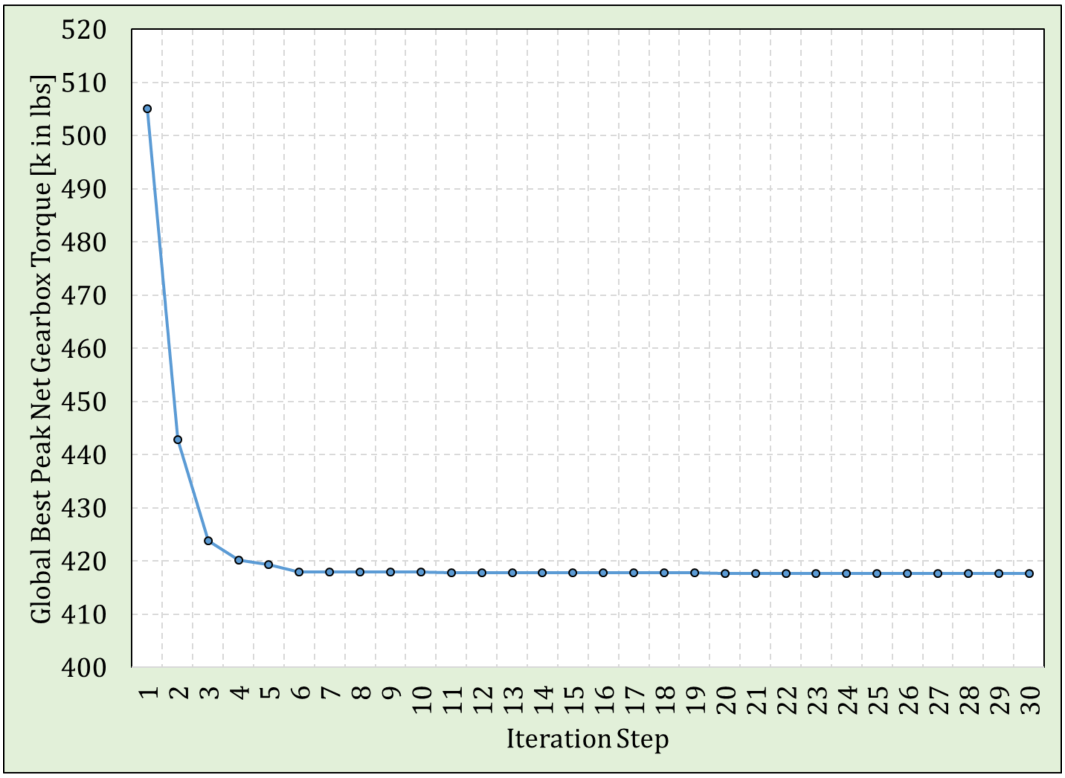


***Figure 42** The improvement in the peak net gearbox torque value with the iteration steps*

6.4 Sensitivity Analysis

A traditional sensitivity analysis cannot be carried out because the applicable counterweights have discrete masses and moments of inertia. For illustration purposes only the simplest analysis can be presented, since in the introduced asymmetrical counterbalancing case the number of the relevant dimensions is 12. Therefore, the representation of the parameter sensitivity is shown for the symmetrical counterweight configuration only, with fixed number of auxiliary weights. In this special case there are only two independent parameters: the type of the main counterweights and their displacement from the long end of the crank. **Figure 43** shows the results for the sensitivity analysis, where 2 auxiliary weights are used, the position of the main counterweights is investigated between 0 and 59 in from the long end of the crank. In this figure the peak net gearbox torque is shown as a function of the applied counterweights and their respective position. It is clearly visible, that in this oversimplified case there are multiple local optima; the determination of the global optimum is difficult. The data used to create **Figure 43** and a 3D representation is included in Appendix C. The number of local optima increases rapidly as more independent parameters allowed to influence the maximum net gearbox torque. The introduced figure supports the previous assumptions on the necessity of a numerical calculation method in the torque optimization procedure.

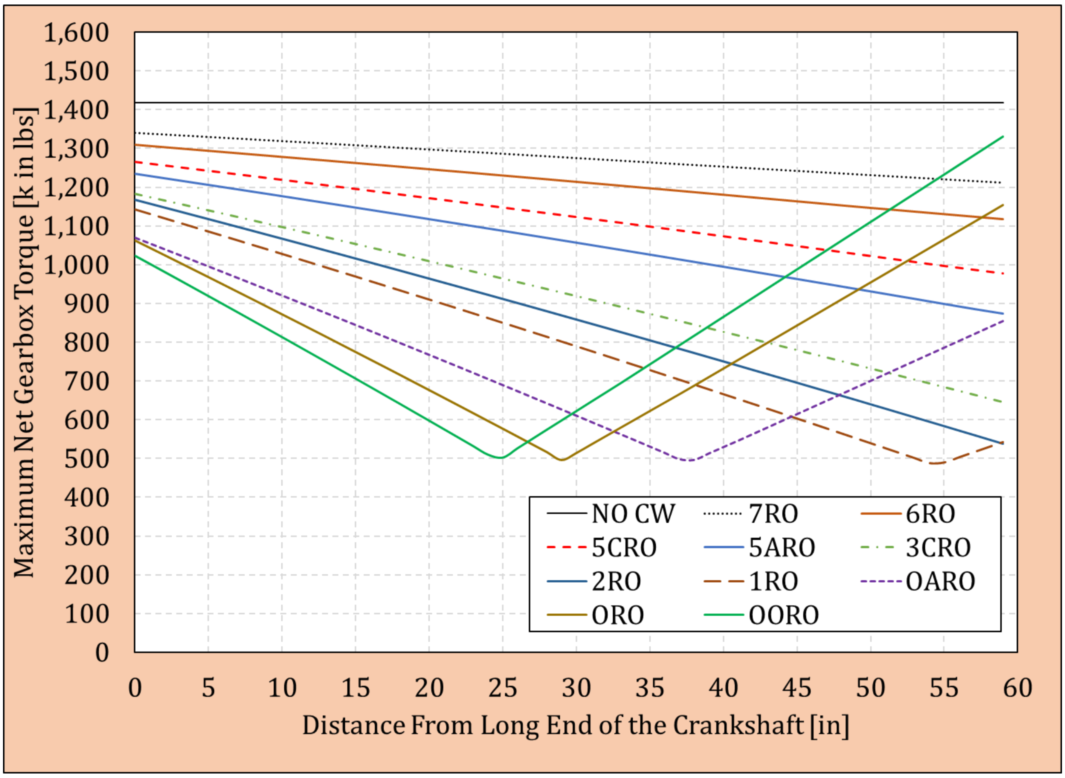


Figure 43 Results of the sensitivity analysis in the simplest symmetrical case

6.5 Finding the Optimum Counterweight Configuration

The original counterweight configuration is 4pcs. ORO main counterweights placed at 10 in from the long end of the crank arm, shown in **Table 1**. The resulting peak net gearbox torque is 597.3 k in lbs, shown in **Figure 44**. The value of the calculated CLF_{mod} is 1.415. The calculation procedure detailed in Chapter 6.3 is used to produce the optimal counterweight configurations along with the optimal net gearbox torque functions with the specified constraints in the optimization procedure.

6.5.1 Optimization of the Peak Net Gearbox Torque

6.5.1.1 Using Identical Counterweights

In this case the counterweights and the auxiliary weights must be identical. When only allowing symmetrical counterbalancing, the placement of the main counterweights cannot differ. A different position of one counterweight only changes the amplitude of the counterbalance torque and the rotary inertial torque, therefore the investigation of a symmetrical solution is sufficient, because there is no benefit placing the same counterweights at different positions on the crank arm.

The optimal symmetrical counterweight configuration is found to be 4pcs. OORO main counterweights with 4 pcs. OOS auxiliary counterweights, placed at 25.9 in from the long end of the crank arm. The maximum net gearbox torque is 491.8 k in lbs, the CLF_{mod} is 1.397. The net torque variation for the original case and the optimized case are shown in **Figure 44**.

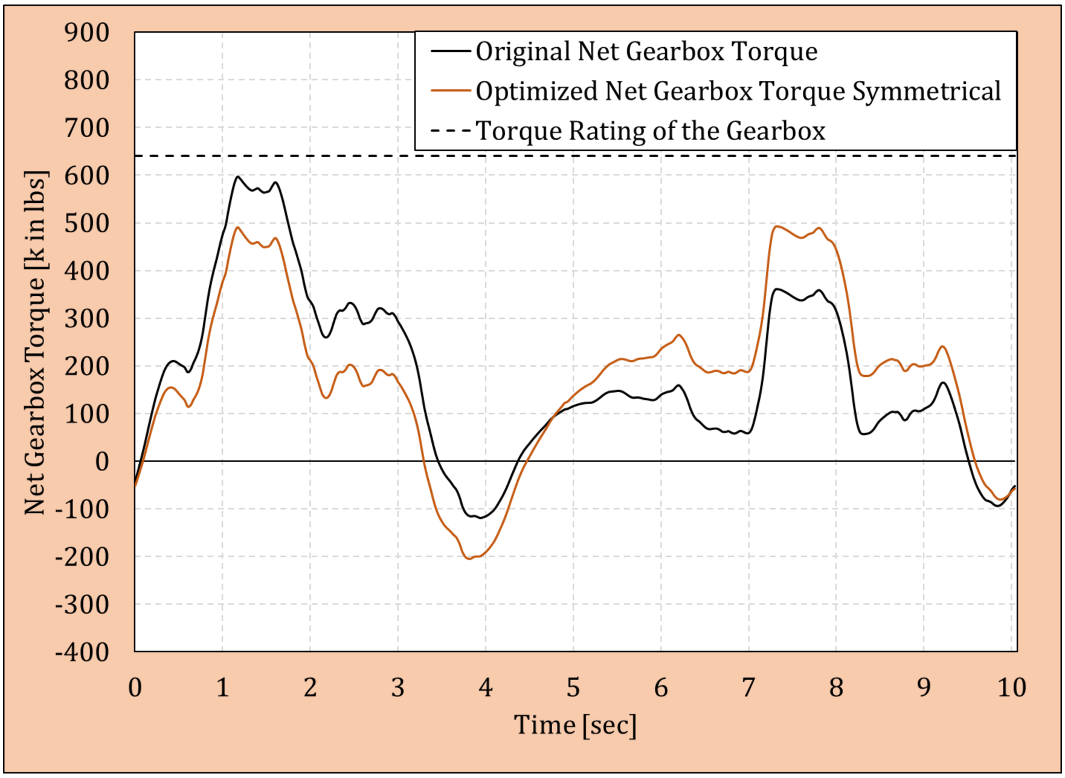


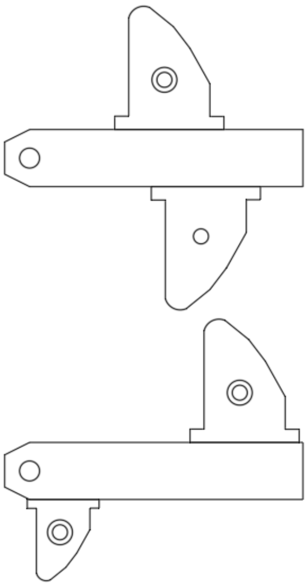
Figure 44 *Optimum net gearbox torque using symmetrical counterweight configuration*

6.5.1.2 Using Different Counterweights

The optimal asymmetrical counterweight configuration is included in **Table 8.** No restrictions were used in this scenario to limit the calculation process, all twelve parameters shown in **Table 7** can change independently.

Table 8 *Asymmetrical optimum counterweight configuration*

Position	Main CW	Auxiliary CW	Distance from long end of crank
1st crank top	OORO	2 pcs. OOS	31.8 in
1st crank bottom	OORO	1 pc. OOS	17.1 in
2nd crank top	OORO	2 pcs. OOS	1.49 in
2nd crank bottom	5ARO	2 pcs. 5S	82 in



The resulting secondary phase angle is 7.43 deg, the maximum net gearbox torque is 418.2 k in lbs, the CLF_{mod} is 1.429. The net torque variation is shown with blue in **Figure 45**. The phase shift of the counterbalance torque causes the net gearbox torque to have 3 maximum points instead of 2 in the symmetrical case. This lowers the peak net torque by 73.6 k in lbs, which is nearly 11.5% of the rating of the gear reducer. It is important to consider the drop of the minimum net gearbox torque since the negative torques can also overload the gear reducer if the rating is exceeded.

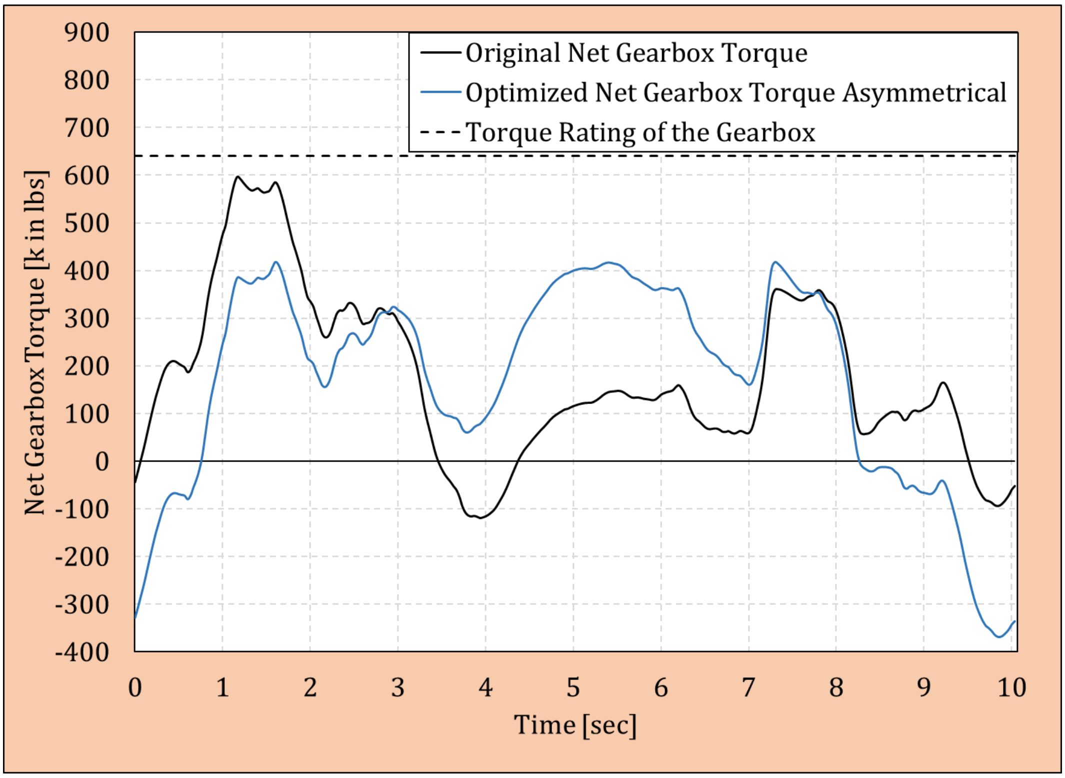


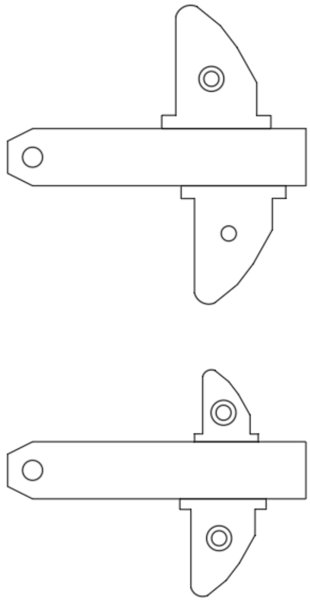
Figure 45 *Optimum net gearbox torque using asymmetrical counterweight configuration*

6.5.2 Optimization of the Modified Cyclic Load Factor

In this case the fitness value is determined based on the CLF_{mod} value calculated from the twelve-dimensional arrays used in the PSO calculation procedure. The counterweight configuration resulting in the minimum CLF_{mod} is included in **Table 9**. The resulting secondary phase angle is -0.03 deg, the maximum net gearbox torque is 484.1 k in lb, the CLF_{mod} is 1.386. The net gearbox torque variation is shown in **Figure 46**.

Table 9 *Counterweight configuration providing minimum CLF_{mod}*

Position	Main CW	Auxiliary CW	Distance from long end of crank
1st crank top	OORO	2 pcs. OOS	14.1 in
1st crank bottom	ORO	1 pc. OS	8.1 in
2nd crank top	5CRO	2pcs. 5CS	19.2 in
2nd crank bottom	1RO	2 pcs. 1S	16 in



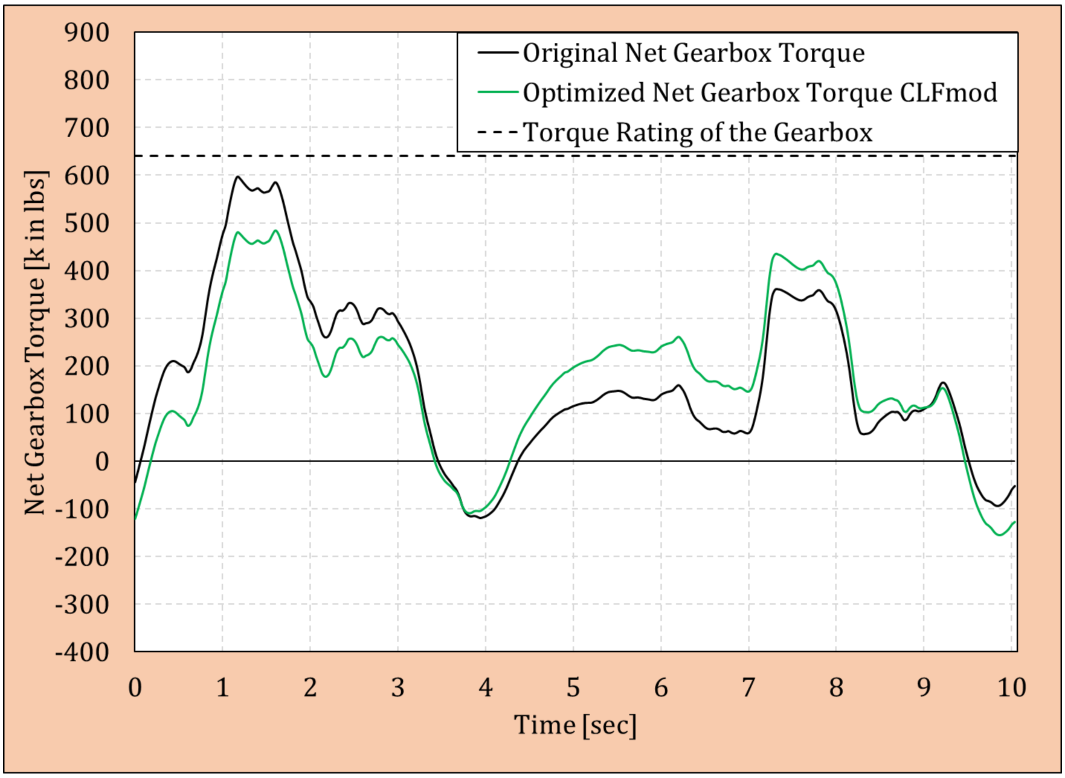


Figure 46 *Torque optimization producing minimal modified cyclic load factor*

6.6 Comparison with TWM Optimization

Figure 47 shows the proposed symmetrical and asymmetrical optimized net torque variation along with the results of the TWM software. By incorrectly neglecting the inertial torques from the torque analysis, and only investigating symmetrical counterweight configurations, TWM gives an optimum peak net torque of 597.3 k in lbs with a 1.42 CLF_{mod} value. By improving the evaluation of the dynamometer survey and the calculation of the mechanical net gearbox torque the resulting solution describes the real operating conditions more accurately. Using these data, the optimization procedure gives more reliable optimum counterweight configurations.

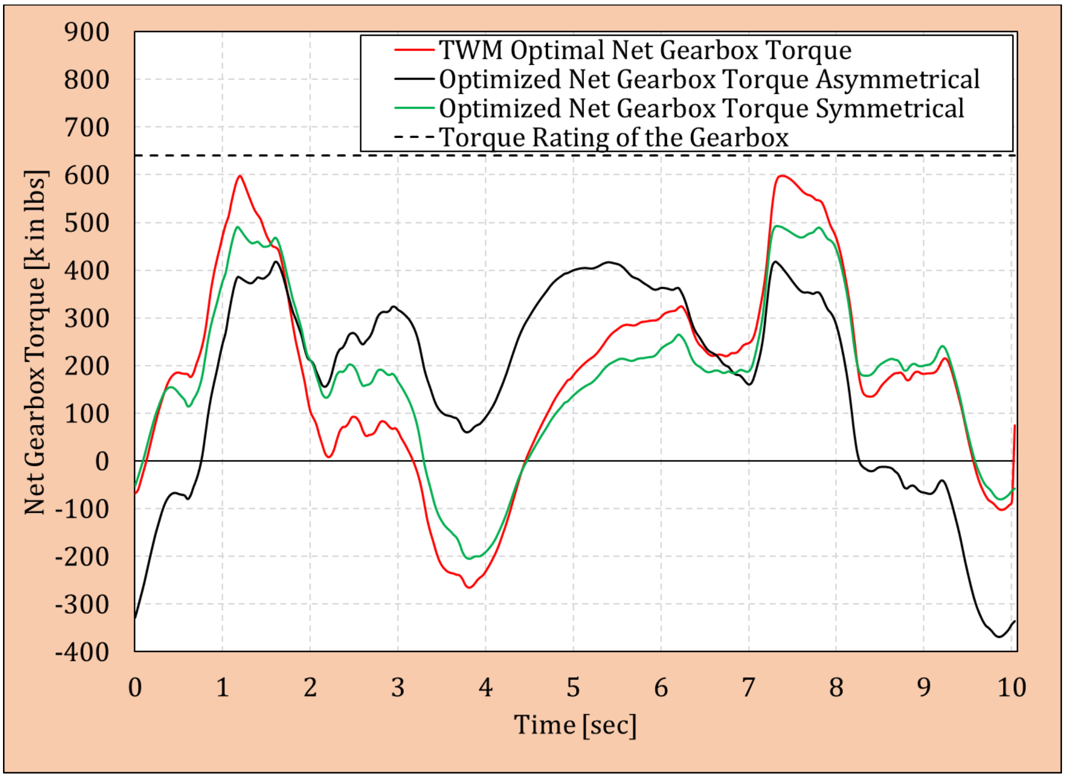


Figure 47 *Comparison of the torque optimization with TWM results*

6.7 Conclusions of the Optimization Procedures

Optimizing the net gearbox torque of a sucker-rod pumping unit is essential to prevent overloading and to save operating costs. **Table 10** contains the results of the torque optimization carried out on the example problem. The optimum result provided by the TWM software neglects the inertial effects, therefore it mischaracterizes the net gearbox torque.

The optimization procedure developed creates the optimum net gearbox torque with different constraints on the corresponding counterweight configuration. The introduced symmetrical counterweight configuration provides a slightly higher peak net torque, but smaller modified cyclic load factor compared to the asymmetrical case. The counterweight configuration corresponding to the minimal modified cyclic factor in the investigated cases does not provide significantly better results, than the symmetrical counterweight configuration.

Table 10 ***Summary of the optimization results***

	Optimization Objective	Peak Net Gearbox Torque [k in lbs]	CLF_{mod} [-]	τ' [deg]
Original Case	-	597.31	1.415	0
TWM Optimum Result*	-	597.33	1.420	0
Results of the Optimization				
Identical Counterweights and Positions	Peak Torque	491.80	1.397	0
No Constraint in the Optimization	Peak Torque	418.20	1.429	7.43
	CLF_{mod}	484.09	1.386	-0.03
Same Counterweight Configuration on Both Cranks	Peak Torque	419.20	1.555	10.69

If the overloading of the pumping unit can be prevented by solving the optimization problem using the same main and auxiliary counterweights, the symmetrical optimum counterweight configuration is recommended. However, if the symmetrically placed counterweights cannot reduce the peak net torque acting on the gearbox below its torque rating, using non-identical counterweights can prevent overloading. The proper asymmetrical counterweight configuration will always result in a lower peak net gearbox torque value, compared to the symmetrical cases.

6.8 Further research possibilities

There are possible future research paths based on the introduced calculation procedures. The exact determination of the change in the crank and beam angular acceleration as a function of the net gearbox torque would be a great addition, but it seems unlikely, that a general solution exists for said problem.

The incorporation of the proposed asymmetrical counterbalancing calculations in works like (Konz, 2018) would be beneficial. Using the introduced methods to update the software evaluating the dynamometer surveys could result in more favorable operating conditions for sucker-rod pumping units. The calculation procedure presented can be modified to improve the results of a multi-balance technology introduced in (Feng, Ding, & Jiang, 2015).

The method introduced can be modified and applied to sucker-rod pumping units with variable speed drives, further improving their efficiency. For this, however, further study of the complex interactions between the controlled crank angular acceleration pattern by the used microcontroller and the resulting net gearbox torque function is needed.

7 New Scientific Results

7.1 Thesis 1

A successive approximation procedure was introduced in Chapter 5.3.2 that produces the crank angle values corresponding to the measured polished rod positions with a higher accuracy than previously existing methods. Since the crank angle variation in time is not measured by a dynamometer survey, it must be calculated from the measured polished rod positions and the kinematic parameters of the sucker-rod pumping unit. The exact calculation procedure developed here has a high importance because any errors in the crank angle vs time function affect almost every other parameter in the evaluation of the torque conditions of sucker rod pumping units. By minimizing the error in the first calculation step, the accuracy of torque calculations as well as counterbalance optimizations are improved.

7.2 Thesis 2

A complex calculation method was developed in Chapter 5.4 and 5.5, that produces the crank angular velocity, the crank angular acceleration and the beam angular acceleration variation throughout the pumping cycle. The proposed method has superior precision compared to the most widely used software in the industry. The numerical calculation models presented have proved to be strong validating tools to help verify the results of the more complex, but cumbersome calculation methods.

7.3 Thesis 3

The effects of asymmetrical counterweight configurations on the counterbalance torque vs time function were investigated; that is an often ignored condition in the professional literature. Asymmetrical counterweight placement affects the net gearbox torque vs time function. In this work a secondary phase angle – τ' – was introduced to adequately describe the deviation of the counterbalance torque from the symmetrical cases. The new equations developed in Chapter 5.2.4 permit the accurate calculation of inertial torques and were incorporated in the gearbox torque optimization procedures introduced.

7.4 Thesis 4

A novel technique to solve the optimization of gearbox torque conditions was developed using the particle swarm optimization (PSO) method. The calculation procedure can be used for both symmetrical and asymmetrical counterweight configurations. It can perform optimizations for different scenarios: minimizing the peak net torque or the cyclic load factor (CLF) values. As proved in this work, use of asymmetrical counterweight placements can significantly reduce the peak net gearbox torque; an often overlooked practice in the oil field.

7.5 Thesis 5

A new calculation procedure was created to improve the crank angle values in the proximity of the start of the upstroke and downstroke. This validation is required if the dynamometer card does not contain the topmost or lowermost point in the dynamometer survey. By using the proposed method, the incorrect calculation of the crank angle in the wrong pumping phase is prevented, therefore, reducing the error in the determination of the crank angular velocity and crank angular acceleration values.

7.6 Thesis 6

A modified cyclic load factor – CLF_{mod} – was developed to describe the relative power consumption of the prime mover with a higher accuracy. This new parameter considers the varying crank angular velocity, therefore it gives improved results when a sucker-rod pumping unit is driven by a high slip, or ultra-high slip electric motor.

8 Summary

In the first part of the thesis the operation of the sucker-rod pumping installation was detailed, followed by the introduction of the measurement by the most dominant testing equipment, the electronic dynamometer.

The evaluation of the dynamometer survey was improved, compared to the previous publications and software used in the petroleum industry. The first important scientific result is the creation of a high-accuracy calculation method to find the crank angles corresponding to the measured polished rod position values. With these more accurate crank angles, the interpretation of the dynamometer survey and the torque analysis will have smaller errors.

The calculation of the angular acceleration of the crank arm and the walking beam was improved, ensuring the accurate description of the inertial torques during the pumping cycle. Every calculation presented is able to consider the varying crank velocity of pumping units driven by high slip or ultra-high slip prime movers. Several previously published methods, basic numerical methods, and novel calculation procedures were introduced and compared, to provide the variation of the necessary variables in time with the highest accuracy possible. The application of Fourier series was essential to improve the calculation of the relevant angles and their acceleration pattern during the pumping cycle.

The complete calculation of the actual net gearbox torque variation was detailed while solving an example problem to help the better understanding of the proposed methods. The proper inclusion of the inertial torques can change the net gearbox torque function significantly, as shown in the comparison with the results of the TWM software.

Most importantly, the in-depth investigation of the effect of asymmetrically placed counterweights on the crank arms was carried out. In previous works application of asymmetrically placed counterweights was not advised, because its effect on the net gearbox torque was unknown. The secondary phase angle was defined to describe the lead- or lag of the center of gravity of the system containing the counterweights and the crank from the crank centerline.

Based on the proposed dynamometer survey interpretation, the determination of the optimum net gearbox torque was carried out using two different optimization criteria. A modified cyclic load factor was introduced to improve the efficiency calculation of the sucker-rod pumping units with varying crank angular velocities. In previous works the cases with non-constant crank angular velocities were not taken properly into account. If the pumping unit is overloaded in the best cyclic load factor case, then a different optimization criteria was used to protect the gearbox: the maximum mechanical net gearbox torque.

A particle swarm optimization technique was developed to find the counterweight configuration that produces the optimum torque loading of the gearbox. Using this method, better torque loading was achieved than the results of previously published methods and software used in the industry by considering the asymmetrical counterweight configurations. Using the secondary phase angle as an additional degree of freedom in the optimization procedure, the results were superior compared to the symmetrical counterbalancing cases.

The knowledge of numerous parameters is required by the complete torque analysis, as seen in the proposed thesis. Some of these variables are usually unknown for the production engineers, or would require extensive and expensive measurements to determine their proper values. Several practical equations are introduced to give a reasonable approximation of these parameters enabling the operators of the sucker-rod pumping unit to carry out an in-depth torque analysis and therefore improve the economic value of the installation.

9 Összefoglalás

Az értekezés első részében a himbás-rudazatos mélyszivattyúk működési mechanizmusa részletesen bemutatásra került, majd a legelterjedtebb mérési módszer – az elektronikus dinamométer – ismertetése következett.

Ezt követően a dinamométeres mérések a korábbi publikációkhoz és az olajiparban használt szoftverekhez képest továbbfejlesztett kiértékelési módszerének bemutatása következett. Az első fontos tudományos eredmény egy nagy pontosságú számítási módszer létrehozása a mért simarúd pozícióknak megfelelő forgattyúszögek meghatározásához. Ezekkel a pontosabb forgattyúszög értékekkel a dinamométeres mérés kiértékelése és a közlőmű nyomatékelemzése kisebb hibákkal terhelt eredményt hoz.

A forgattyúkar és a himbagerenda szöggyorsulásának meghatározási módszerét fejlesztettem, így biztosítva a tehetetlenségi nyomatékok pontos leírását az egész szivattyúzási ciklus alatt. Minden bemutatott számítási lépés figyelembe veszi a nagy szlipű vagy ultra nagy szlipű elektromotorok által hajtott szivattyúegységek változó forgattyúszög-sebességét. Számos korábban publikált számítási módszert, egyszerűsített numerikus megközelítéseket és új számítási eljárásokat vezettem be és hasonlítottam össze, hogy a szükséges változók időbeni változásának leírását a lehető legnagyobb pontossággal biztosítsam. A Fourier sorok alkalmazása elengedhetetlen volt a szögek és gyorsulási mintázatuk kiszámításához a szivattyúzási ciklus során.

A közlőmű eredőnyomaték-változásának teljes kiszámításának módszerét részletesen kidolgoztam, miközben egy példa problémát megoldva segítettem a javasolt módszerek könnyebb megértését. A tehetetlenségi nyomatékok megfelelő beépítése a számítási módszerbe jelentősen megváltoztathatja a közlőmű eredő nyomatékterhelését, amit a TWM szoftver eredményeivel való összehasonlítás is alátámaszt.

Az értekezés legfontosabb számítási része az aszimmetrikusan elhelyezett ellensúlyok hatásának mélyreható vizsgálata volt. Korábbi publikációkban és munkaanyagokban az aszimmetrikusan elhelyezett ellensúlyok alkalmazása nem volt ajánlott, mert annak hatása a közlőmű eredőnyomatékára nem volt ismert. Definiáltam a másodlagos fázisszöget, ami pontosan leírja az ellensúlyokat és a hajtókart tartalmazó rendszer súlypontjának szögeltérését a forgattyúkar középvonalától.

A javasolt dinamométeres mérés értelmezése alapján az optimális eredő közlőműnyomaték meghatározása egy új optimalizálási eljárás segítségével történt, ahol két különböző optimalizálási kritérium került alkalmazásra. Módosított ciklikus terhelési tényezőt vezettem be a himbás-rudazatos szivattyúegységek hatékonysági számításának javítására változó forgattyúszög-sebességek esetére. Korábbi munkákban a nem állandó hajtókar szögsebességű eseteket nem vették megfelelően figyelembe. Ha a szivattyúegység a legjobb ciklikus terhelési tényező esetén túl van terhelve, akkor egy másik optimalizálási kritériumot használtam a közlőmű védelmére: azon eredő közlőműnyomaték-függvény meghatározása, amelyhez a minimális csúcsnyomaték tartozik.

Egyéni részecske raj optimalizálási technikát fejlesztettem ki, hogy megtaláljam azt az ellensúly konfigurációt, amely biztosítja a közlőmű optimális nyomatékterhelését. Ezzel a módszerrel az aszimmetrikus ellensúly-elhelyezések figyelembevételével jobb nyomatékterhelést értem el, mint a korábban publikált módszerek és az iparban használt szoftverek eredményei. A másodlagos fázisszöget további szabadságfokként használva az optimalizálási eljárásban a minimális csúcsnyomatékok meghatározásának eredményei jobbak lettek, mint a szimmetrikusan elhelyezett ellensúlyokat tartalmazó esetek.

A közlőmű teljes nyomatékelemzése számos paraméter ismeretét igényli, amint azt jelen értekezés is alátámaszt. Ezen adatok némelyikét a termelési mérnökök általában nem ismerik, vagy pontos értékeik megállapításához költséges mérések szükségesek. Több egyenletet vezettem be, hogy az ismeretlen paraméterek megfelelően megbecsülhetők lehessenek, lehetővé téve a himbás-rudazatos mélyszivattyús egység kezelői számára, hogy teljes nyomatékelemzést végezhessenek el, ezáltal javítva a berendezés gazdasági értékét.

10 Bibliography

API. (2008). Specification for Pumping Units. *18th Edition*. Washington DC.: American Petroleum Institute.

BakerHughes. (2018, 08). Retrieved 04 21, 2020, from https://www.bhge.com/sites/default/files/2018-08/200171760_MANUAL_INSTALLATION_CONV_RM_EN.pdf

Bommer, P. M., & Podio, A. L. (2012). *The beam lift handbook* (1st ed.). Austin, Texas: University of Texas at Austin.

Clegg, J. (2007). *Petroleum Engineering Handbook* (Vol. IV). Society of Petroleum Engineers.

Danel. (2015, 04). *Welloperation*. Retrieved 4 21, 2020, from http://welloperation.blogspot.com/2015/04/komponen-srp-sucker-rod-pump-pompa.html

Eberhart, R. C., & Kennedy, J. (2001). *Swarm Intelligence.* Morgan Kaufmann.

Echometer. (2007). Well Analyzer and TWM Software Operating Manual Rev. C. Wichita Falls, Texas: Echometer Company.

Echometer. (2011). Retrieved 04 21, 2020, from http://www.echometer.com/Products/Transducers

Engelbrecht, A. (2007). *Computational Intelligence.* John Wiley & Sons, Ltd.

Feng, Z., Ding, H., & Jiang, M. (2015). New secondary balancing method saves energy for crank-balanced rod-pumping application. *SPE Production & Operations, 30*(02), 141-145. doi:https://doi.org/10.2118/173889-PA

Fernández-Martínez, J. (2012, June). A Brief Historical Review of Particle Swarm Optimization (PSO). *Journal of Bioinformatics and Intelligent Control*, 3-16.

Gibbs, S. G. (1975, September). Computing Gearbox Torque and Motor Loading for Beam Pumping Units with Consideration of Inertia Effects. *Journal of Petroleum Technology*, 1153-1159.

Gibbs, S. G. (2012). *Rod Pumping. Modern Methods of Design, Diagnosis and Surveillance.* USA: BookMasters Inc.

Kemler, E. M. (1943). Counterbalancing Of Oil-Well Pumping Machines. *API Drilling and Production Practice*, 87-107.

Konz, Á. (2018). *Sucker Rod Pumping Analysis Based on Measured Electrical Parameters, PhD Thesis.*

Lea, F. (2007). Artificial Lift Selection, Chapter 10. In *Petroleum Engineering Handbook* (Vol. IV.). Dallas, Texas: Society of Petroleum Engineers.

Lufkin. (1997). Conventional Counterbalance Torque (CBT) Data.

Microsoft. (2019). Microsoft Office Support. Retrieved 4 21, 2020, from https://support.microsoft.com/en-us/office/correl-function-995dcef7-0c0a-4bed-a3fb-239d7b68ca92

Pidenergy. (2016, 06). Retrieved 04 21, 2020, from https://pidenergy.com/wp-content/uploads/2016/06/IMG_7402-1024x768.jpg

Production Technology 1. (2018, 06 06). Retrieved 04 21, 2020, from https://production-technology.org/types-of-rod-pumping-units/

Production Technology 2. (2018, 09 03). Retrieved 04 21, 2020, from https://production-technology.org/the-difference-between-conventional-mark-ii-pumping-units/

Richards, C. D. (1957). Counterbalancing Beam-Type Pumping Units. *4th West Texas Oil Lifting Short Course*, (pp. 173-176). Lubbock, Texas.

Rowlan, O., McCoy, J., & Podio, A. (2005, July). Best Method to Balance Torque Loadings on a Pumping Unit Gearbox. *Journal of Canadian Petroleum Technology, 44*(07), 27-33. doi:https://doi.org/10.2118/05-07-TN3

Schlumberger. (2019). Sucker Rod Pump Surface Units Catalog. Schlumberger.

Serway, R. A. (1986). *Physics for Scientists and Engineers* (2nd ed.). Saunders College Publishing.

SPE. (n.d.). *PetroWiki*. Retrieved from PetroWiki: https://petrowiki.org/Artificial_lift, Accessed: 2020. 06. 20

Svinos, J. G. (1983). Exact kinematic analysis of pumping units. *58th Annual Technical Conference and Exhibition of the SPE. SPE 012201-MS.* San Francisco, California: Society of Petroleum Engineers.

Takács, G. (1990). Új módszer mélyszivattyús himbaegységek nyomatékviszonyainak és optimális kiegyensúlyozásának meghatározására. *Kőolaj és Földgáz, 23*(1), 8-13.

Takács, G. (2003). *Sucker-Rod Pumping Manual.* Tulsa, Oklahoma: PennWell Books.

Takács, G. (2015). *Sucker-Rod Pumping Handbook.* Elsevier.

Takács, G., & Kis, L. (2014). Finding the best way to calculate articulating torque from dynamometer survey. *Kőolaj és Földgáz*, 17-20.

Takács, G., Kis, L., & Koncz, Á. (2015). The use of Dynamometer Data for Calculating the Torsional Load on Sucker-Rod Pumping Gearboxes. (pp. 176-183). Lubbock, Texas: Proceedings of the 62nd Southwestern Petroleum Short Course.

Takács, G., Kis, L., & Koncz, Á. (2016). The calculation of gearbox torque components on sucker-rod pumping units using dynamometer card data. *Journal of Petroleum Exploration and Production Technology*, 101-110. doi:https://doi.org/10.1007/s13202-015-0172-z

11 Relevant Publications by Date

Kis, L. (2013): Calculation of the gearbox torque including inertia effects. Doktoranduszok Fóruma, Conference Proceeding. Miskolc. 46-50

Kis, L. (2014): Comparison of beam acceleration calculation models. XXVIII. MicroCad International Multidisciplinary Scientific Conference, Conference Proceeding. Miskolc

Takács, G., & **Kis, L**. (2014): Finding the best way to calculate articulating torque on sucker-rod pumping gear reducers. *Kőolaj és földgáz*, 2014/3. 17-20.

Kis, L. (2014): A dinamométeres mérések kiértékelésének nehézségei (in Hungarian). Doktoranduszok Fóruma, Conference Proceeding. Miskolc

Takács, G., & **Kis, L.,** & Koncz, Á. (2015): The use of Dynamometer Data for Calculating the Torsional Load on Sucker-Rod Pumping Gearboxes, Southwestern Petroleum Short Course, Texas, 22-23 April, 2015. 176-183.

Kis, L. (2015): Mechanical Net Torque Optimization of Sucker-Rod Pumping Units, XXIX. MicroCad International Multidisciplinary Scientific Conference, Conference Proceeding. Miskolc

Kis, L. (2015): The effect of the articulating inertial torque on the permissible loads of sucker-rod pumping units, *Műszaki Földtudományi Közlemények*, 85(1), 118-122.

Takács, G., & **Kis, L.**, & Koncz, Á. (2016): The calculation of gearbox torque components on sucker-rod pumping units using dynamometer card data. *Journal of Petroleum Exploration and Production Technology*, 101-110. doi: https://doi.org/10.1007/s13202-015-0172-z

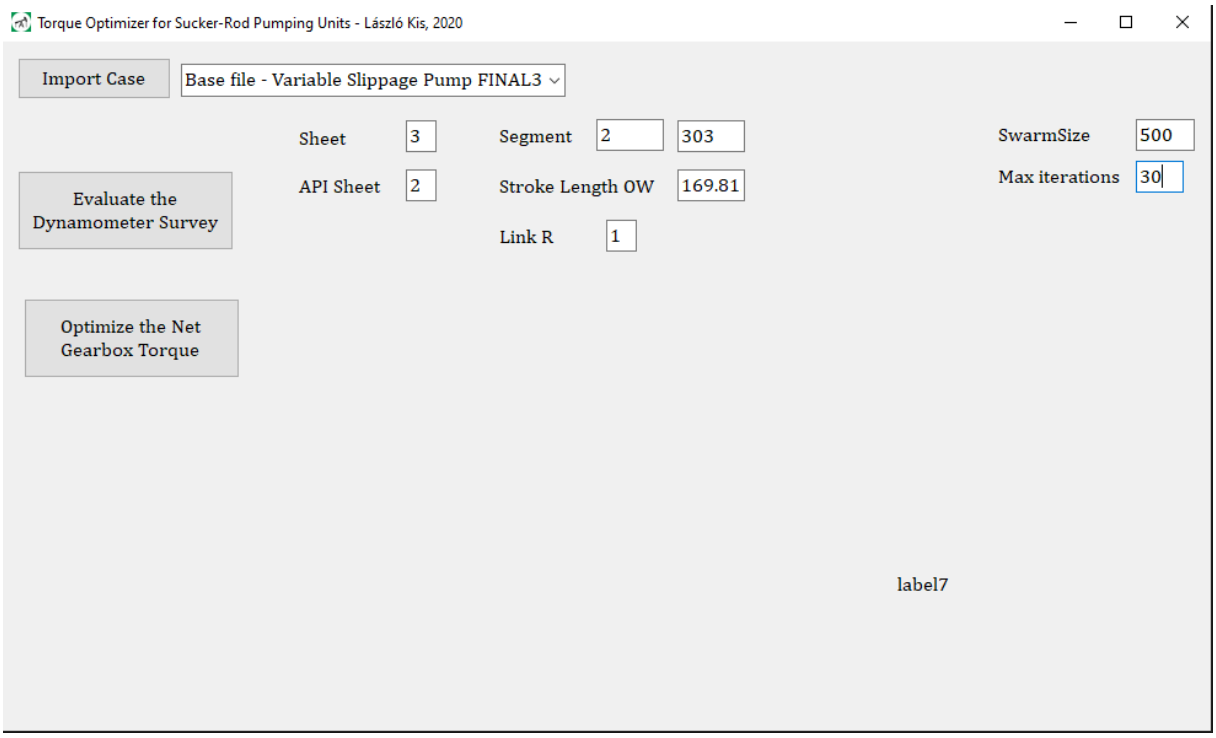
Koncz, Á., & **Kis, L.**, & Szabó, T. (2018): New method for stripper well supervision. 18th International Multidisciplinary Scientific Geoconference SGEM 2018, Conference Proceedings. Sofia, Bulgaria. STEF92 Technology Ltd. 649-656. doi: https://doi.org/10.5593/sgem2018/1.4/S06.085

12 List of Figures

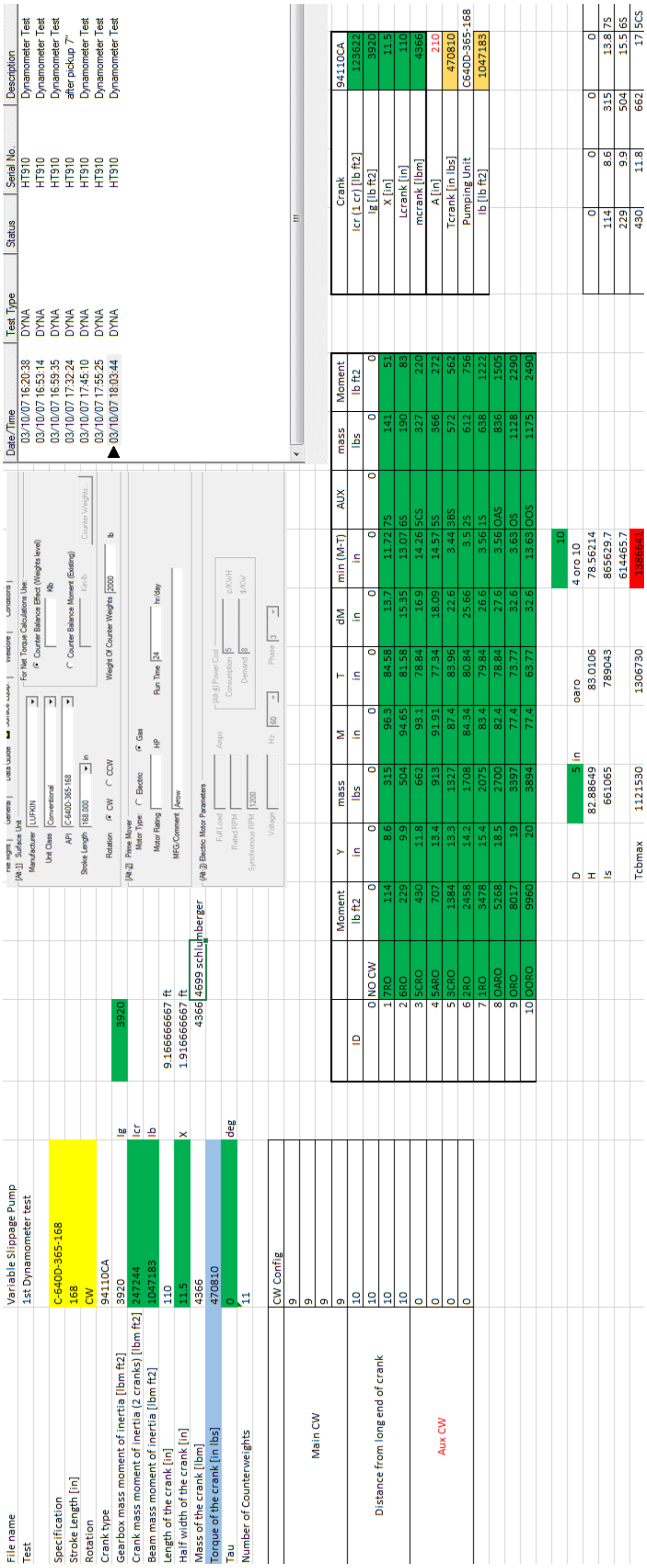
13 List of Tables

14 Appendices

14.1 Appendix A The Developed Program and Parts of its Input and Output Files



Input Excel File



File name	Variable Slippage Pump		
Test	1st Dynamometer test		
Specification	C-640D-365-168		
Stroke Length [in]	168		
Rotation	CW		
Crank type	94110CA		
Gearbox mass moment of inertia [lbm ft2]	3920	Ig	3920
Crank mass moment of inertia (2 cranks) [lbm ft2]	247244	Icr	
Beam mass moment of inertia [lbm ft2]	1047183	Ib	
Length of the crank [in]	110		9.166666667 ft
Half width of the crank [in]	11.5	X	1.916666667 ft
Mass of the crank [lbm]	4366		4366 4699 schlumberger
Torque of the crank [in lbs]	470810		
Tau	0	deg	
Number of Counterweights	11		

	CW Config
Main CW	9
	9
	9
	9
Distance from long end of crank	10
	10
	10
	10
Aux CW	0
	0
	0
	0

Date/Time	Test Type	Status	Serial No.	Description
03/10/07 16:20:38	DYNA		HT910	Dynamometer Test
03/10/07 16:53:14	DYNA		HT910	Dynamometer Test
03/10/07 16:59:35	DYNA		HT910	Dynamometer Test
03/10/07 17:32:24	DYNA		HT910	after pickup 7"
03/10/07 17:45:10	DYNA		HT910	Dynamometer Test
03/10/07 17:55:25	DYNA		HT910	Dynamometer Test
03/10/07 18:03:44	DYNA		HT910	Dynamometer Test

ID		Moment lb ft2	Y in	mass lbs	M in	T in	dM in	min (M-T) in	AUX	mass lbs	Moment lb ft2
0	NO CW	0	0	0	0	0	0	0	0	0	0
1	7RO	114	8.6	315	96.3	84.58	13.7	11.72	7S	141	51
2	6RO	229	9.9	504	94.65	81.58	15.35	13.07	6S	190	83
3	5CRO	430	11.8	662	93.1	78.84	16.9	14.26	5CS	327	220
4	5ARO	707	13.4	913	91.91	77.34	18.09	14.57	5S	366	272
5	3CRO	1384	13.3	1327	87.4	83.96	22.6	3.44	3BS	572	562
6	2RO	2458	14.2	1708	84.34	80.84	25.66	3.5	2S	612	756
7	1RO	3478	15.4	2075	83.4	79.84	26.6	3.56	1S	638	1222
8	OARO	5268	18.5	2700	82.4	78.84	27.6	3.56	OAS	836	1505
9	ORO	8017	19	3397	77.4	73.77	32.6	3.63	OS	1128	2290
10	OORO	9960	20	3894	77.4	63.77	32.6	13.63	OOS	1175	2490

				10
D	5	in	oaro	4 oro 10
H	82.88649		83.0106	78.56214
Is	661065		789043	865629.7
				614465.7
Tcbmax	1121530		1306730	1386641

Crank	94110CA
Icr (1 cr) [lb ft2]	123622
Ig [lb ft2]	3920
X [in]	11.5
Lcrank [in]	110
mcrank [lbm]	4366
A [in]	210
Tcrank [in lbs]	470810
Pumping Unit	C640D-365-168
Ib [lb ft2]	1047183

0	0	0	0	
114	8.6	315	13.8	7S
229	9.9	504	15.5	6S
430	11.8	662	17	5CS

	Time [s]	PR Load [K lbs]	PR Acceleration [G]	PR Position Modified [in]	PR Velocity [in/s]	Filtered Acceleration [?]	Polished Rod Velocity [in/s]	Polished Rod Acceleration [in/s2]	Filtered Polished Rod Acceleration [in/s^2]	Balanced Net Gearbox Torque [K in lbs]	Existing Net Gearbox Torque [K in lbs]	Rod Torque [K in lbs]	Counter-balance Torque [K in lbs]	Instant-aneous SPM	Torque Factor [in]	Crank Angle [deg]
0	0.0000	15.9775	0.0000	0.0200	2.4896	0.0000	2.4896	74.3343	31.3044	-67.9083	-5.5422	0.0000	5.5422	-4.7397	0.0000	2.4769
1	0.0333	16.4863	0.0000	0.1678	5.8719	0.0000	5.8719	74.3961	31.5694	-60.6219	25.5726	33.2324	7.6598	10.5168	1.9014	3.4242
2	0.0667	16.9942	0.0000	0.4157	8.4774	0.0000	8.4774	88.7593	31.6735	-40.7098	98.2975	110.6505	12.3530	8.9334	6.1519	5.5276
3	0.1000	17.4000	0.0000	0.7390	11.0198	0.0000	11.0198	66.8966	31.6531	-19.0007	164.7239	181.0507	16.3269	8.1589	9.7896	7.3143
4	0.1333	17.7957	0.0000	1.1583	13.7198	0.0000	13.7198	84.9907	31.4085	3.7430	228.1530	248.0954	19.9424	8.0590	13.1267	8.9460
5	0.1667	18.0800	0.0000	1.6636	16.5511	0.0000	16.5511	76.3099	31.0687	29.1308	293.5491	317.0469	23.4978	8.0378	16.4310	10.5578
6	0.2000	18.2137	0.0000	2.2736	19.5343	0.0000	19.5343	92.8327	30.6666	55.1250	359.2383	386.2636	27.0253	8.1611	19.7274	12.1654
7	0.2333	18.2685	0.0000	2.9799	22.0949	0.0000	22.0949	85.3834	30.1225	80.0151	424.1873	454.7724	30.5852	8.1636	23.0689	13.7976
8	0.2667	18.2042	0.0000	3.7625	24.5697	0.0000	24.5697	67.5906	29.4059	103.7656	487.7292	521.8505	34.1213	8.0363	26.3981	15.4303
9	0.3000	18.1394	0.0000	4.6356	26.9225	0.0000	26.9225	80.2535	28.4276	123.9623	546.7926	584.3678	37.5752	7.9696	29.6570	17.0376
10	0.3333	18.0434	0.0000	5.5767	28.8647	0.0000	28.8647	60.3053	27.3523	143.3364	604.3823	645.3535	40.9713	7.8428	32.8602	18.6315
11	0.3667	17.8517	0.0000	6.5807	30.8028	0.0000	30.8028	55.7226	26.1640	160.6062	658.9115	703.1938	44.2824	7.6671	35.9812	20.2001
12	0.4000	17.5991	0.0000	7.6525	32.9886	0.0000	32.9886	60.0630	24.9397	172.7289	707.0982	754.5854	47.4872	7.5883	38.9933	21.7335
13	0.4333	17.3085	0.0000	8.8037	35.1774	0.0000	35.1774	70.5210	23.6200	180.5365	750.2224	800.8480	50.6257	7.6038	41.9311	23.2511
14	0.4667	16.9910	0.0000	10.0230	37.1578	0.0000	37.1578	60.2374	22.1943	184.6889	789.3625	843.0974	53.7349	7.5513	44.8252	24.7719
15	0.5000	16.6786	0.0000	11.3077	38.7636	0.0000	38.7636	58.0730	20.6324	185.1874	824.1858	880.9709	56.7852	7.5112	47.6431	26.2822
16	0.5333	16.4333	0.0000	12.6351	40.0727	0.0000	40.0727	37.8598	19.0241	183.4628	856.1634	915.9435	59.7802	7.3537	50.3859	27.7844
17	0.5667	16.2284	0.0000	14.0081	41.0254	0.0000	41.0254	40.3441	17.2650	182.6450	887.8932	950.5657	62.6725	7.2462	53.0056	29.2551
18	0.6000	15.9649	0.0000	15.3997	41.6226	0.0000	41.6226	16.5751	15.6111	181.9230	918.7884	984.2707	65.4822	7.0380	55.5194	30.7044
19	0.6333	15.9902	0.0000	16.8129	42.7195	0.0000	42.7195	19.1012	13.8684	175.8186	942.9418	1011.1130	68.1711	6.8493	57.8940	32.1120
20	0.6667	16.1834	0.0000	18.2785	44.1827	0.0000	44.1827	46.4274	12.1888	184.8268	980.9523	1051.7007	70.7484	6.8657	60.1307	33.4818
21	0.7000	16.3505	0.0000	19.7903	45.3876	0.0000	45.3876	40.9852	10.4546	203.7433	1028.4836	1101.7750	73.2913	6.8333	62.3055	34.8550
22	0.7333	16.5899	0.0000	21.3370	45.7661	0.0000	45.7661	30.9981	8.6727	220.8787	1073.6286	1149.4090	75.7804	6.7722	64.3907	36.2216
23	0.7667	16.9509	0.0000	22.8743	45.5515	0.0000	45.5515	-8.3856	6.9470	242.5521	1122.5826	1200.7873	78.2047	6.5503	66.3790	37.5761
24	0.8000	17.4970	0.0000	24.4066	45.4443	0.0000	45.4443	-4.4351	5.2160	272.4566	1178.4062	1258.9142	80.5080	6.3517	68.2304	38.8861
25	0.8333	18.0427	0.0000	25.9367	45.7518	0.0000	45.7518	-1.9687	3.6183	315.4694	1246.0999	1328.8013	82.7013	6.1903	69.9478	40.1565
26	0.8667	18.4523	0.0000	27.4897	46.3843	0.0000	46.3843	20.3382	2.0812	359.1874	1313.4317	1398.2315	84.7998	6.1526	71.5476	41.3946
27	0.9000	18.7693	0.0000	29.0624	46.8606	0.0000	46.8606	17.4448	0.6194	393.7555	1371.0281	1457.8744	86.8462	6.1096	73.0681	42.6251
28	0.9333	19.0479	0.0000	30.6475	46.7345	0.0000	46.7345	11.0141	-0.8114	421.6926	1421.3867	1510.2255	88.8387	6.0424	74.5079	43.8470
29	0.9667	19.3549	0.0000	32.2117	45.9071	0.0000	45.9071	-18.5485	-2.1468	446.5535	1467.9754	1558.7450	90.7696	5.8548	75.8591	45.0555
30	1.0000	19.6173	0.0000	33.7411	44.6782	0.0000	44.6782	-30.8819	-3.3633	473.1454	1515.1871	1607.7891	92.6020	5.6397	77.0940	46.2264
31	1.0333	19.8122	0.0000	35.2224	43.5651	0.0000	43.5651	-42.5337	-4.3937	495.9744	1557.4669	1651.7975	94.3305	5.3897	78.2200	47.3544
32	1.0667	20.1783	0.0000	36.6768	42.9270	0.0000	42.9270	-23.9653	-5.3207	513.0619	1592.7586	1688.7068	95.9483	5.2271	79.2367	48.4323
33	1.1000	20.4982	0.0000	38.1152	42.6148	0.0000	42.6148	-14.1525	-6.0133	543.4082	1640.3949	1737.8797	97.4847	5.1126	80.1668	49.4777
34	1.1333	20.7397	0.0000	39.5485	42.4232	0.0000	42.4232	-4.4982	-6.5097	569.8485	1683.3932	1782.3494	98.9562	5.0430	81.0227	50.5002
35	1.1667	20.8395	0.0000	40.9739	41.7467	0.0000	41.7467	-6.9482	-6.9760	589.6323	1719.1623	1819.5390	100.3767	4.9690	81.8149	51.5088
36	1.2000	20.7172	0.0000	42.3617	40.3261	0.0000	40.3261	-33.4691	-7.2769	597.3225	1742.2608	1844.0068	101.7460	4.7969	82.5449	52.5026
37	1.2333	20.5513	0.0000	43.6914	38.7590	0.0000	38.7590	-51.3971	-7.4246	585.9922	1745.4786	1848.5174	103.0388	4.5627	83.2022	53.4620

Output Excel File

Time	Polished Rod Load	Polished Rod Position	Calculated Crank Angle	TWM Crank Angle	Calculated Torque Factor	TWM Troque Factor	Calculated Rod Torque	TWM Rod Torque	Beam Accelerati on Fourier 10 term	Beam Accelerati on Fourier 5 term	Beam Accelerati on Fourier 30 term	Beam Accelerati on Numerical	Beam Angular Accelerati on Svinos 5p	MidpointTime	FourierTime	Theta direct before Fourier	Theta direct	Theta p direct	Theta pp direct	Dtheta Simple Numerical	Dtheta Fourier	Dtheta 5Point Numerical	Dtheta 5Point Fourier	Crank Angular Velocity Fourier	Crank Angular Accelerati on Fourier	Crank Angular Velocity Fourier 5 point
0	15.97752	0.019983	3.544	2.476887	2.14	0	37.402	0	0.405175	0.382762	0.397497	0.429094	0.393	0.016667	0	3.544	125.1971	-682.819	868.1946	1.059775	1.1265	0.850769	0.8392	0.844168	0.02817	0.848249
0.033333	16.4863	0.167767	5.568	3.424212	6.232	1.901436	112.091	33.23239	0.407008	0.37837	0.409092	0.429094	0.396	0.05	0.033322	5.5673	103.398	-649.002	1195.772	0.926228	0.993	0.850769	0.8508	0.844801	0.010413	0.848261
0.066667	16.99416	0.415671	7.337	5.527567	9.834	6.151933	181.872	110.6505	0.407001	0.372598	0.418469	0.323402	0.397	0.083334	0.066643	7.3357	82.87333	-605.001	1489.69	0.844573	0.8855	0.873851	0.8623	0.844838	-0.0083	0.847646
0.1	17.40004	0.739034	8.95	7.314256	13.134	9.789619	248.233	181.0507	0.405093	0.365482	0.42311	0.410875	0.396	0.116667	0.099965	8.9483	63.93767	-552.053	1741.948	0.849809	0.8472	0.841388	0.8577	0.844248	-0.02796	0.846398
0.133333	17.79571	1.158266	10.573	8.946039	16.462	13.12672	317.646	248.0954	0.40124	0.357069	0.421364	0.368909	0.393	0.15	0.133287	10.5708	46.85945	-491.631	1945.961	0.837741	0.8438	0.842828	0.8421	0.843	-0.04851	0.844515
0.166667	18.08001	1.663575	12.173	10.55783	19.742	16.43095	386.549	317.0469	0.395417	0.347413	0.412922	0.448792	0.388	0.183334	0.166608	12.1702	31.85583	-425.397	2096.743	0.854522	0.8461	0.84501	0.8439	0.841067	-0.06992	0.841993
0.2	18.21365	2.2736	13.805	12.16539	23.084	19.72744	455.07	386.2636	0.387614	0.336579	0.398862	0.41277	0.38	0.216667	0.19993	13.8016	19.08858	-355.152	2191.047	0.856093	0.8553	0.858274	0.8516	0.83842	-0.09209	0.83883
0.233333	18.26846	2.979936	15.44	13.79762	26.418	23.06891	522.243	454.7724	0.377843	0.324636	0.381233	0.326754	0.371	0.25	0.233251	15.436	8.657887	-282.769	2227.486	0.837218	0.8467	0.846755	0.8525	0.835038	-0.11495	0.835025
0.266667	18.20421	3.762513	17.039	15.43033	29.66	26.39814	584.427	521.8505	0.366136	0.311665	0.362353	0.387976	0.36	0.283334	0.266573	17.0345	0.602523	-210.134	2206.534	0.837766	0.8375	0.837417	0.8421	0.830898	-0.13839	0.830573
0.3	18.13935	4.635616	18.639	17.03758	32.876	29.65701	645.663	584.3678	0.352544	0.297751	0.344046	0.291537	0.346	0.316667	0.299895	18.634	-5.09921	-139.096	2130.494	0.819964	0.8289	0.830305	0.8339	0.825984	-0.16226	0.825473
0.333333	18.04338	5.576743	20.205	18.6315	35.991	32.86022	703.386	645.3535	0.337139	0.282986	0.327084	0.26938	0.331	0.35	0.333216	20.1995	-8.52851	-71.3914	2003.39	0.803184	0.8116	0.810844	0.8206	0.820285	-0.18644	0.819723
0.366667	17.85166	6.580724	21.739	20.20005	39.005	35.98117	754.812	703.1938	0.320016	0.267465	0.311051	0.290372	0.314	0.383334	0.366538	21.7331	-9.82071	-8.60096	1830.787	0.794307	0.7988	0.797231	0.8041	0.813792	-0.21076	0.813319
0.4	17.59913	7.652457	23.256	21.73347	41.942	38.99331	801.056	754.5854	0.301289	0.251292	0.294629	0.340918	0.296	0.416667	0.39986	23.2496	-9.15961	47.89376	1619.602	0.795878	0.7951	0.794787	0.796	0.806505	-0.23505	0.80626
0.433333	17.30854	8.803736	24.776	23.25113	44.835	41.93113	843.281	800.848	0.281095	0.234571	0.276229	0.291212	0.275	0.45	0.433181	24.7691	-6.77027	96.96111	1377.805	0.790618	0.7933	0.79374	0.7943	0.798426	-0.25914	0.798546
0.466667	16.99105	10.02296	26.286	24.77189	47.653	44.82524	881.154	843.0974	0.259588	0.217409	0.254708	0.280746	0.254	0.483334	0.466503	26.2786	-2.91019	137.7427	1114.13	0.786453	0.7886	0.789508	0.7916	0.789565	-0.28283	0.790176
0.5	16.67857	11.3077	27.788	26.28215	50.394	47.64311	916.091	880.9709	0.236944	0.199918	0.22992	0.183025	0.231	0.516667	0.499824	27.7801	2.139402	169.6704	837.7847	0.769698	0.7782	0.778599	0.7841	0.77994	-0.30593	0.781153
0.533333	16.4333	12.63513	29.258	27.7844	53.015	50.3859	950.734	915.9435	0.213356	0.182209	0.202884	0.195042	0.208	0.55	0.533146	29.2498	8.084093	192.4837	558.0713	0.759203	0.7645	0.764985	0.7719	0.769571	-0.32825	0.771483
0.566667	16.22842	14.00808	30.708	29.25514	55.529	53.00564	984.441	950.5657	0.189029	0.164392	0.175509	0.080127	0.184	0.583334	0.566468	30.6993	14.62595	206.2222	284.0788	0.736187	0.7478	0.748361	0.7568	0.758488	-0.34959	0.761174
0.6	15.96491	15.39972	32.114	30.70438	57.899	55.51938	1011.201	984.2707	0.164189	0.146579	0.14996	0.092338	0.159	0.616667	0.599789	32.1051	21.4725	211.2145	24.39441	0.717861	0.7271	0.725061	0.7369	0.746726	-0.36977	0.750238
0.633333	15.99024	16.81291	33.485	32.11198	60.141	57.89397	1051.88	1011.113	0.139068	0.12888	0.127897	0.224452	0.135	0.65	0.633111	33.4759	28.34709	208.058	-213.235	0.718363	0.7181	0.716814	0.721	0.734326	-0.3886	0.738693
0.666667	16.18343	18.27847	34.857	33.48184	62.311	60.13072	1101.872	1051.701	0.113906	0.111401	0.109856	0.198137	0.11	0.683334	0.666433	34.8474	34.99655	197.5857	-422.128	0.715767	0.7171	0.717643	0.7172	0.721334	-0.40591	0.726559
0.7	16.35054	19.79026	36.224	34.85498	64.398	62.30549	1149.539	1101.775	0.088953	0.094248	0.095008	0.149853	0.086	0.716667	0.699754	36.2139	41.1977	180.8301	-596.861	0.709483	0.7127	0.714458	0.7161	0.707804	-0.42153	0.713865
0.733333	16.58986	21.33702	37.579	36.22163	66.389	64.39073	1200.968	1149.409	0.064456	0.07752	0.081383	-0.04054	0.062	0.75	0.733076	37.5686	46.76434	158.9766	-733.46	0.684854	0.6974	0.69827	0.7065	0.693792	-0.43531	0.700645
0.766667	16.95092	22.87431	38.887	37.57608	68.235	66.37903	1258.999	1200.787	0.040661	0.061312	0.066518	-0.02145	0.039	0.783334	0.766397	38.8764	51.55084	133.3156	-829.477	0.665501	0.6753	0.674533	0.6866	0.679362	-0.44711	0.68694
0.8	17.49704	24.40661	40.158	38.88615	69.953	68.23043	1328.9	1258.914	0.017812	0.045715	0.04829	-0.00951	0.017	0.816667	0.799719	40.1473	55.45437	105.1939	-884.024	0.648745	0.6573	0.655901	0.6654	0.664581	-0.45679	0.672796
0.833333	18.04267	25.93668	41.397	40.15649	71.557	69.94779	1398.415	1328.801	-0.00386	0.030814	0.025666	0.098321	-0.004	0.85	0.833041	41.3861	58.41672	75.95978	-897.769	0.644014	0.6464	0.645386	0.6507	0.64952	-0.46426	0.658268
0.866667	18.45227	27.48969	42.627	41.39456	73.077	71.5476	1458.052	1398.232	-0.02414	0.016685	-0.00089	0.084336	-0.023	0.883334	0.866362	42.6157	60.42307	46.91522	-872.846	0.63932	0.6417	0.641851	0.6437	0.634251	-0.46942	0.643417
0.9	18.76933	29.06239	43.848	42.62507	74.513	73.06811	1510.328	1457.874	-0.04282	0.003399	-0.02951	0.053242	-0.041	0.916667	0.899684	43.8364	61.49962	19.27278	-812.727	0.632514	0.636	0.637139	0.6395	0.618853	-0.47219	0.62831
0.933333	19.0479	30.6475	45.056	43.84699	75.862	74.50792	1558.805	1510.225	-0.05973	-0.00898	-0.05746	-0.08966	-0.058	0.95	0.933006	45.0441	61.71028	-5.8906	-722.041	0.613122	0.623	0.624092	0.6307	0.603404	-0.47252	0.613021
0.966667	19.35492	32.21169	46.227	45.05547	77.101	75.85908	1607.935	1558.745	-0.07471	-0.0204	-0.08194	-0.1493	-0.072	0.983334	0.966327	46.2151	61.15145	-27.665	-606.353	0.591149	0.6024	0.602755	0.6136	0.587984	-0.47036	0.59763
1	19.61734	33.74105	47.356	46.22643	78.228	77.094	1651.967	1607.789	-0.08764	-0.03081	-0.10095	-0.20563	-0.084	1.016667	0.999649	47.3441	59.94625	-45.3327	-471.929	0.564445	0.5781	0.577404	0.5903	0.572676	-0.46569	0.582221
1.033333	19.81217	35.22243	48.434	47.35438	79.245	78.21998	1688.883	1651.797	-0.0984	-0.04018	-0.11382	-0.11585	-0.095	1.05	1.03297	48.4223	58.23774	-58.3901	-325.45	0.54715	0.556	0.554584	0.5662	0.557563	-0.45851	0.566881
1.066667	20.17831	36.67678	49.479	48.43232	80.175	79.23675	1738.058	1688.707	-0.10695	-0.04847	-0.12128	-0.06842	-0.103	1.083334	1.066292	49.4672	56.1818	-66.5551	-173.742	0.535123	0.5413	0.540272	0.5476	0.542724	-0.44884	0.551703
1.1	20.49815	38.11517	50.501	49.47773	81.031	80.16677	1782.532	1737.88	-0.11324	-0.05566	-0.12507	-0.02175	-0.109	1.116667	1.099614	50.4892	53.9401	-69.7667	-23.5218	0.528316	0.5318	0.531414	0.5359	0.528242	-0.43672	0.536781
1.133333	20.73971	39.54848	51.51	50.50024	81.823	81.02269	1819.72	1782.349	-0.11727	-0.06175	-0.12723	-0.03358	-0.113	1.15	1.132935	51.498	51.67256	-68.1783	118.8948	0.519923	0.5242	0.525	0.5283	0.514196	-0.42219	0.522207
1.166667	20.83945	40.97395	52.503	51.50884	82.553	81.81486	1844.189	1819.539	-0.11908	-0.06673	-0.12928	-0.16181	-0.115	1.183334	1.166257	52.4908	49.53083	-62.1401	247.8118	0.50266	0.5115	0.512696	0.519	0.500662	-0.40534	0.508076
1.2	20.71717	42.36167	53.463	52.50264	83.21	82.54486	1848.691	1844.007	-0.11874	-0.0706	-0.13161	-0.24846	-0.116	1.216667	1.199579	53.4509	47.65237	-52.1759	358.3482	0.477527	0.4904	0.490312	0.5018	0.487717	-0.38626	0.49448

Crank Angular Acceleration Fourier 5 point	Pitman Angular Velocity Fourier	Pitman Angular Acceleration Fourier	Beam Angular Velocity Fourier	Beam Angular Acceleration Fourier	Pitman Angular Velocity Fourier 5p	Pitman Angular Acceleration Fourier 5p	Beam Angular Velocity Fourier 5p	Beam Angular Acceleration Fourier 5p	TWM Polished Rod Velocity	TWM Polished Rod Acceleration	TWM Polished Rod Filtered Acceleration	TWM Net Gearbox Torque	TWM Balanced Net Gearbox Torque	TWM Polished Rod Torque	TWM Counterbalance Torque	Parameters	Actual Counterblance Torque	Actual Rotary Inertial Torque	Articulating Inertial Torque Svinos 5p	Actual Net Gearbox Torque	Asymmetrical Optimized Net Gearbox Torque	Parameters	Net Gearbox Torque Double	Parameters	Net Gearbox Torque Symmetrical	Parameters	Net Gearbox Torque CLF	Parameters
0.009919	0.269421	0.109	0.008601	0.392	0.270723	0.104	0.008643	0.393	2.489594	74.33427	31.30444	-5.54222	-67.9083	0	5.542217	ORO	-85.883	2.965	1.563	-43.953	-331.229	ORO	-322.775	OORO	-51.329	OARO	-113.288	ORO
-0.0092	0.273675	0.097	0.025066	0.396	0.274795	0.091	0.025169	0.396	5.871865	74.39606	31.56942	25.57263	-60.6219	33.23239	7.659755	ORO	-134.805	-2.752	4.586	-20.88	-308.828	1RO	-300.894	3CRO	-33.377	OARO	-93.381	ORO
-0.02848	0.276981	0.084	0.039552	0.397	0.277902	0.078	0.039683	0.397	8.477386	88.75932	31.67346	98.29751	-40.7098	110.6505	12.35301	ORO	-177.428	-8.514	7.255	3.185	-285.059	OORO	-277.63	OORO	-13.869	OARO	-72.095	OORO
-0.04787	0.279571	0.071	0.052788	0.397	0.280283	0.064	0.052923	0.396	11.01983	66.8966	31.65315	164.7239	-19.0007	181.0507	16.32686	ORO	-216.146	-14.312	9.665	27.44	-260.836	OARO	-253.904	3CRO	6.177	OARO	-50.372	1RO
-0.06736	0.281723	0.056	0.066065	0.394	0.282229	0.05	0.066184	0.393	13.71983	84.99071	31.40848	228.153	3.743016	248.0954	19.9424	9.8 in	-254.931	-20.137	12.023	54.601	-233.478	15.5 in	-227.051	20.1 in	29.119	8.4 in	-25.698	18.2 in
-0.08692	0.283353	0.041	0.079045	0.389	0.283665	0.036	0.079132	0.388	16.55111	76.30988	31.06873	293.5491	29.13076	317.0469	23.49778	9.8 in	-292.966	-25.986	14.235	81.832	-205.827	25.7 in	-199.911	13.6 in	52.195	8.4 in	-0.869	4.1 in
-0.10656	0.28451	0.025	0.092134	0.382	0.284649	0.02	0.092179	0.38	19.53426	92.83271	30.66658	359.2383	55.12502	386.2636	27.02531	9.8 in	-331.526	-31.857	16.301	107.988	-179.011	35.1 in	-173.618	20.1 in	74.146	8.4 in	22.91	60.9 in
-0.12627	0.285114	0.008	0.105016	0.373	0.28511	0.004	0.105015	0.371	22.09489	85.38336	30.12254	424.1873	80.01507	454.7724	30.58518	9.8 in	-369.888	-37.751	18.214	132.818	-153.285	27.8 in	-148.422	13.6 in	94.787	8.4 in	45.426	11.1 in
-0.14607	0.285108	-0.01	0.117317	0.361	0.284997	-0.013	0.117271	0.36	24.56965	67.5906	29.40592	487.7292	103.7656	521.8505	34.12129	0 x OS	-407.113	-43.668	19.843	153.489	-131.516	2 xOS	-127.183	2 xOOS	111.368	1 xOAS	63.881	2 xOS
-0.16594	0.284511	-0.028	0.12927	0.347	0.284335	-0.03	0.12919	0.346	26.92246	80.25353	28.42755	546.7926	123.9623	584.3678	37.5752	0 x OS	-444.045	-49.609	21.139	173.148	-110.534	1 x1S	-106.74	1 x3BS	126.961	1 xOAS	81.386	1 xOS
-0.18588	0.283289	-0.047	0.140544	0.331	0.283095	-0.047	0.140448	0.331	28.86465	60.30531	27.35227	604.3823	143.3364	645.3535	40.97127	0 x OS	-479.857	-55.572	22.139	190.096	-92.079	2 xOOS	-88.824	2 xOOS	139.942	1 xOAS	96.276	1 xOOS
-0.20589	0.281451	-0.065	0.151107	0.313	0.281287	-0.064	0.151019	0.314	30.80278	55.72258	26.16399	658.9115	160.6062	703.1938	44.28236	0 x OS	-514.589	-61.554	22.76	201.429	-79.067	2 xOAS	-76.353	1 x3BS	147.401	1 xOAS	105.638	2 x1S
-0.22594	0.279008	-0.084	0.161029	0.294	0.278924	-0.081	0.16098	0.296	32.98863	60.063	24.9397	707.0982	172.7289	754.5854	47.48722	0	-548.575	-67.549	23.071	208.003	-70.633	6.32 deg	-68.464	11.45 deg	150.169	0 deg	110.32	4.03 deg
-0.246	0.275963	-0.102	0.170413	0.273	0.276004	-0.098	0.170439	0.275	35.17739	70.52105	23.62002	750.2224	180.5365	800.848	50.62567	1.4151	-582.241	-73.546	22.913	210.407	-66.172	1.4041	-64.551	1.5958	148.793	1.3924	110.891	1.3567
-0.26602	0.272321	-0.12	0.179117	0.25	0.272531	-0.114	0.179256	0.254	37.15778	60.23735	22.19428	789.3625	184.6889	843.0974	53.73489		-615.28	-79.53	22.493	208.837	-65.505		-64.436		143.503		107.559	
-0.28592	0.268096	-0.137	0.187111	0.226	0.268514	-0.13	0.187402	0.231	38.7636	58.07298	20.63238	824.1858	185.1874	880.9709	56.78519		-647.721	-85.479	21.633	204.524	-67.405		-66.886		135.527		101.559	
-0.30562	0.263329	-0.153	0.194225	0.201	0.263983	-0.145	0.194708	0.208	40.07265	37.8598	19.02407	856.1634	183.4628	915.9435	59.78016		-679.039	-91.368	20.492	200.819	-68.567		-68.595		128.269		96.258	
-0.325	0.258046	-0.168	0.200508	0.176	0.258959	-0.16	0.201218	0.184	41.02542	40.34406	17.26498	887.8932	182.645	950.5657	62.67254		-709.494	-97.163	18.987	196.771	-69.933		-70.502		120.757		90.699	
-0.34394	0.252314	-0.182	0.205826	0.151	0.253501	-0.173	0.206794	0.159	41.62261	16.57507	15.61115	918.7884	181.923	984.2707	65.48224		-738.591	-102.825	17.108	186.893	-77.048		-78.148		107.554		79.407	
-0.36228	0.246174	-0.195	0.210251	0.126	0.247638	-0.185	0.211501	0.135	42.71949	19.1012	13.86843	942.9418	175.8186	1011.113	68.17112		-766.535	-108.31	15.088	192.123	-68.97		-70.59		109.584		83.319	
-0.37987	0.239618	-0.205	0.213982	0.1	0.241354	-0.196	0.215532	0.11	44.18268	46.42737	12.18879	980.9523	184.8268	1051.701	70.74843		-794.061	-113.568	12.738	206.981	-51.113		-53.24		121.31		96.94	
-0.39652	0.232691	-0.215	0.217002	0.076	0.234684	-0.206	0.218861	0.086	45.38761	40.9852	10.45462	1028.484	203.7433	1101.775	73.29132		-821.033	-118.544	10.292	220.254	-34.704		-37.325		131.534		109.064	
-0.41202	0.225449	-0.223	0.219284	0.052	0.227676	-0.215	0.22145	0.062	45.76612	30.99805	8.672719	1073.629	220.8787	1149.409	75.78041		-847.307	-123.18	7.649	238.13	-13.575		-16.67		146.466		125.89	
-0.42619	0.218014	-0.229	0.220696	0.029	0.220445	-0.221	0.223158	0.039	45.55151	-8.3856	6.947047	1122.583	242.5521	1200.787	78.20473		-872.221	-127.415	4.945	264.308	15.879		12.334		169.873		151.133	
-0.4388	0.210431	-0.233	0.221334	0.007	0.213031	-0.227	0.22407	0.017	45.44432	-4.43509	5.216003	1178.406	272.4566	1258.914	80.50805		-895.994	-131.186	2.21	303.93	58.808		54.846		206.884		189.934	
-0.44966	0.202741	-0.236	0.221279	-0.012	0.205471	-0.23	0.224259	-0.004	45.75181	-1.96867	3.618315	1246.1	315.4694	1328.801	82.70134		-918.745	-134.431	-0.532	344.707	102.928		98.582		245.208		230.008	
-0.45854	0.19494	-0.237	0.220669	-0.031	0.197757	-0.233	0.223858	-0.023	46.38426	20.33818	2.081215	1313.432	359.1874	1398.232	84.79981		-940.905	-137.087	-3.123	376.937	138.589		133.89		275.109		261.647	
-0.46526	0.187076	-0.236	0.219545	-0.048	0.189935	-0.233	0.2229	-0.041	46.86063	17.4448	0.619426	1371.028	393.7555	1457.874	86.84624		-962.475	-139.097	-5.677	403.079	168.25		163.235		299.058		287.321	
-0.46964	0.179203	-0.235	0.21794	-0.063	0.182059	-0.233	0.221414	-0.058	46.73451	11.01414	-0.81137	1421.387	421.6926	1510.225	88.83875		-983.385	-140.406	-8.177	426.837	195.597		190.307		320.769		310.736	
-0.47151	0.171429	-0.231	0.215841	-0.076	0.174242	-0.231	0.219382	-0.072	45.9071	-18.5485	-2.14678	1467.975	446.5535	1558.745	90.7696		-1003.24	-140.965	-10.316	453.417	225.757		220.238		345.489		337.106	
-0.47073	0.163823	-0.227	0.213299	-0.087	0.166553	-0.227	0.216854	-0.084	44.6782	-30.8819	-3.36327	1515.187	473.1454	1607.789	92.602		-1021.98	-140.733	-12.212	477.042	252.927		247.228		367.453		360.653	
-0.4672	0.156446	-0.221	0.210372	-0.095	0.15906	-0.222	0.213888	-0.095	43.56507	-42.5337	-4.3937	1557.467	495.9744	1651.797	94.33053		-1039.51	-139.677	-13.99	495.709	275.063		269.239		384.671		379.377	
-0.46084	0.149285	-0.214	0.207179	-0.102	0.151755	-0.217	0.210607	-0.103	42.92702	-23.9653	-5.32069	1592.759	513.0619	1688.707	95.94825		-1056.15	-137.774	-15.346	528.792	311.589		305.692		416.488		412.648	
-0.4516	0.142345	-0.207	0.203805	-0.107	0.144646	-0.21	0.207099	-0.109	42.61483	-14.1525	-6.01332	1640.395	543.4082	1737.88	97.48475		-1072.08	-135.012	-16.414	559.027	345.264		339.344		445.629		443.2	
-0.43949	0.135627	-0.198	0.200326	-0.11	0.13774	-0.202	0.203447	-0.113	42.42323	-4.49824	-6.50966	1683.393	569.8485	1782.349	98.95618		-1087.47	-131.39	-17.182	583.674	373.378		367.485		469.336		468.292	
-0.42453	0.129154	-0.19	0.196795	-0.111	0.131067	-0.194	0.199709	-0.115	41.74668	-6.94818	-6.97597	1719.162	589.6323	1819.539	100.3767		-1102.3	-126.919	-17.642	597.331	390.515		384.7		482.211		482.52	
-0.40681	0.123006	-0.18	0.193234	-0.11	0.124712	-0.185	0.195913	-0.116	40.32611	-33.4691	-7.27695	1742.261	597.3225	1844.007	101.746		-1116.31	-121.623	-17.938	592.818	389.431		383.744		477.092		478.697	

14.2 Appendix B Parts of the Source Code of the Created Program

```
public void IndependentFromCrankAngle()
    {
        if (Geometry == "Conventional" || Geometry == "TorqMaster")
        {
            Phi = Math.Round(Math.Asin(I / K) * 180 / Math.PI, 4);
            PsiBottom = Math.Round(Math.Acos((C * C + K * K - (P + R) * (P + R)) / (2 * C * K)) * 180 / Math.PI,
4);
            PsiTop = Math.Round(Math.Acos((C * C + K * K - (P - R) * (P - R)) / (2 * C * K)) * 180 / Math.PI, 4);
            ThetaUpstroke = Math.Round(Norm(Phi - Math.Asin(C / (P + R) * Math.Sin(PsiBottom / 180 *
Math.PI)) * 180 / Math.PI), 4);
            ThetaDownstroke = Math.Round(Norm(Phi - Math.Asin(C / (P - R) * Math.Sin(PsiTop / 180 *
Math.PI)) * 180 / Math.PI + 180), 4);

            if (Rotation == "CCW" && Geometry == "Conventional")
            {
                ThetaUpstroke = 360 - ThetaUpstroke;
                ThetaDownstroke = 360 - ThetaDownstroke;
            }
        }

        if (Geometry == "Mark II")
        {
            Phi = Math.Round(Math.Asin(I / K) * 180 / Math.PI + 180, 4);
            PsiTop = Math.Round(Math.Acos((C * C + K * K - (P + R) * (P + R)) / (2 * C * K)) * 180 / Math.PI, 4);
            PsiBottom = Math.Round(Math.Acos((C * C + K * K - (P - R) * (P - R)) / (2 * C * K)) * 180 / Math.PI,
4);
            ThetaUpstroke = Math.Round(Norm(Phi - Math.Asin(C / (P - R) * Math.Sin(PsiBottom / 180 *
Math.PI)) * 180 / Math.PI + 180), 4);
            ThetaDownstroke = Math.Round(Phi - Math.Asin(C / (P + R) * Math.Sin(PsiTop / 180 * Math.PI)) *
180 / Math.PI, 4);
        }

        if (Geometry == "Air Balanced")
        {
            Phi = Math.Round(-Math.Asin(I / K) * 180 / Math.PI + 180, 4);
            PsiTop = Math.Round(Math.Acos((C * C + K * K - (P + R) * (P + R)) / (2 * C * K)) * 180 / Math.PI, 4);
            PsiBottom = Math.Round(Math.Acos((C * C + K * K - (P - R) * (P - R)) / (2 * C * K)) * 180 / Math.PI,
4);
            ThetaUpstroke = Math.Round(Norm(Phi + Math.Asin(C / (P - R) * Math.Sin(PsiBottom / 180 *
Math.PI)) * 180 / Math.PI - 180), 4);
            ThetaDownstroke = Math.Round(Norm(Phi + Math.Asin(C / (P + R) * Math.Sin(PsiTop / 180 *
Math.PI)) * 180 / Math.PI), 4);

            if (Rotation == "CCW")
            {
                ThetaUpstroke = 360 - ThetaUpstroke;
                ThetaDownstroke = 360 - ThetaDownstroke;
            }
        }
```

```
        Upstroke = Norm(ThetaDownstroke - ThetaUpstroke);
        Downstroke = 360 - Upstroke;

        StrokeLength = Math.Round(A * Math.Abs((PsiTop - PsiBottom)) / 180 * Math.PI, 4);
    }

public void CrankAngleCalculation()
    {
        double k = ThetaUpstroke;   //Independent Crank Angle moving trough the whole interval
        double diff = 1;   //Auxiliary variable to determine the correct Crank Angle
        double Pos1, Pos2 = 0;
        Epsilon = 0.0001;

        StrokeLength = Convert.ToDouble(textBoxStrokeLengthOverWrite.Text);

        for (int j = 0; j < twm.PolishedRodPosition.Count; j++)
        {
            if (twm.PolishedRodPosition[j] == StrokeLength)
            {
                calc.CrankAngle.Add(Math.Round(ThetaDownstroke, 3));
                k = ThetaDownstroke;
                DependentFromCrankAngle(k, true);
                calc.Theta2.Add(Theta2);
                calc.Theta3.Add(Theta3);
                calc.Theta4.Add(Theta4);
                calc.TorqueFactor.Add(0);
            }
            if (twm.PolishedRodPosition[j] == 0)
            {
                calc.CrankAngle.Add(Math.Round(ThetaUpstroke, 3));
                k = ThetaUpstroke;
                DependentFromCrankAngle(k, true);
                calc.Theta2.Add(Theta2);
                calc.Theta3.Add(Theta3);
                calc.Theta4.Add(Theta4);
                calc.TorqueFactor.Add(0);
            }
            if (twm.PolishedRodPosition[j] != 0 && twm.PolishedRodPosition[j] != StrokeLength)
            {
                while (diff > 0)
                {
                    k = Norm(k + Epsilon);

                    DependentFromCrankAngle(k, true);
                    Pos1 = (PsiBottom - Psi) / (PsiBottom - PsiTop);
                    DependentFromCrankAngle(k + Epsilon, true);
                    Pos2 = (PsiBottom - Psi) / (PsiBottom - PsiTop);
                    diff    =    (Pos1    -    twm.PolishedRodPosition[j]    /    StrokeLength)    *    (Pos2    -
twm.PolishedRodPosition[j] / StrokeLength);
```

```
            }

            calc.CrankAngle.Add(Norm(Math.Round(k + Epsilon / 2, 3)));
            calc.TorqueFactor.Add(Math.Round(TF, 3));
            calc.Theta2.Add(Theta2);
            calc.Theta3.Add(Theta3);
            calc.Theta4.Add(Theta4);



            diff = 1;
          }

          calc.Beta.Add(Beta);
          calc.J.Add(J);
          calc.Rho.Add(Rho);
          calc.Ksi.Add(Ksi);
          calc.Psi.Add(Psi);
          calc.Alpha.Add(Alpha);
        }
      }

public void DependentFromCrankAngle(double Angle, bool samestart)
      {
        int b = 0;
        if (Geometry == "Conventional" || Geometry == "TorqMaster")
        {
          // Rotationation assign
          if (Rotation == "CCW")
          {
            Theta = 360 - Angle;
          }
          else
          {
            Theta = Angle;
          }

          Theta2 = Norm(360 - Theta + Phi);

          if (Theta2 < 180 & Theta2 >= 0)
          { b = -1; }
          else { b = 1; }
          Beta = Math.Acos((C * C + P * P - R * R - K * K + 2 * K * R * Math.Cos(Theta2 / 180 * Math.PI)) / (2 *
C * P)) * 180 / Math.PI;
          J = Math.Sqrt(K * K + R * R - 2 * R * K * Math.Cos(Theta2 / 180 * Math.PI));
          Rho = Math.Acos((J * J + K * K - R * R) / (2 * J * K)) * 180 / Math.PI * b;
          if ((J * J + K * K - R * R) / (2 * J * K) > 1)
          { Rho = 0; }
          if ((J * J + K * K - R * R) / (2 * J * K) < -1)
          { Rho = 0; }
```

```
        Ksi = Math.Acos((C * C + J * J - P * P) / (2 * C * J)) * 180 / Math.PI;
        Psi = Ksi - Rho;
        Theta3 = Math.Acos((P * P + J * J - C * C) / (2 * P * J)) * 180 / Math.PI + Rho;
        Theta4 = 180 - Psi;
        Alpha = Beta + Psi - (Theta - Phi);
        TF = R * A / C * Math.Sin(Alpha / 180 * Math.PI) / Math.Sin(Beta / 180 * Math.PI);
      }

      if (Geometry == "Mark II")
      {
        Theta = Angle;

        Theta2 = Phi - Theta;

        Beta = Math.Acos((C * C + P * P - R * R - K * K + 2 * K * R * Math.Cos(Theta2 / 180 * Math.PI)) / (2 *
C * P)) * 180 / Math.PI;
        J = Math.Sqrt(K * K + R * R - 2 * R * K * Math.Cos(Theta2 / 180 * Math.PI));
        Rho = Math.Asin((R / J * Math.Sin(Theta2 / 180 * Math.PI))) * 180 / Math.PI;
        Ksi = Math.Asin((P / J * Math.Sin(Beta / 180 * Math.PI))) * 180 / Math.PI;
        Psi = Ksi + Rho;
        Theta3 = Math.Acos((P * P + J * J - C * C) / (2 * P * J)) * 180 / Math.PI - Rho;
        Theta4 = 180 - Psi;
        Alpha = -(Beta + Psi - (Theta - Phi));
        TF = R * A / C * Math.Sin(Alpha / 180 * Math.PI) / Math.Sin(Beta / 180 * Math.PI);
      }

      if (Geometry == "Air Balanced")
      {
        // Rotationation assign
        if (Rotation == "CCW")
        {
          Theta = 360 - Angle;
        }
        else
        {
          Theta = Angle;
        }

        Theta2 = Theta - Phi;

        Beta = Math.Acos((C * C + P * P - R * R - K * K + 2 * K * R * Math.Cos(Theta2 / 180 * Math.PI)) / (2 *
C * P)) * 180 / Math.PI;

        if (Theta2 > 360)
        {
          Theta2 = Theta2 - 360;
        }
        J = Math.Sqrt(K * K + R * R - 2 * R * K * Math.Cos(Theta2 / 180 * Math.PI));
        Rho = Math.Asin((R / J * Math.Sin(Theta2 / 180 * Math.PI))) * 180 / Math.PI;
        Ksi = Math.Asin((P / J * Math.Sin(Beta / 180 * Math.PI))) * 180 / Math.PI;
```

```
            Psi = Ksi + Rho;
            Theta3 = Math.Acos((P * P + J * J - C * C) / (2 * P * J)) * 180 / Math.PI - Rho;
            Theta4 = 180 - Psi;
            Alpha = Beta + Psi + (Theta - Phi);
            TF = R * A / C * Math.Sin(Alpha / 180 * Math.PI) / Math.Sin(Beta / 180 * Math.PI);
        }
    }

private void SvinosCalculation()
    {
        for (int i = 0; i < twm.Time.Count - 1; i++)
        {
            calc.BeamVelocityNumerical.Add((twm.PolishedRodPosition[i + 1] - twm.PolishedRodPosition[i])
/ twm.Time[1]);
        }

        calc.BeamAccelerationNumerical.Add(0);
        for (int i = 1; i < twm.Time.Count - 1; i++)
        {
            calc.BeamAccelerationNumerical.Add((calc.BeamVelocityNumerical[i] -
calc.BeamVelocityNumerical[i - 1]) / twm.Time[1] / A);
        }
        calc.BeamAccelerationNumerical[0] = calc.BeamAccelerationNumerical[1];

calc.BeamAccelerationNumerical.Add(calc.BeamAccelerationNumerical[calc.BeamAccelerationNumerical.
Count - 1]);

        FourierPreparation();

        FourierPrepTheta2(calc.Theta2, twm.Time, calc.Theta2f, calc.TimeFourier);
        FourierPrep(calc.Theta3, twm.Time, calc.Theta3f, calc.TimeFourier);
        FourierPrep(calc.Theta4, twm.Time, calc.Theta4f, calc.TimeFourier);

        FourierPrep(twm.PolishedRodLoad, twm.Time, calc.PolishedRodLoadF, calc.TimeFourier);

        FourierPrep(twm.PolishedRodPosition, twm.Time, calc.PolishedRodPositionF10term,
calc.TimeFourier);

        FourierPrep(calc.CrankAngle, twm.Time, calc.Theta2fDirectDummy, calc.TimeFourier);



        Fourier(calc.DTheta5PointFourier, 10, calc.CrankAngularVelocity5Point,
calc.CrankAngularAcceleration5Point, calc.CrankAngularAccelerationChange5Point);
        Fourier(calc.DThetaFourier, 10, calc.Theta2p, calc.Theta2pp,
calc.CrankAngularAccelerationChange);

        Fourier(calc.Theta2fDirectDummy, 10, calc.Theta2fDirect, calc.Theta2pfDirect,
calc.Theta2ppfDirect);

        Fourier(calc.PolishedRodLoadF, 200, calc.PolishedRodLoadFourier, calc.Dummy, calc.Dummy);
```

```
        Fourier(calc.PolishedRodPositionF10term, 10, calc.PolishedRodPositionFourier10term, calc.PolishedRodPositionpFourier10term, calc.PolishedRodPositionppFourier10term);
        Fourier(calc.PolishedRodPositionF10term, 5, calc.PolishedRodPositionFourier5term, calc.PolishedRodPositionpFourier5term, calc.PolishedRodPositionppFourier5term);
        Fourier(calc.PolishedRodPositionF10term, 30, calc.PolishedRodPositionFourier20term, calc.PolishedRodPositionpFourier20term, calc.PolishedRodPositionppFourier20term);

        for (int i = 0; i < twm.Time.Count; i++)
        {
          calc.BeamAccelerationFourier10term.Add(calc.PolishedRodPositionppFourier10term[i] / A);
          calc.BeamAccelerationFourier5term.Add(calc.PolishedRodPositionppFourier5term[i] / A);
          calc.BeamAccelerationFourier20term.Add(calc.PolishedRodPositionppFourier20term[i] / A);
        }



        for (int i = 0; i < calc.Theta2p.Count; i++)
        {
          if (Rotation == "CW")
          {
            calc.Theta3p.Add(Math.Round(-R / P * calc.Theta2p[i] * Math.Sin((calc.Theta4f[i] - calc.Theta2f[i]) / 180 * Math.PI) / Math.Sin((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI), 10));
            calc.Theta4p.Add(Math.Round(-R / C * calc.Theta2p[i] * Math.Sin((calc.Theta3f[i] - calc.Theta2f[i]) / 180 * Math.PI) / Math.Sin((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI), 10));

            calc.Theta3pp.Add(Math.Round(calc.Theta3p[i] * (calc.Theta2pp[i] / calc.Theta2p[i] - (calc.Theta3p[i] - calc.Theta4p[i]) / (Math.Tan((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)) + (calc.Theta4p[i] + calc.Theta2p[i]) / (Math.Tan((calc.Theta4f[i] - calc.Theta2f[i]) / 180 * Math.PI))), 3));

            calc.Theta4pp.Add(Math.Round(calc.Theta4p[i] * (calc.Theta2pp[i] / calc.Theta2p[i] - (calc.Theta3p[i] - calc.Theta4p[i]) / (Math.Tan((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)) - (calc.Theta3p[i] + calc.Theta2p[i]) / (Math.Tan((calc.Theta2f[i] - calc.Theta3f[i]) / 180 * Math.PI))), 3));

            calc.Theta3p5p.Add(Math.Round(-R / P * calc.CrankAngularVelocity5Point[i] * Math.Sin((calc.Theta4f[i] - calc.Theta2f[i]) / 180 * Math.PI) / Math.Sin((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI), 10));
            calc.Theta4p5p.Add(Math.Round(-R / C * calc.CrankAngularVelocity5Point[i] * Math.Sin((calc.Theta3f[i] - calc.Theta2f[i]) / 180 * Math.PI) / Math.Sin((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI), 10));

            calc.Theta3pp5p.Add(Math.Round(calc.Theta3p[i] * (calc.CrankAngularAcceleration5Point[i] / calc.CrankAngularVelocity5Point[i] - (calc.Theta3p[i] - calc.Theta4p[i]) / (Math.Tan((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)) + (calc.Theta4p[i] + calc.CrankAngularVelocity5Point[i]) / (Math.Tan((calc.Theta4f[i] - calc.Theta2f[i]) / 180 * Math.PI))), 3));

            calc.Theta4pp5p.Add(Math.Round(calc.Theta4p[i] * (calc.CrankAngularAcceleration5Point[i] / calc.CrankAngularVelocity5Point[i] - (calc.Theta3p[i] - calc.Theta4p[i]) / (Math.Tan((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)) - (calc.Theta3p[i] + calc.CrankAngularVelocity5Point[i]) / (Math.Tan((calc.Theta2f[i] - calc.Theta3f[i]) / 180 * Math.PI))), 3));
```

```
            }

            else
            {
              calc.Theta3p.Add(Math.Round((-R / P * calc.Theta2p[i] * Math.Sin((calc.Theta4f[i] - calc.Theta2f[i]) / 180 * Math.PI) / Math.Sin((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)), 10));
              calc.Theta4p.Add(Math.Round((-R / C * calc.Theta2p[i] * Math.Sin((calc.Theta3f[i] - calc.Theta2f[i]) / 180 * Math.PI) / Math.Sin((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)), 10));

              calc.Theta3pp.Add(Math.Round((calc.Theta3p[i] * (calc.Theta2pp[i] / calc.Theta2p[i] - (calc.Theta3p[i] - calc.Theta4p[i]) / (Math.Tan((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)) + (calc.Theta4p[i] + calc.Theta2p[i]) / (Math.Tan((calc.Theta4f[i] - calc.Theta2f[i]) / 180 * Math.PI)))), 3));

              calc.Theta4pp.Add(Math.Round((calc.Theta4p[i] * (calc.Theta2pp[i] / calc.Theta2p[i] - (calc.Theta3p[i] - calc.Theta4p[i]) / (Math.Tan((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)) - (calc.Theta3p[i] + calc.Theta2p[i]) / (Math.Tan((calc.Theta2f[i] - calc.Theta3f[i]) / 180 * Math.PI)))), 3));

              calc.Theta3p5p.Add(Math.Round((-R / P * calc.CrankAngularVelocity5Point[i] * Math.Sin((calc.Theta4f[i] - calc.Theta2f[i]) / 180 * Math.PI) / Math.Sin((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)), 10));
              calc.Theta4p5p.Add(Math.Round((-R / C * calc.CrankAngularVelocity5Point[i] * Math.Sin((calc.Theta3f[i] - calc.Theta2f[i]) / 180 * Math.PI) / Math.Sin((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)), 10));

              calc.Theta3pp5p.Add(Math.Round((calc.Theta3p[i] * (calc.CrankAngularAcceleration5Point[i] / calc.CrankAngularVelocity5Point[i] - (calc.Theta3p[i] - calc.Theta4p[i]) / (Math.Tan((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)) + (calc.Theta4p[i] + calc.CrankAngularVelocity5Point[i]) / (Math.Tan((calc.Theta4f[i] - calc.Theta2f[i]) / 180 * Math.PI)))), 3));

              calc.Theta4pp5p.Add(Math.Round(-(calc.Theta4p[i] * (calc.CrankAngularAcceleration5Point[i] / calc.CrankAngularVelocity5Point[i] - (calc.Theta3p[i] - calc.Theta4p[i]) / (Math.Tan((calc.Theta3f[i] - calc.Theta4f[i]) / 180 * Math.PI)) - (calc.Theta3p[i] + calc.CrankAngularVelocity5Point[i]) / (Math.Tan((calc.Theta2f[i] - calc.Theta3f[i]) / 180 * Math.PI)))), 3));

            }
          }


        }

public void Fourier(List<double> data, int order, List<double> dataf, List<double> datafd, List<double> datafdd)
        {
          double a0 = 0;
          double four = 0;
          double fourd = 0;
          double fourdd = 0;
          double k = 0;
          double l = 0;
```

```
            List<double> af = new List<double>();
            List<double> bf = new List<double>();
            List<double> fouriertime = new List<double>();

            for (int i = 0; i < data.Count; i++)
            {
                fouriertime.Add(-Math.PI + 2 * Math.PI * i / (data.Count));
            }

            a0 = Sum(data) / data.Count;

            for (int i = 0; i < order; i++)
            {
                for (int j = 0; j < data.Count; j++)
                {
                    k += data[j] * Math.Sin((i + 1) * fouriertime[j]);
                    l += data[j] * Math.Cos((i + 1) * fouriertime[j]);
                }

                af.Add(Math.Round(k / data.Count * 2, 5));
                bf.Add(Math.Round(l / data.Count * 2, 5));
                k = 0;
                l = 0;
            }

            for (int i = 0; i < fouriertime.Count; i++)
            {
                for (int j = 0; j < order; j++)
                {
                    four = four + af[j] * Math.Sin((twm.Time[i] * 2 * Math.PI / calc.PeriodTime - Math.PI) * (j + 1)) +
bf[j] * Math.Cos((twm.Time[i] * 2 * Math.PI / calc.PeriodTime - Math.PI) * (j + 1));
                    fourd = fourd + af[j] * (j + 1) * Math.Cos((twm.Time[i] * 2 * Math.PI / calc.PeriodTime - Math.PI)
* (j + 1)) - bf[j] * (j + 1) * Math.Sin((twm.Time[i] * 2 * Math.PI / calc.PeriodTime - Math.PI) * (j + 1));
                    fourdd = fourdd - af[j] * (j + 1) * (j + 1) * Math.Sin((twm.Time[i] * 2 * Math.PI / calc.PeriodTime -
Math.PI) * (j + 1)) - bf[j] * (j + 1) * (j + 1) * Math.Cos((twm.Time[i] * 2 * Math.PI / calc.PeriodTime -
Math.PI) * (j + 1));

                }

                dataf.Add(four + a0);
                datafd.Add(fourd * 2 / Math.PI);
                datafdd.Add(fourdd * 4 / Math.PI / Math.PI);

                four = 0;
                fourd = 0;
                fourdd = 0;
            }
        }
```

```
public void FourierPrep(List<double> data, List<double> time, List<double> dataf, List<double> timef,
bool mod)
    {
      if (mod)
      {
        dataf.Add(1);

        int k = 0;

        for (int i = 1; i < time.Count; i++)
        {
          if (i + k >= time.Count)
          {
            k--;
          }

          if (timef[i] > time[i + k])
          {
            k++;
          }

          if (i + k >= time.Count)
          {
            k--;
          }

          if (Math.Abs(data[i - 2 + k] - data[i + k - 1]) > 300)
          {
            if (data[i - 2 + k] < data[i + k - 1])
            {
              double change = time[i + k - 3];

              while (dataf[i + k - 3] + (dataf[i + k - 3] - dataf[i - 4 + k]) * (change - timef[i + k - 3]) / (timef[i
+ k - 3] - timef[i - 4 + k]) > 0)
              {
                change += 0.0001;
              }

              change = Math.Round(change, 5);
              dataf[dataf.Count - 1] = Math.Round(dataf[i + k - 3] * (change - timef[i + k - 2]) / (change -
timef[i + k - 3]), 4);
            }

            else
            {
              MessageBox.Show("Theta2 in the upper side of the seesaw curve :(");
            }

          }
```

```
                dataf.Add(Math.Round(data[i - 1 + k] + (data[i + k] - data[i - 1 + k]) * (timef[i] - time[i - 1 + k]) /
(time[i + k] - time[i - 1 + k]), 4));

            }
            dataf[0] = Math.Round(data[0] + (data[0] - data[1]) / (time[1] - time[0]) * time[0], 4);
        }

    }

public void FourierPrepTheta2(List<double> data, List<double> time, List<double> dataf, List<double>
timef)
    {
        dataf.Add(1);

        int k = 0;

        for (int i = 1; i < time.Count; i++)
        {
            if (data[i] < data[i - 1])
            {
                dataf.Add(Math.Round(data[i - 1] + (data[i] - data[i - 1]) * (timef[i] - time[i - 1]) / (time[i] -
time[i - 1]), 4));
            }
            else
            {
                dataf.Add(Norm(Math.Round(data[i] + (360 + data[i - 1] - data[i]) * (time[i] - timef[i]) / (time[i]
- time[i - 1]), 4)));
            }

        }
        dataf[0] = Math.Round(data[0] + (data[0] - data[1]) / (time[1] - time[0]) * time[0], 4);
    }

private void TorqueCalculation()
    {

        for (int j = 0; j < twm.PolishedRodPosition.Count; j++)
        {
            calc.RodTorque.Add(Math.Round(calc.TorqueFactor[j] * (twm.PolishedRodLoad[j] -
StructuralUnbalance / 1000), 3));
            calc.ArticulatingInertialTorqueSvinos5p.Add(Math.Round(12 / 32.2 * calc.TorqueFactor[j] / A *
MassMomentBeam * calc.Theta4pp5p[j] / 1000, 3));
            calc.ArticulatingInertialTorqueGibbs10term.Add(Math.Round(12 / 32.2 * calc.TorqueFactor[j] / A
* MassMomentBeam * calc.BeamAccelerationFourier10term[j] / A / 1000, 3));
            calc.ArticulatingInertialTorqueGibbs5term.Add(Math.Round(12 / 32.2 * calc.TorqueFactor[j] / A *
MassMomentBeam * calc.BeamAccelerationFourier5term[j] / A / 1000, 3));
            calc.ArticulatingInertialTorqueGibbs20term.Add(Math.Round(12 / 32.2 * calc.TorqueFactor[j] / A
* MassMomentBeam * calc.BeamAccelerationFourier20term[j] / A / 1000, 3));
            calc.ArticulatingInertialTorqueNumerical.Add(Math.Round(12 / 32.2 * calc.TorqueFactor[j] / A *
MassMomentBeam * calc.Theta4pp5p[j] / 1000, 3));
```

```
        }

        CounterbalanceTorquecalculation(ActualCounterweightConfiguration);

        for (int j = 0; j < twm.PolishedRodPosition.Count; j++)
        {
            calc.ActualNetGearboxTorque.Add(calc.RodTorque[j] + calc.ActualCounterbalanceTorque[j] +
calc.ActualRotaryInertialTorque[j] + calc.ArticulatingInertialTorqueSvinos5p[j]);
        }

        SW = new Stopwatch();
        SW.Start();

        AsymmetricCBcalculation();

        AsymmetricCBcalculationDouble();

        SymmetricCBcalculation();

        CLFOptimization();

        double ElapsedTime = Convert.ToDouble(SW.ElapsedMilliseconds) / 1000;
        SW.Stop();



    }


private void AsymmetricCBcalculation()
    {
        GlobalBest.BestFitnessValue = double.PositiveInfinity;

        Swarm = new Particle[SwarmSize];

        for (int i = 0; i < SwarmSize; i++)
        {
            Swarm[i].Position = PSO_RandPosition(LowerLimits, UpperLimits, DimensionSize);
            Swarm[i].Speed = PSO_Rand(LowerLimits, UpperLimits2, DimensionSize);
            Swarm[i].FitnessValue = FitnessCalculation(Swarm[i].Position, false, false);
            Swarm[i].BestFitnessValue = Swarm[i].FitnessValue;
            Swarm[i].BestPosition = (double[])Swarm[i].Position.Clone();
            Swarm[i].Tau = TauCalculation(Swarm[i].Position);
            Swarm[i].CLF = CLFCalculation(Swarm[i].Position);


            if (Swarm[i].BestFitnessValue <= GlobalBest.BestFitnessValue)
            {
                GlobalBest.BestPosition = (double[])Swarm[i].Position.Clone();
                GlobalBest.BestPositionText = new string[14] { "", "", "", "", "", "", "", "", "", "", "", "", "", "" };
```

```
            GlobalBest.BestPositionText[0] =
api.CounterweightName[Convert.ToInt16(Math.Round(Swarm[i].Position[0], 0))];
            GlobalBest.BestPositionText[1] =
api.CounterweightName[Convert.ToInt16(Math.Round(Swarm[i].Position[1], 0))];
            GlobalBest.BestPositionText[2] =
api.CounterweightName[Convert.ToInt16(Math.Round(Swarm[i].Position[2], 0))];
            GlobalBest.BestPositionText[3] =
api.CounterweightName[Convert.ToInt16(Math.Round(Swarm[i].Position[3], 0))];

            GlobalBest.BestPositionText[4] = Math.Round(Swarm[i].Position[4],1).ToString() + " in";
            GlobalBest.BestPositionText[5] = Math.Round(Swarm[i].Position[5],1).ToString() + " in";
            GlobalBest.BestPositionText[6] = Math.Round(Swarm[i].Position[6],1).ToString() + " in";
            GlobalBest.BestPositionText[7] = Math.Round(Swarm[i].Position[7],1).ToString() + " in";

            GlobalBest.BestPositionText[8] = Convert.ToInt16(Math.Round(Swarm[i].Position[8],
0)).ToString() + " x" + api.AuxCounterweightName[Convert.ToInt16(Math.Round(Swarm[i].Position[0],
0))];
            GlobalBest.BestPositionText[9] = Convert.ToInt16(Math.Round(Swarm[i].Position[9],
0)).ToString() + " x" + api.AuxCounterweightName[Convert.ToInt16(Math.Round(Swarm[i].Position[1],
0))];
            GlobalBest.BestPositionText[10] = Convert.ToInt16(Math.Round(Swarm[i].Position[10],
0)).ToString() + " x" + api.AuxCounterweightName[Convert.ToInt16(Math.Round(Swarm[i].Position[2],
0))];
            GlobalBest.BestPositionText[11] = Convert.ToInt16(Math.Round(Swarm[i].Position[11],
0)).ToString() + " x" + api.AuxCounterweightName[Convert.ToInt16(Math.Round(Swarm[i].Position[3],
0))];
            GlobalBest.BestPositionText[12] = Swarm[i].Tau.ToString() + " deg";
            GlobalBest.BestPositionText[13] = Swarm[i].CLF.ToString();

            GlobalBest.BestFitnessValue = Swarm[i].BestFitnessValue;
            label7.Text = GlobalBest.BestFitnessValue.ToString();
            Application.DoEvents();
          }
        }

        for (int iter = 0; iter < MaxIteration; iter++)
        {
          labelSwarmSize.Text = $"{iter}. iteration";

          for (int i = 0; i < SwarmSize; i++)
          {
            labelMaxIteration.Text = $"{i}. swarm";
            Application.DoEvents();
            for (int j = 0; j < DimensionSize; j++)
            {
              Swarm[i].Speed[j] = W * Swarm[i].Speed[j] + C1 * Math.Round(Randd.NextDouble(), 5) *
(Swarm[i].BestPosition[j] - Swarm[i].Position[j]) + C2 * Math.Round(Randd.NextDouble(), 5) *
(GlobalBest.BestPosition[j] - Swarm[i].Position[j]);
              Swarm[i].Speed[j] = UpdateSwarmSpeed(Swarm[i].Speed[j]);
```

```
                if (j > 3 && j < 8)
                {
                    UpperLimits[j] = CrankLength - api.CounterweightdM[Convert.ToInt16(Math.Round(Swarm[i].Position[j - 4], 0))];
                    LowerLimits[j] = api.Counterweightdm[Convert.ToInt16(Math.Round(Swarm[i].Position[j - 4], 0))];
                }
                Swarm[i].Position[j] = UpdateSwarmPosition(Swarm[i].Position[j], Swarm[i].Speed[j], UpperLimits[j], LowerLimits[j]);
            }

            Swarm[i].FitnessValue = FitnessCalculation(Swarm[i].Position, false, false);

            if (Swarm[i].FitnessValue < Swarm[i].BestFitnessValue)
            {
                Swarm[i].BestPosition = (double[])Swarm[i].Position.Clone();
                Swarm[i].BestFitnessValue = Swarm[i].FitnessValue;

                if (Swarm[i].BestFitnessValue < GlobalBest.BestFitnessValue)
                {
                    calc.RotaryInertialTorque.Clear();
                    calc.CounterBalanceTorque.Clear();
                    calc.NetGearboxTorque.Clear();

                    GlobalBest.BestPosition = (double[])Swarm[i].Position.Clone();
                    GlobalBest.BestFitnessValue = Swarm[i].BestFitnessValue;

                    Positionlist.Add(Swarm[i].Position.Clone());
                    Fitnesslist.Add(Swarm[i].BestFitnessValue);
                    label7.Text = GlobalBest.BestFitnessValue.ToString();

                    for (int j = 0; j < calc.RotaryInertialTorqueDummy.Count; j++)
                    {
                        calc.RotaryInertialTorque.Add(calc.RotaryInertialTorqueDummy[j]);
                        calc.CounterBalanceTorque.Add(calc.CounterBalanceTorqueDummy[j]);
                        calc.NetGearboxTorque.Add(calc.NetGearboxTorqueDummy[j]);
                    }

                    Draw();

                }
            }

        }
        W *= Wdamp;

    }
}

public double[] PSO_RandPosition(double[] a, double[] b, int n)
```

```
        {
            double[] x = new double[n];
            for (int i = 0; i < n; i++)
            {
                x[i] = PSO_Rand(a[i], b[i]);
                if (i > 3 && i < 8)
                {
                    x[i] = PSO_Rand(api.Counterweightdm[Convert.ToInt16(Math.Round(x[i - 4], 0))], CrankLength
- api.CounterweightdM[Convert.ToInt16(Math.Round(x[i - 4], 0))]);
                }
            }
            return x;
        }

public double[] PSO_Rand(double[] a, double[] b, int n)
        {
            double[] x = new double[n];
            for (int i = 0; i < n; i++)
            {
                x[i] = PSO_Rand(a[i], b[i]);
            }
            return x;
        }

public double FitnessCalculation(double[] a, bool noinertia, bool CLF)
        {
            int Cw1topID = Convert.ToInt16(Math.Round(a[0], 0));
            int Cw1botID = Convert.ToInt16(Math.Round(a[1], 0));
            int Cw2topID = Convert.ToInt16(Math.Round(a[2], 0));
            int Cw2botID = Convert.ToInt16(Math.Round(a[3], 0));
            double M1 = CrankLength - api.CounterweightdM[Cw1topID];
            double M2 = CrankLength - api.CounterweightdM[Cw1botID];
            double M3 = CrankLength - api.CounterweightdM[Cw2topID];
            double M4 = CrankLength - api.CounterweightdM[Cw2botID];
            double D1 = Math.Round(a[4], 1);
            double D2 = Math.Round(a[5], 1);
            double D3 = Math.Round(a[6], 1);
            double D4 = Math.Round(a[7], 1);

            if (D1 > M1)
            {
                D1 = M1;
            }

            if (D2 > M2)
            {
                D2 = M2;
            }

            if (D3 > M3)
```

```
{
  D3 = M3;
}

if (D4 > M4)
{
  D4 = M4;
}

double Y1 = api.CounterweightY[Cw1topID];
double Y2 = api.CounterweightY[Cw1botID];
double Y3 = api.CounterweightY[Cw2topID];
double Y4 = api.CounterweightY[Cw2botID];
double H1 = Math.Sqrt((Y1 + CrankHalfwidth) * (Y1 + CrankHalfwidth) + (M1 - D1) * (M1 - D1));
double H2 = Math.Sqrt((Y2 + CrankHalfwidth) * (Y2 + CrankHalfwidth) + (M2 - D2) * (M2 - D2));
double H3 = Math.Sqrt((Y3 + CrankHalfwidth) * (Y3 + CrankHalfwidth) + (M3 - D3) * (M3 - D3));
double H4 = Math.Sqrt((Y4 + CrankHalfwidth) * (Y4 + CrankHalfwidth) + (M4 - D4) * (M4 - D4));

int AuxCw1topID = Convert.ToInt16(Math.Round(a[8], 0));
int AuxCw1botID = Convert.ToInt16(Math.Round(a[9], 0));
int AuxCw2topID = Convert.ToInt16(Math.Round(a[10], 0));
int AuxCw2botID = Convert.ToInt16(Math.Round(a[11], 0));

double Icg1 = api.CounterweightMoment[Cw1topID] + AuxCw1topID * api.AuxCounterweightMoment[Cw1topID];
double Icg2 = api.CounterweightMoment[Cw1botID] + AuxCw1botID * api.AuxCounterweightMoment[Cw1botID];
double Icg3 = api.CounterweightMoment[Cw2topID] + AuxCw2topID * api.AuxCounterweightMoment[Cw2topID];
double Icg4 = api.CounterweightMoment[Cw2botID] + AuxCw2botID * api.AuxCounterweightMoment[Cw2botID];

double mcw1 = api.CounterweightMass[Cw1topID] + AuxCw1topID * api.AuxCounterweightMass[Cw1topID];
double mcw2 = api.CounterweightMass[Cw1botID] + AuxCw1botID * api.AuxCounterweightMass[Cw1botID];
double mcw3 = api.CounterweightMass[Cw2topID] + AuxCw2topID * api.AuxCounterweightMass[Cw2topID];
double mcw4 = api.CounterweightMass[Cw2botID] + AuxCw2botID * api.AuxCounterweightMass[Cw2botID];

double Icw1 = Icg1 + mcw1 * (H1 / 12) * (H1 / 12);
double Icw2 = Icg2 + mcw2 * (H2 / 12) * (H2 / 12);
double Icw3 = Icg3 + mcw3 * (H3 / 12) * (H3 / 12);
double Icw4 = Icg4 + mcw4 * (H4 / 12) * (H4 / 12);

double Icw = Icw1 + Icw2 + Icw3 + Icw4;

double Is = Icw + MassMomentCranks + MassMomentGearbox;
```

```
double Tcbmax = CrankTorque + mcw1 * (M1 - D1) + mcw2 * (M2 - D2) + mcw3 * (M3 - D3) + mcw4 * (M4 - D4);

double Sumx = (CrankMass * 2 * CrankLength / 2 + mcw1 * (M1 - D1) + mcw2 * (M2 - D2) + mcw3 * (M3 - D3) + mcw4 * (M4 - D4)) / (CrankMass * 2 + mcw1 + mcw2 + mcw3 + mcw4);
double Sumy = (mcw1 * (Y1 + CrankHalfwidth) - mcw2 * (Y2 + CrankHalfwidth) + mcw3 * (Y3 + CrankHalfwidth) - mcw4 * (Y4 + CrankHalfwidth)) / (CrankMass * 2 + mcw1 + mcw2 + mcw3 + mcw4);
double Taumod = Math.Round(Math.Atan(Sumy / Sumx) * 180 / Math.PI, 2);



calc.RotaryInertialTorqueDummy.Clear();
calc.CounterBalanceTorqueDummy.Clear();
calc.NetGearboxTorqueDummy.Clear();

for (int j = 0; j < twm.PolishedRodPosition.Count; j++)
{
  calc.CounterBalanceTorqueDummy.Add(Math.Round(-Math.Sin((Taumod + PhaseAngle + calc.CrankAngle[j]) / 180 * Math.PI) * Tcbmax / 1000, 3));
}

if (!noinertia)
{
  for (int j = 0; j < twm.PolishedRodPosition.Count; j++)
  {
    calc.RotaryInertialTorqueDummy.Add(Math.Round(12 / 32.2 * Is * calc.CrankAngularAcceleration5Point[j] / 1000, 3));
  }
}
else
{
  calc.ArticulatingInertialTorqueSymmetricalNoInertia.Clear();
  for (int j = 0; j < twm.PolishedRodPosition.Count; j++)
  {
    calc.RotaryInertialTorqueDummy.Add(0);
    calc.ArticulatingInertialTorqueSymmetricalNoInertia.Add(0);
  }
}

if (!noinertia)
{
  for (int j = 0; j < twm.PolishedRodPosition.Count; j++)
  {
    calc.NetGearboxTorqueDummy.Add(Math.Round(calc.RodTorque[j] + calc.ArticulatingInertialTorqueSvinos5p[j] + calc.RotaryInertialTorqueDummy[j] + calc.CounterBalanceTorqueDummy[j], 3));
  }
}
else
{
```

```
            for (int j = 0; j < twm.PolishedRodPosition.Count; j++)
            {
                calc.NetGearboxTorqueDummy.Add(Math.Round(calc.RodTorque[j] + calc.ArticulatingInertialTorqueSymmetricalNoInertia[j] + calc.RotaryInertialTorqueDummy[j] + calc.CounterBalanceTorqueDummy[j], 3));
            }
        }


        double max = 0;

        if (!CLF)
        {
            for (int i = 0; i < twm.PolishedRodPosition.Count; i++)
            {
                if (Math.Abs(calc.NetGearboxTorqueDummy[i]) > max)
                {
                    max = Math.Abs(calc.NetGearboxTorqueDummy[i]);
                }
            }
        }

        else
        {
            double Squaresum = 0;
            double Sum = 0;
            for (int i = 0; i < twm.PolishedRodPosition.Count - 1; i++)
            {
                Squaresum += (Math.Pow(calc.NetGearboxTorqueDummy[i], 2) + Math.Pow(calc.NetGearboxTorqueDummy[i + 1], 2)) / 2 * twm.Time[1];
                Sum += (calc.NetGearboxTorqueDummy[i] + calc.NetGearboxTorqueDummy[i]) / 2 * twm.Time[1];
            }

            max = Math.Round(Math.Sqrt(Squaresum / twm.Time[twm.Time.Count - 1]) / (Sum / twm.Time[twm.Time.Count - 1]), 4);
        }


        return max;
    }

public double TauCalculation(double[] a)
    {
        int Cw1topID = Convert.ToInt16(Math.Round(a[0], 0));
        int Cw1botID = Convert.ToInt16(Math.Round(a[1], 0));
        int Cw2topID = Convert.ToInt16(Math.Round(a[2], 0));
        int Cw2botID = Convert.ToInt16(Math.Round(a[3], 0));
        double M1 = CrankLength - api.CounterweightdM[Cw1topID];
        double M2 = CrankLength - api.CounterweightdM[Cw1botID];
```

```
double M3 = CrankLength - api.CounterweightdM[Cw2topID];
double M4 = CrankLength - api.CounterweightdM[Cw2botID];
double D1 = Math.Round(Math.Round(a[4] * 2.54, 0) / 2.54, 1);
double D2 = Math.Round(Math.Round(a[5] * 2.54, 0) / 2.54, 1);
double D3 = Math.Round(Math.Round(a[6] * 2.54, 0) / 2.54, 1);
double D4 = Math.Round(Math.Round(a[7] * 2.54, 0) / 2.54, 1);

if (D1 > M1)
{
  D1 = M1;
}

if (D2 > M2)
{
  D2 = M2;
}

if (D3 > M3)
{
  D3 = M3;
}

if (D4 > M4)
{
  D4 = M4;
}

double Y1 = api.CounterweightY[Cw1topID];
double Y2 = api.CounterweightY[Cw1botID];
double Y3 = api.CounterweightY[Cw2topID];
double Y4 = api.CounterweightY[Cw2botID];
double H1 = Math.Sqrt((Y1 + CrankHalfwidth) * (Y1 + CrankHalfwidth) + (M1 - D1) * (M1 - D1));
double H2 = Math.Sqrt((Y2 + CrankHalfwidth) * (Y2 + CrankHalfwidth) + (M2 - D2) * (M2 - D2));
double H3 = Math.Sqrt((Y3 + CrankHalfwidth) * (Y3 + CrankHalfwidth) + (M3 - D3) * (M3 - D3));
double H4 = Math.Sqrt((Y4 + CrankHalfwidth) * (Y4 + CrankHalfwidth) + (M4 - D4) * (M4 - D4));

int AuxCw1topID = Convert.ToInt16(Math.Round(a[8], 0));
int AuxCw1botID = Convert.ToInt16(Math.Round(a[9], 0));
int AuxCw2topID = Convert.ToInt16(Math.Round(a[10], 0));
int AuxCw2botID = Convert.ToInt16(Math.Round(a[11], 0));

double Icg1 = api.CounterweightMoment[Cw1topID] + AuxCw1topID * api.AuxCounterweightMoment[Cw1topID];
double Icg2 = api.CounterweightMoment[Cw1botID] + AuxCw1botID * api.AuxCounterweightMoment[Cw1botID];
double Icg3 = api.CounterweightMoment[Cw2topID] + AuxCw2topID * api.AuxCounterweightMoment[Cw2topID];
double Icg4 = api.CounterweightMoment[Cw2botID] + AuxCw2botID * api.AuxCounterweightMoment[Cw2botID];
```

```
    double mcw1 = api.CounterweightMass[Cw1topID] + AuxCw1topID * api.AuxCounterweightMass[Cw1topID];
    double mcw2 = api.CounterweightMass[Cw1botID] + AuxCw1botID * api.AuxCounterweightMass[Cw1botID];
    double mcw3 = api.CounterweightMass[Cw2topID] + AuxCw2topID * api.AuxCounterweightMass[Cw2topID];
    double mcw4 = api.CounterweightMass[Cw2botID] + AuxCw2botID * api.AuxCounterweightMass[Cw2botID];

    double Icw1 = Icg1 + mcw1 * (H1 / 12) * (H1 / 12);
    double Icw2 = Icg2 + mcw2 * (H2 / 12) * (H2 / 12);
    double Icw3 = Icg3 + mcw3 * (H3 / 12) * (H3 / 12);
    double Icw4 = Icg4 + mcw4 * (H4 / 12) * (H4 / 12);

    double Icw = Icw1 + Icw2 + Icw3 + Icw4;

    double Is = Icw + MassMomentCranks + MassMomentGearbox;

    double Tcbmax = CrankTorque + mcw1 * (M1 - D1) + mcw2 * (M2 - D2) + mcw3 * (M3 - D3) + mcw4 * (M4 - D4);

    double Sumx = (CrankMass * 2 * CrankLength / 2 + mcw1 * (M1 - D1) + mcw2 * (M2 - D2) + mcw3 * (M3 - D3) + mcw4 * (M4 - D4)) / (CrankMass * 2 + mcw1 + mcw2 + mcw3 + mcw4);
    double Sumy = (mcw1 * (Y1 + CrankHalfwidth) - mcw2 * (Y2 + CrankHalfwidth) + mcw3 * (Y3 + CrankHalfwidth) - mcw4 * (Y4 + CrankHalfwidth)) / (CrankMass * 2 + mcw1 + mcw2 + mcw3 + mcw4);
    double Taumod = Math.Round(Math.Atan(Sumy / Sumx) * 180 / Math.PI, 2);

    return Taumod;

  }

public double CLFCalculation(double[] a)
  {
    int Cw1topID = Convert.ToInt16(Math.Round(a[0], 0));
    int Cw1botID = Convert.ToInt16(Math.Round(a[1], 0));
    int Cw2topID = Convert.ToInt16(Math.Round(a[2], 0));
    int Cw2botID = Convert.ToInt16(Math.Round(a[3], 0));
    double M1 = CrankLength - api.CounterweightdM[Cw1topID];
    double M2 = CrankLength - api.CounterweightdM[Cw1botID];
    double M3 = CrankLength - api.CounterweightdM[Cw2topID];
    double M4 = CrankLength - api.CounterweightdM[Cw2botID];
    double D1 = Math.Round(Math.Round(a[4] * 2.54, 0) / 2.54, 1);
    double D2 = Math.Round(Math.Round(a[5] * 2.54, 0) / 2.54, 1);
    double D3 = Math.Round(Math.Round(a[6] * 2.54, 0) / 2.54, 1);
    double D4 = Math.Round(Math.Round(a[7] * 2.54, 0) / 2.54, 1);

    if (D1 > M1)
    {
      D1 = M1;
    }
```

```
if (D2 > M2)
{
   D2 = M2;
}

if (D3 > M3)
{
   D3 = M3;
}

if (D4 > M4)
{
   D4 = M4;
}

double Y1 = api.CounterweightY[Cw1topID];
double Y2 = api.CounterweightY[Cw1botID];
double Y3 = api.CounterweightY[Cw2topID];
double Y4 = api.CounterweightY[Cw2botID];
double H1 = Math.Sqrt((Y1 + CrankHalfwidth) * (Y1 + CrankHalfwidth) + (M1 - D1) * (M1 - D1));
double H2 = Math.Sqrt((Y2 + CrankHalfwidth) * (Y2 + CrankHalfwidth) + (M2 - D2) * (M2 - D2));
double H3 = Math.Sqrt((Y3 + CrankHalfwidth) * (Y3 + CrankHalfwidth) + (M3 - D3) * (M3 - D3));
double H4 = Math.Sqrt((Y4 + CrankHalfwidth) * (Y4 + CrankHalfwidth) + (M4 - D4) * (M4 - D4));

int AuxCw1topID = Convert.ToInt16(Math.Round(a[8], 0));
int AuxCw1botID = Convert.ToInt16(Math.Round(a[9], 0));
int AuxCw2topID = Convert.ToInt16(Math.Round(a[10], 0));
int AuxCw2botID = Convert.ToInt16(Math.Round(a[11], 0));

double Icg1 = api.CounterweightMoment[Cw1topID] + AuxCw1topID * api.AuxCounterweightMoment[Cw1topID];
double Icg2 = api.CounterweightMoment[Cw1botID] + AuxCw1botID * api.AuxCounterweightMoment[Cw1botID];
double Icg3 = api.CounterweightMoment[Cw2topID] + AuxCw2topID * api.AuxCounterweightMoment[Cw2topID];
double Icg4 = api.CounterweightMoment[Cw2botID] + AuxCw2botID * api.AuxCounterweightMoment[Cw2botID];

double mcw1 = api.CounterweightMass[Cw1topID] + AuxCw1topID * api.AuxCounterweightMass[Cw1topID];
double mcw2 = api.CounterweightMass[Cw1botID] + AuxCw1botID * api.AuxCounterweightMass[Cw1botID];
double mcw3 = api.CounterweightMass[Cw2topID] + AuxCw2topID * api.AuxCounterweightMass[Cw2topID];
double mcw4 = api.CounterweightMass[Cw2botID] + AuxCw2botID * api.AuxCounterweightMass[Cw2botID];

double Icw1 = Icg1 + mcw1 * (H1 / 12) * (H1 / 12);
double Icw2 = Icg2 + mcw2 * (H2 / 12) * (H2 / 12);
```

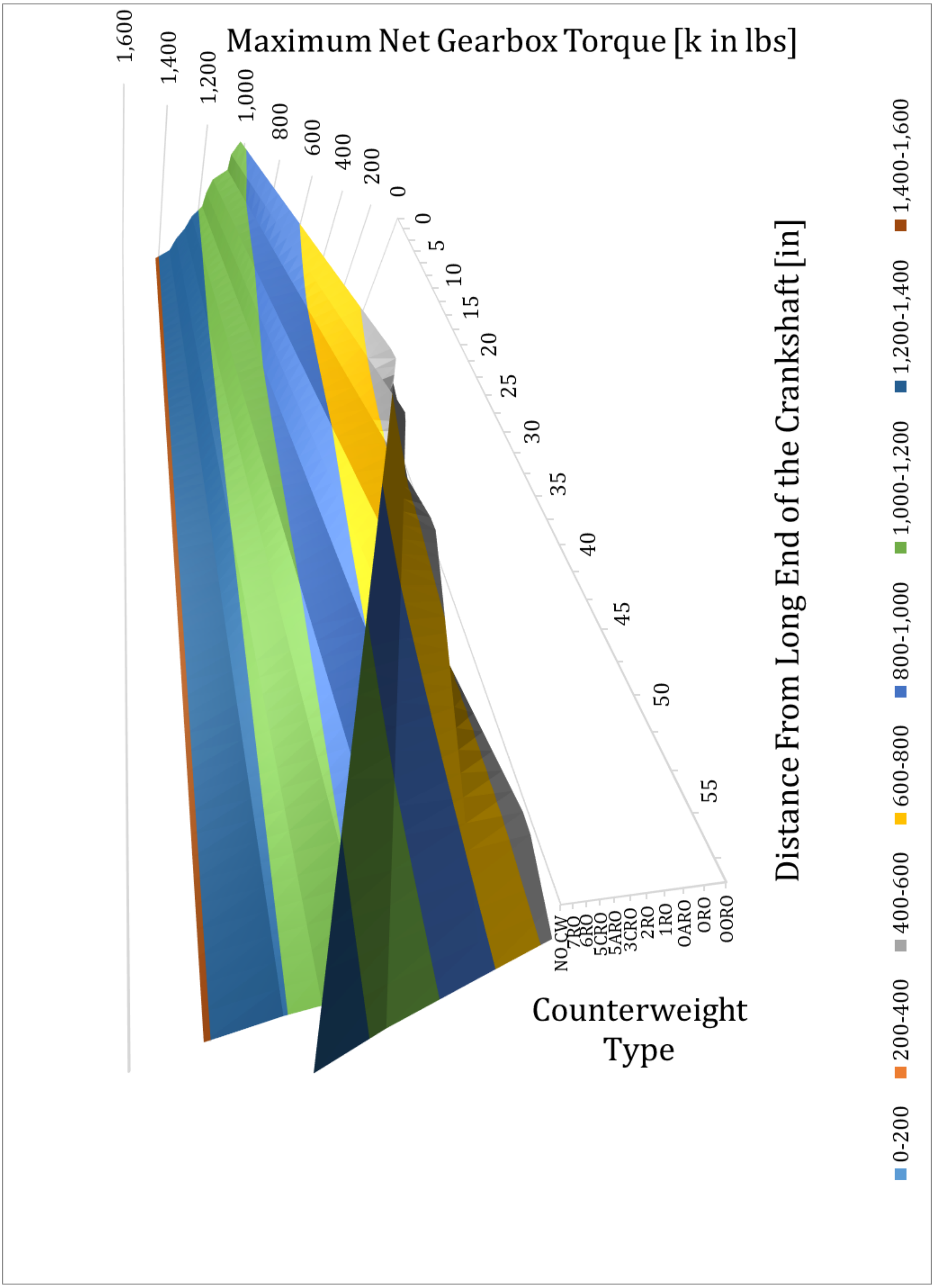
```
        double Icw3 = Icg3 + mcw3 * (H3 / 12) * (H3 / 12);
        double Icw4 = Icg4 + mcw4 * (H4 / 12) * (H4 / 12);

        double Icw = Icw1 + Icw2 + Icw3 + Icw4;

        double Is = Icw + MassMomentCranks + MassMomentGearbox;

        double Tcbmax = CrankTorque + mcw1 * (M1 - D1) + mcw2 * (M2 - D2) + mcw3 * (M3 - D3) + mcw4
* (M4 - D4);

        double Sumx = (CrankMass * 2 * CrankLength / 2 + mcw1 * (M1 - D1) + mcw2 * (M2 - D2) + mcw3 *
(M3 - D3) + mcw4 * (M4 - D4)) / (CrankMass * 2 + mcw1 + mcw2 + mcw3 + mcw4);
        double Sumy = (mcw1 * (Y1 + CrankHalfwidth) - mcw2 * (Y2 + CrankHalfwidth) + mcw3 * (Y3 +
CrankHalfwidth) - mcw4 * (Y4 + CrankHalfwidth)) / (CrankMass * 2 + mcw1 + mcw2 + mcw3 + mcw4);
        double Taumod = Math.Round(Math.Atan(Sumy / Sumx) * 180 / Math.PI, 2);



        calc.RotaryInertialTorqueDummy.Clear();
        calc.CounterBalanceTorqueDummy.Clear();
        calc.NetGearboxTorqueDummy.Clear();

        for (int j = 0; j < twm.PolishedRodPosition.Count; j++)
        {
          calc.CounterBalanceTorqueDummy.Add(Math.Round(-Math.Sin((Taumod + PhaseAngle +
calc.CrankAngle[j]) / 180 * Math.PI) * Tcbmax / 1000, 3));
        }

        for (int j = 0; j < twm.PolishedRodPosition.Count; j++)
        {
          calc.RotaryInertialTorqueDummy.Add(Math.Round(12 / 32.2 * Is *
calc.CrankAngularAcceleration5Point[j] / 1000, 3));
        }



        for (int j = 0; j < twm.PolishedRodPosition.Count; j++)
        {
          calc.NetGearboxTorqueDummy.Add(Math.Round(calc.RodTorque[j] +
calc.ArticulatingInertialTorqueSvinos5p[j] + calc.RotaryInertialTorqueDummy[j] +
calc.CounterBalanceTorqueDummy[j], 3));
        }



        double max = 0;

        double Squaresum = 0;
        double Sum = 0;
        for (int i = 0; i < twm.PolishedRodPosition.Count - 1; i++)
```

```
        {
          Squaresum += (Math.Pow(calc.NetGearboxTorqueDummy[i], 2) + Math.Pow(calc.NetGearboxTorqueDummy[i + 1], 2)) / 2 * twm.Time[1];
          Sum += (calc.NetGearboxTorqueDummy[i] + calc.NetGearboxTorqueDummy[i]) / 2 * twm.Time[1];
        }

        max = Math.Round(Math.Sqrt(Squaresum / twm.Time[twm.Time.Count - 1]) / (Sum / twm.Time[twm.Time.Count - 1]), 4);



        return max;
      }


      double UpdateSwarmPosition(double Pos, double Speed, double upperlimit, double lowerlimit)
      {
        double OutPos = Pos + Speed;
        OutPos = Math.Max(Math.Min(OutPos, upperlimit), lowerlimit);
        return OutPos;
      }

      double UpdateSwarmSpeed(double Speed)
      {
        double OutPos = Math.Max(Math.Min(Speed, ub_SpeedXi), lb_SpeedXi);
        return OutPos;
      }

public struct Particle
      {
        public double[] Position;
        public double[] Speed;
        public double FitnessValue;
        public double[] BestPosition;
        public string[] BestPositionText;
        public double BestFitnessValue;
        public double Tau;
        public double CLF;
      }
```

14.3 Appendix C Results of the Sensitivity Analysis

	Distance from long end of the crank [in]																													
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
NO CW	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417
7RO	1340	1338	1336	1334	1332	1330	1328	1325	1323	1321	1319	1317	1315	1313	1310	1308	1306	1304	1302	1299	1297	1295	1293	1291	1288	1286	1284	1282	1279	1277
6RO	1310	1306	1303	1300	1297	1294	1291	1288	1285	1281	1278	1275	1272	1269	1266	1262	1259	1256	1253	1250	1246	1243	1240	1237	1233	1230	1227	1224	1220	1217
5CRO	1265	1260	1256	1251	1247	1242	1237	1233	1228	1224	1219	1214	1209	1205	1200	1195	1191	1186	1181	1176	1172	1167	1162	1157	1152	1147	1143	1138	1133	1128
5ARO	1234	1228	1223	1217	1211	1205	1200	1194	1188	1182	1177	1171	1165	1159	1153	1147	1141	1135	1129	1123	1118	1112	1106	1100	1094	1087	1081	1075	1069	1063
3CRO	1182	1174	1165	1157	1148	1140	1131	1122	1114	1105	1097	1088	1079	1070	1062	1053	1044	1035	1027	1018	1009	1000	991	982	973	964	955	946	937	928
2RO	1168	1158	1148	1138	1128	1118	1108	1098	1088	1078	1068	1057	1047	1037	1027	1016	1006	996	985	975	964	954	943	933	922	912	901	891	880	869
1RO	1142	1131	1119	1108	1096	1085	1073	1062	1050	1039	1027	1016	1004	992	981	969	957	945	933	921	910	898	886	874	862	850	838	825	813	801
OARO	1070	1056	1041	1026	1011	996	981	966	951	936	921	906	891	876	860	845	830	814	799	783	768	752	737	721	706	690	674	658	643	627
ORO	1063	1044	1025	1006	987	968	949	930	910	891	872	853	833	814	794	775	755	736	716	696	676	657	637	617	597	577	557	537	517	496
OORO	1023	1003	982	961	940	919	898	877	856	834	813	792	770	749	727	706	684	662	641	619	597	575	553	531	509	502	526	550	574	598

	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
NO CW	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417	1417
7RO	1275	1273	1270	1268	1266	1264	1261	1259	1257	1255	1253	1250	1248	1246	1244	1242	1239	1237	1235	1233	1231	1228	1226	1224	1222	1220	1218	1215	1213	1211
6RO	1214	1210	1207	1204	1200	1197	1194	1190	1187	1184	1180	1177	1173	1170	1167	1163	1160	1156	1153	1149	1146	1143	1140	1136	1133	1130	1127	1123	1120	1117
5CRO	1123	1118	1113	1108	1103	1098	1093	1088	1083	1078	1073	1068	1063	1058	1053	1048	1043	1038	1033	1028	1023	1017	1012	1007	1002	997	992	987	982	978
5ARO	1057	1051	1045	1039	1033	1026	1020	1014	1008	1001	995	989	983	976	970	964	957	951	945	938	932	925	919	913	906	900	893	887	881	875
3CRO	919	910	901	892	883	873	864	855	846	836	827	818	808	799	790	780	771	761	752	743	733	723	714	704	695	685	675	666	656	646
2RO	859	848	837	826	816	805	794	783	772	761	750	739	728	717	706	695	684	673	662	651	639	628	617	606	594	583	572	560	549	537
1RO	789	777	765	752	740	728	715	703	690	678	665	653	640	628	615	603	590	577	564	552	539	526	513	500	488	489	502	516	529	542
OARO	611	595	579	563	547	531	515	499	497	513	530	546	563	581	598	615	632	649	666	684	701	718	735	752	769	787	804	821	838	855
ORO	514	536	558	579	601	623	644	666	688	710	732	754	777	799	821	843	866	888	910	932	954	976	999	1021	1043	1065	1087	1110	1132	1154
OORO	622	646	670	694	718	743	767	792	817	841	866	890	915	939	964	988	1013	1037	1062	1086	1111	1135	1160	1184	1209	1233	1258	1282	1307	1331


Maximum Net Gearbox Torque [k in lbs]
1,600
1,400
1,200
1,000
800
600
400
200
0
0
5
10
15
20
25
30
35
40
45
50
55
Distance From Long End of the Crankshaft [in]
NO CW
7RO
6RO
5CRO
5ARO
3CRO
2RO
1RO
OARO
ORO
OORO
Counterweight Type
0-200
200-400
400-600
600-800
800-1,000
1,000-1,200
1,200-1,400
1,400-1,600