Universität der Bundeswehr München

Linda Eckel

Laboratoy Experiments of Wave Interactions on Submerged Oscillating Bodies

Bachelorthesis

Supervisior Universität der Bundeswehr München:
Prof. Dr. Oliver Meyer

Supervisor University of Dundee:
Dr. Masoud Hayatdavoodi

14. May 2018
Wehrtechnik MT 2015
Matriculation Number 1151148
Submission Date 30.09.2018

Abstract

Content of thesis is the experimental analysis of prime movers, whose field of application are wave enery devices in shallow water areas. These are in particular submerged point heaving devices with a linear generator. The basic oscillating motion behavior in vertical direction of different forms of the prime movers under varying conditions is investigated. The aim is to make statements about possible optimized forms and to suggest further test forms. Conditions are represented by different submerence depths of the prime movers, as well as wave period and wave height, and by different restoring forces of the connection between prime movers and linear generators. The mentioned restoring forces are simulated in the experiments by a spring force. The horizontal position of the prime movers is fixed, the vertical movement is guided by linear bearings on a shaft. The test forms used are a circular plate and a cone whose surface shape has been adapted to an optimized surface function of surface heaving devices with asymmetrical prime movers. Test environment is a wave flume with piston wave maker.

Furthermore, various measuring programs for determining the oscillation are presented and described. These programs are able to evaluate video recordings of the experiments with regard to the movement of the prime movers. Matlab is used as the program interface. The presented programs are based on Gaussian Mixture model and color seperation. With the CamerCalibratorApp camera features are included.

Declaration

I hereby declare that the work contained in this document is my own work, and that has not been presented in a previous application for a higher degree.

Munich, 15th September 2018

Acknowledgements

First of all I would like to thank Dr. Hayatdavoodi for his unselfish support. Only because of this support, I was able to write my bachelor thesis in Scotland. Throughout the entire period, he was always available to help me and was also available to talk to me in his office outside of the usual regular appointments. Through professional discussions, he was always able to open my perspective to new aspects and motivate me anew. The experiences I gained at the University of Dundee have had a lasting impact on me and will provide me with a foundation for my future career.

I would also like to thank the German Armed Forces. They offered me financial support for my student stay abroad. The participation in this project abroad has strongly promoted my development.

List of Abbreviations

GMM	Gaussian Mixture Model
PTO	Power-take-off system
WEC	Wave Energy Converter

Contents

List of Figures

List of Tables

1. Introduction

Renewable Energy

In this section the aspects of renewable energies are considered. Renewable energy means energy from renewable resources, which are naturally replenished on a time-scale of human life or sunlight, wind, rain, tides, waves, and geothermal heat[5].

Technologies for generating energy from renewable energy sources is a major issue in society. Due to the increasing global warming and its more visible effects, a rethinking of politics and society is beginning. One goal of this new way of thinking is to preserve the earth in such a way, that future generations can also use this habitat. This includes stopping the emission of greenhouse gases and creating long-term energy sources. Renewable energy sources should make a significant contribution to this.

Another aspect that draws attention to new energy sources, is the fact, that fossil fuels are sinking while the energy demand increases[6]. This energy consumption is mainly covered by fossil fuels which should be replaced by regenerative energies, Fig. 9.

Furthermore, the EU has to import about 55% of the total electricity demand and is therefore heavily dependent[7]. New sources of energy must be developed in order to prevent this increasing dependence and thus guarantee political and economic capacity to act. Since the majority of countries within the EU have few fossil resources, regenerative technologies are forward-looking. In particular, marine renewable energy offers far-reaching possibilities as explained below.

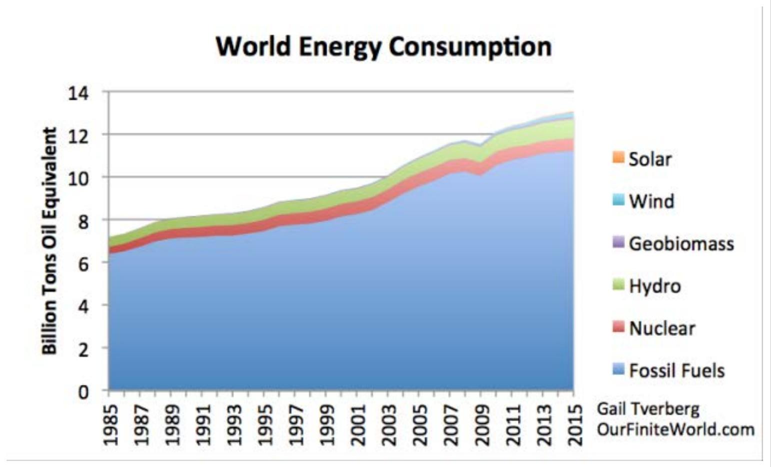


Figure 1: World Energy Demand[8].

The Ocean as an Energy Carrier

Oceans and seas are a possible source of renewable energies. Renewable energies produced by using marine sources and marine space are called marine renewable energy[7]. Offshore wind farms also belong to the category of marine renewable energy, since by definition they use marine space, Fig. 2.

Due to its large energy capacity, which is many times greater than global electricity demand, the ocean has an important position in the implementation of climate targets[9]. One of these climate targets is the reduction of CO_2 to 136 million tonnes[7] and many experts believe, that marine renewable energy is the best option to reach this goal[9].
However, this energy source has hardly been used to date. This is due to the various challenges that projects of this kind have to overcome. The ocean and the seas are used in many different ways. On the one hand through fishing, shipping, oil platforms but also through the sea inhabitants. All this must be taken into account in the development of regenerative technologies.[9]

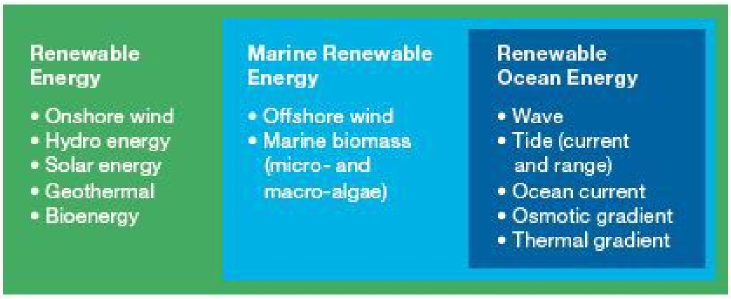


Figure 2: Delimitation of Maritime Renewable Energy[7].

Thesis Structure

This paper deals with the increasingly use of wave energy. Therefore, the following sections, are only dealing with so-called wave energy converters (WEC). The following section 4 describes and explains the model of the WEC on which this thesis is based. This is the foundation for the test setup and the test parameters in section 4.. Then follows the design and development of the test objects and their components, like bearings and springs. Moving on to the analysis of the experimental work, the measuring equipment and the programs for motion analysis of the test objects/prime mover are explained next. In section 8. a general error analysis and discussion takes place and finally in section 9. the results and the discussion of these.

2. Literature Review

Wave Energy Converter

WEC are systems that convert the energy, contained in waves, into other forms of energy to make them usable in form of electricity. Wave energy has the highest energy density of all renewable resources[10]. Since this energy source has rarely been used so far [6], Fig. 9, wave energy offers many extensive possibilities. Especially the coasts of Great Britain and France are large sea areas whose wave energy can be used[10]. However, many projects are still in the R&D stage and are not yet fully developed[10]. In general, WEC are divided into four groups according to their mode of function, shown in figure 3[6].
In general, WECs consist of four components. The first component is the foundation. The second the structure and the prime mover that captures the wave energy. This captured energy is converted in a power-take-off system (PTO) into another form of energy and this process is monitored by a control system.[11]

Heaving Devices

Heaving devices use the vertical oscillation of a body. This body, the prime mover, can be on the surface, like a buoy, as well as underwater. The wave movement raises and lowers the prime mover. This kinetic energy is transferred to the PTO, a linear generator. The linear generator is connected to the prime mover and the sea floor. If the prime mover goes up and down, only the rotor is moving.[11]
A linear generator corresponds to a generator whose rotor and stator have been unwound and are moved linear to each other.

Pitching Devices

Pitching devices are semi-submerged or surface floating cylinders. These cylinders are connected with special swivel joints, which are absorbing the wave-induced relative movements. The movements are absorbed by a hydraulic system and drive the pumps. This can be used to generate electricity.[10]

Oscillating water Columns

Oscillating water columns consist of

> "particular submerged hollow structures"

[10, p.16] and one or more turbines. This structure is opened under the water surface. The wave motion causes the water within the structure to sink and rise and causes a pressure change of the air above the water surface within the structure. This changes the pressure and drives a turbine that is used to generate electricity.[10]

Overtopping Devices

Overtopping Devices are using turbines to generate power like the oscillating water columns as described above. By collecting water from overturning waves in a higher reservoir, the turbines can be driven.[10]

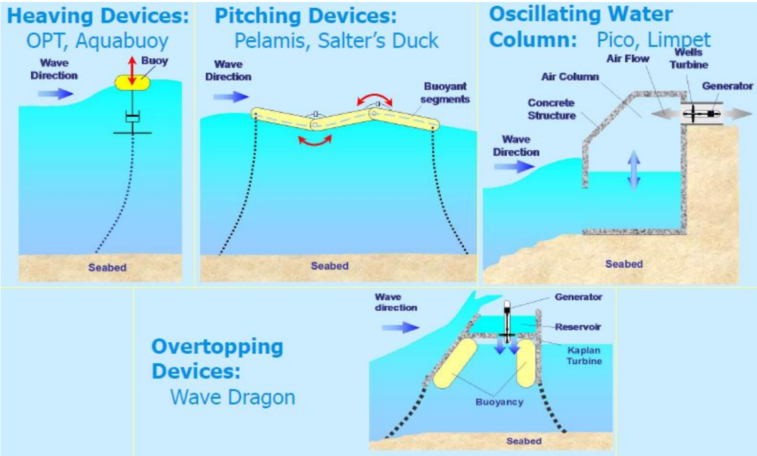


Figure 3: Types of Wave Energy Converter[6].

Current Challenges

Many systems are based on these four types. However, there are still many problems to solve. This includes irregular wave amplitude, phase and directions which has an impact on efficiency.[10]
During development, operating conditions are assume. For theses conditions the system is designed and works efficiently. Due to an irregular operating range, it is difficult to define the conditions and achieve high efficiency.
Furthermore, the irregular slow waves have an effect on the generator. The generator produces a constant frequency of 50 Hz and thus 500 times higher than the frequency of the waves, which additionally fluctuates. Despite these fluctuations, the generator must be able to generate electricity that can be fed into the power grid. The components are also exposed to extreme weather situations and must resist without damage.[10]

Another problem is the infrastructure and interaction with other users of the marine habitat. The existing infrastructure for power supply is designed for systems whose power supply can be controlled. This is not possible with wave energy, it depends on the current environmental conditions. Therefore, electricity storage capacities or other solutions must be created to compensate for such fluctuations.

It is also necessary for the WEC to lay supply connections on land, which means the construction of an underwater power grid.[12]
Furthermore, the ocean is a much-used habitat. On the one hand, marine life, on the other hand, shipping traffic, fishermen and other consumers who should not be influenced or restricted by WECs. This poses particular challenges to the development of noise for animal species and the closure of sea areas for shipping.[12]

Goals and Objectives

The content of this scientific work is the experimental analysis of the motion behaviour of different test objects with different shapes, which are used as prime movers in a WEC. The effects on the motion of the prime mover by different harmonic waves, submergence steps and shapes of the object, as well as different springs are investigated. The results of these tests should be used as a basis for optimising the prime movers and give an impression of the general behaviour of the various test shapes in order to develop and formulate new test conditions or test shapes.

Test parameters are wavelength, wave height, spring stiffness, submergenstep and form of the prime mover. The water depth remains constant and is assumed to be shallow water. Five different wavelengths, six wave heights, two springs and four submerged steps are tested. The tests are carried out on two test forms, a cone and a circular plate. Test environment is a waveflume with a pistonwavemaker. In this waveflume the test objects are mounted on a horizontal fixed shaft on different submergensetps. Test object and shaft are connected by a spring. The change in the spring stiffness of this link between shaft and object is also included in the tests. Measuring systems record the movement of the waves as well as the movement of the test objects.

3. The Wave Energy Device

This intendet experimental analysis of the motion behavior of a prime mover here is about the model of a WEC shown in Fig. 4. The prime mover is used later in a device, which is structured in this way or something similar. Figure 4 gives a possible impression of a constructive solution.

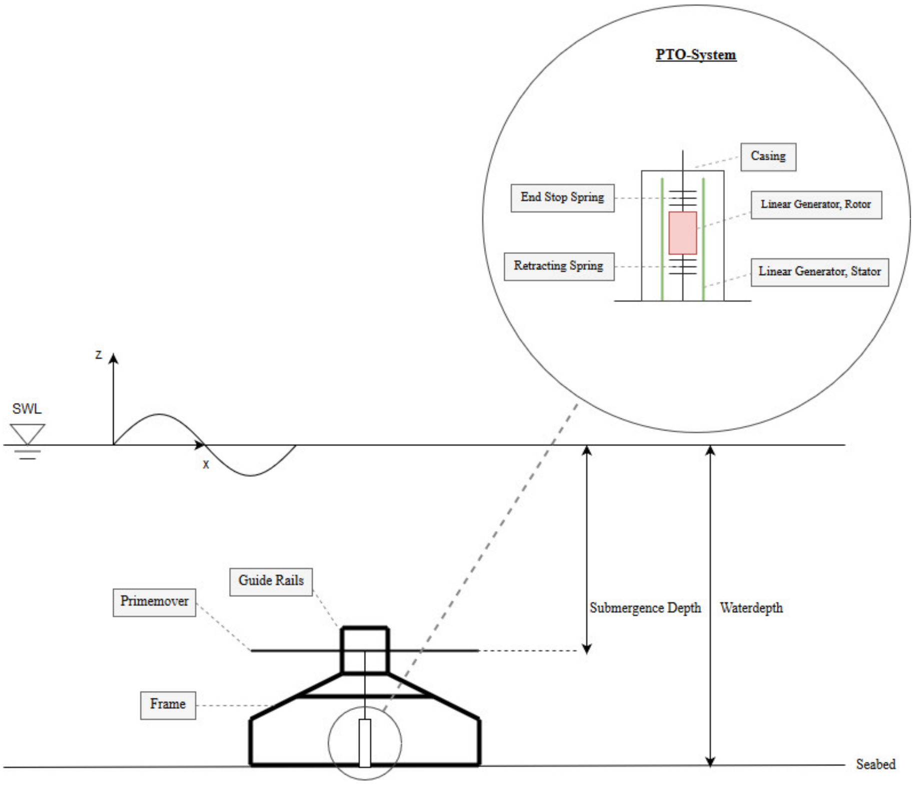


Figure 4: The Wave Energy Device.

The WEC presented here is a submerged heaving device. The idea and development of this WEC is originated after structual analysis. In this analysis material, wave load and the structure resbond were taken into account. The result is the design which is used here as well.[13]

The construction is as follows: The prime mover is located below the water surface and is therefore submerged. As only a vertical movement of the prime mover is allowed or preferred, it is a heaving device. Application area is shallow water. In our experiments, these are water depths of 0.3 m.

As explained in chapter 2., the WEC consists of the same basic building elements. A frame that is held in position by its attachment to the seabed. A prime mover that absorbs and transmits the wave energy and a PTO system that generates electricity from the movement of the prime mover. The prime mover can be mounted in various submerged steps.
The rectangular framework provides the necessary stability and foundation on the seabed. The prime mover is moved vertically up and down on four guided rails of the frame. This prevents horizontal deviation or rotation of the prime

movers. The prime mover is connected to the PTO system. This design consists of a linear generator, with the stator firmly anchored and the rotor connected to the prime mover. The Rotor shows limited possibilities of movements.
In the lower area, the rotor is attached by a spring, which causes a restoring force. In the upper area, a second spring limits the vertical deflection by the prime mover.[14]

With the lifting and lowering movement of the prime mover, power is generated by the linear generator. The power generation depends on the wave movement and thus the movement of the prime mover. This leads to the general problems with WEC mentioned in section 2.. It is difficult to control the power generation. The power P is not only dependent on the already mentioned factors of wave height H and wave period T, but also on the wathedepth h, the shape ans the size of the prime mover, as well as the submergence depth s, the bearing friction coefficient μ and the spring stiffness k. This is shown in equation 1.

$$P = f(H, T, h, shape, size, s, \mu, k) \tag{1}$$

4. Experiments

Experimental Assembly

Based on the device which is presented in section 4 an experimental assembly was developed. This is shown in figure 5 and is explained in the following. Note the coordinate system in this figure, which is used continuously.

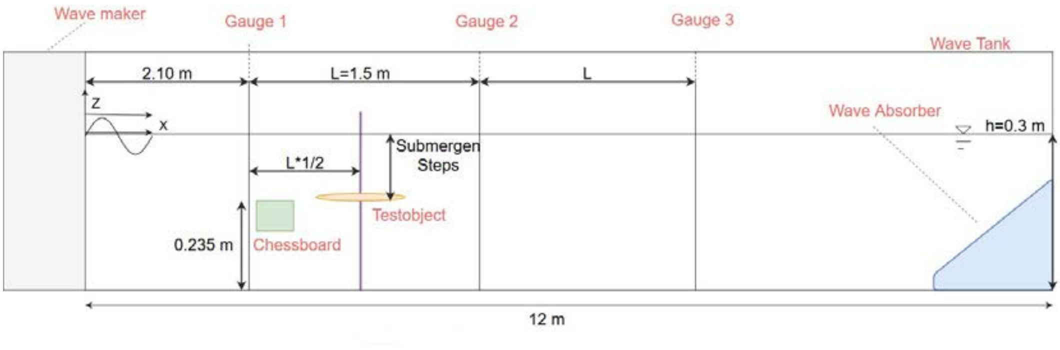


Figure 5: Overview of experimental assembly.

The experiments are carried out in a waveflume with a 12 m long wave tank. On the left side of the picture, the waves are generated by a wave maker and captured at the end of the wave tank by a wave absorber. This wave absorber, approx. 1.5 m long, is designed to prevent measurement distortion caused by reflection of returning waves.

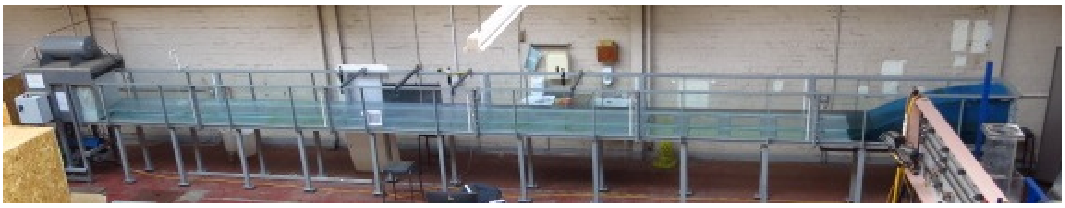


Figure 6: Wave flume.

A piston wave maker is used as wave maker. This is used to simulate waves in shallow water.
Generally there are three different types of wave makers. Piston, flap and plunger wave makers. The first two being the most common. Flap wave makers are used for depth water.[15]
This has the background that the water particles move on an orbital. In deep water, the orbital motion decreases and is no longer perceptible on the ground. In shallow water, the orbital trajectory becomes elliptical with increasing depth. The movement in z-direction decreases, whereas the movement in x-direction remains. This is illustrated in figure 7.[16]
The shafts are generated by three different systems: pneumatic, hydraulic or electric. The test equipment used here was operated pneumatically.[15]
The movement of the wave is detected by three measuring points. Gauge one is located at a distance of 2.1 m from the wave maker, the following two measuring

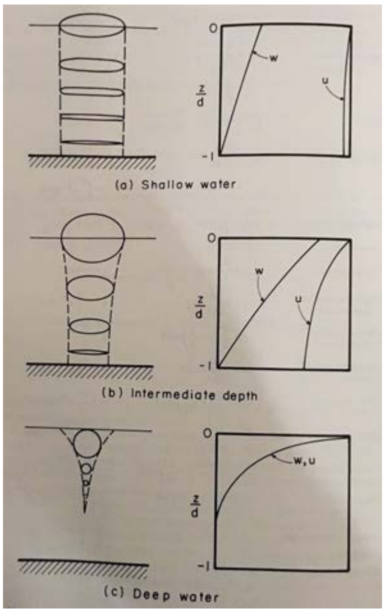


Figure 7: Particle orbits with depth[16].

points each have a distance of 1.5 m from the adjacent measuring point. All three gauges are mounted vertically at half the width of the wave tank.

The test object is located in the middle of the measuring points one and two and is driven by a linear bearing on a fixed shaft in vertical position, Fig. 8. Shaft and test object are connected by a spring, Fig. 8. This is based on the structure of the PTO-system. Both springs in the PTO behave like one spring on the connected driving prime mover. Therefore, only one spring is connected to the test object during the test.

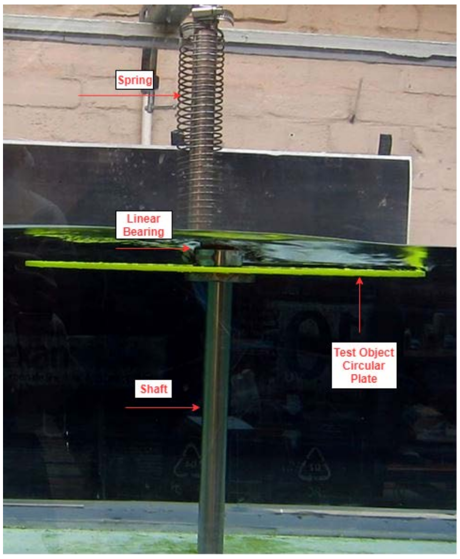


Figure 8: Structure of the circular object

The movement of the test object, a cone and a circular plate, is detected by a camera which position is approximately 1.5 m away from the wave tank. To determine the exact camera position and angle with the position of the test object, a chessboard is used for calculation. This will later serve as a reference with the help of calibration images and visibility in the video recordings. The origin of a coordinate system on the chessboard pattern, defined by the caliberization images, corresponds to a distance of 0.22 m to the bottom of the wave tank. This is important because the absolute movement in the shaft tank can be identified and not only the relative movement is known.

Experimental Parameters

As already mentioned in chapter 2., the tests are carried out with different test parameters, whereby the wave conditions and the test setup are changed. Table 1 shows some of the parameters used here and displays them visually. In contrast to the sketch, water depth is not designated as d but as h and is constantly 0.3 m.

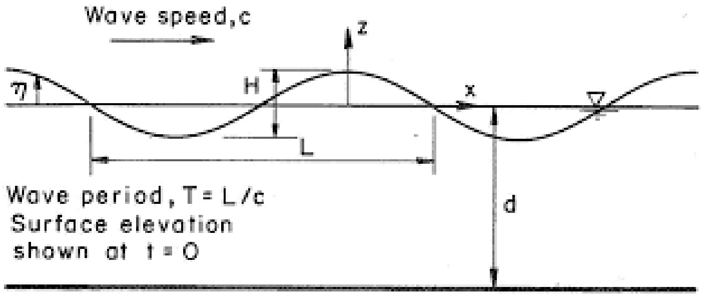


Figure 9: Sketch progressive wave train[16].

For test object one, all parameters except s_0 are tested, whereby for the periods T_4 and T_5 wave height H_1 is not tested. Test object two is only tested with spring one and s_0 and s_1 as well as H_6. This has the background that problems with water adsorption by the test object two has limitted the test time. A change in weight changes the behaviour and therefore misrepresents the results. However, this will be explained in the section of the describing and creating the test objects.

Table 1: Dimensionless experimental parameters.

$\bar{H}$= **Wave height**	$\bar{T}$= **Period**	$\bar{s}$= **Submergence depth**
$\bar{H}_1$= 0.034	$\bar{T}_1$= 4	$\bar{s}_0$= 0.0
$\bar{H}_2$= 0.067	$\bar{T}_2$= 5.72	$\bar{s}_1$= 0.1
$\bar{H}_3$= 0.1	$\bar{T}_3$= 7.15	$\bar{s}_2$= 0.3
$\bar{H}_4$= 0.134	$\bar{T}_4$= 8.58	$\bar{s}_3$= 0.5
$\bar{H}_5$= 0.167	$\bar{T}_5$= 10	$\bar{s}_4$= 0.7
$\bar{H}_6$= 0.2	–	–

In the next step, these parameters are referred to the water depth h which is used in the experiments, Eqs. (2) to (4).

$$\bar{T} = \frac{T}{\sqrt{\frac{g}{h}}} \tag{2}$$

$$\bar{H} = \frac{H}{h} \tag{3}$$

$$\bar{s} = s/h \tag{4}$$

This results in the following test parameters shown in table 2.

Table 2: Experimental parameters.

H= Wave height [m]	**T= Period [s]**	**s= Submergence depth [m]**	**IS= Inertial submergence depth [m]**
H_1= 0.01	T_1= 0.70	s_0= 0.00	IS_0= 0.30
H_2= 0.02	T_2= 1.00	s_1= 0.03	IS_1= 0.27
H_3= 0.03	T_3= 1.25	s_2= 0.09	IS_2= 0.21
H_4= 0.04	T_4= 1.50	s_3= 0.15	IS_3= 0.15
H_5= 0.05	T_5= 1.75	s_4= 0.21	IS_4= 0.09
H_6= 0.06	–	–	–

Due to the higher accuracy of the positioning of the test objects, the position of the objects is shown below as IS as the distance from the tank bottom to the upper edge of the test objects. This is expressed by equation (5).

$$IS = h - s \tag{5}$$

All parameters are combined in a way that both test objects are tested with two different springs. The test objects are tested for all theire wave conditions, i.e. period and wave height, which were explained before.

The water density ρ is constant during the experiments and is 997 kg/m^3.

5. Design and creation of test objects with setup

Circular Plate

Object Circular Plate

One of the test objects used here for the experiments is a circular plate made of Plexiglas, shown in figure 10. It has a diameter of 30 cm and a thickness of 3 mm. The object is prepared with a bearing to guide it later on a vertical fixed shaft. It therefore has a central circular opening of 32 mm diameter for the bearing. Unlike the cone described below, the plate is equipped with an LMF20UU linear bearing and can hold a shaft with a diameter of 20 mm. The diameter of the bearing and the shaft are small compared to the test object and can therefore be neglected. Bearing and shaft are connected by a spring. The test object with the bearing and the screws weights 432g.
In order to capture the transparent material with the video camera, a part of the edge is covered with coloured adhesive tape.

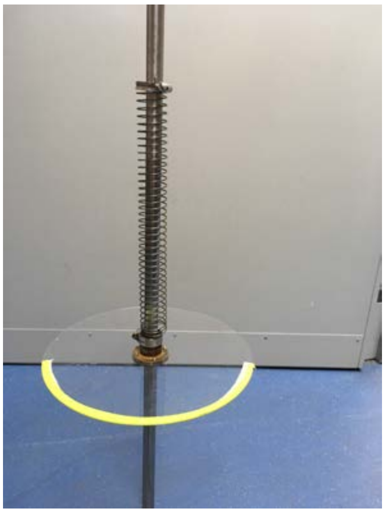


Figure 10: Installed test object one.

Object Cone

As already mentioned in chapter 4., a kind of cone is used as test object for WEC, Fig. 11. Shape and hydrodynamic design are based on the knowledge of an article about asymmetric, energy-capturing floating breakwater.
This scientific paper deals with the development of an optimal form of the prime mover which should be used for heaving devices with linear generator, Fig.12[17]. This is equivalent to the way the cone presented here is used as PTO, but the application of the energy capturing system under the water surface presented here is limited.
Berkeley and Wedges article is limited to the application on the water surface. Furthermore, the assumptions of the theoretical framework made there are consistent with the assumptions made in this thesis. The non-dimensionaled scaled form-function, in equation(6),[17] developed there was optimized with regard to

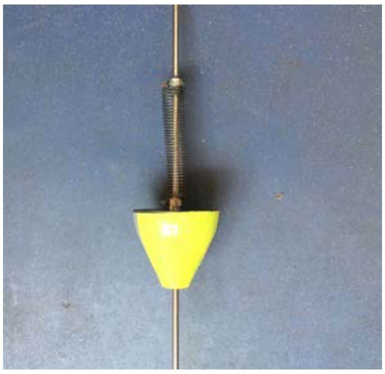


Figure 11: Installed test object one on the shaft and spring.

the influence of viscosity and high efficiency. At 45% of the optimal damping, an energy capture of 82% can be realized.[17]

$$F(\bar{y}) = \bar{x} = 0.05926(\bar{y}+1)^2 + 3.88147(\bar{y}+1)^3 - 2.94074(\bar{y}+1)^4[17] \qquad (6)$$

This developed function is modified in so far to the test conditions underwater, as only the range of the mathematical equation is considered, which describes the range in figure 14 under water. Furthermore, a symmetrical shape is realized. However, it should be noted that the orientation of the coordinate system, in which the form function is used, does not correspond to the previously introduced orientation. The formulas are taken from the source and used with the coordinate system there. Later, with respect to the orientation in the wave flume, the y-axis corresponds to the z-axis of the previously introduced coordinate system. This process is explained in more detail in the following section.

The model is realized by a 3D printer using the material ABS-P430 tm. For this purpose, a CAD model is created using Autodesk and an STL format is generated later. This STL file is then used by the printer's control system to generate the model. The subsequent printing process took around 22 hours. Due to production conditions, the model will later not consist entirely of material but will have hollow areas on the inside.
The basic design of the test object provides for two LMF100UU linear bearings at the top and bottom of the object. This is shown in figure 14, whereby the blue areas represent the linear bearings. The bearing on the underside is fixed in such a way that only the flange is visible on the outside. These reduce the disturbing influence on the shape of the object. On the upper side, on the contrary, this influence-reducing attachment cannot be applied, because in the upper area the spring must be fixed on the bearing.
Furthermore, it should be noted that two bearings are used to reduce possible tilting. However, this has the disadvantageous consequences that the influence of friction on the movement of the test object increases. Linear bearings are used for mounting on a shaft, only vertical movements are possible and should also only be possible. A rotation of the test object is undesirable despite rotation-symmetry, because of the application as PTO with a linear generator, only vertical movement

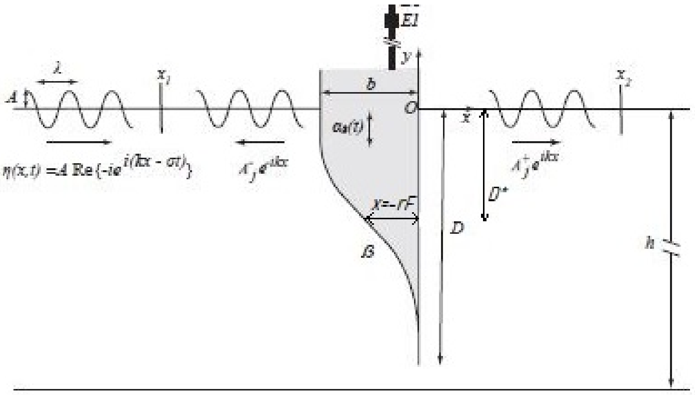


Figure 12: Asymmetric floating energy capturer. Based on [17].

can be changed into electrical energy.
Also the influence of the movement at the shaft should be kept as small as possible. In relation to the expected size of the test object, a shaft with a diameter of 10 mm is selected. This diameter is small compared to the final test object and has no significant influence on the experiments. The selection of shaft diameters therefore also justifies the selection of bearings.

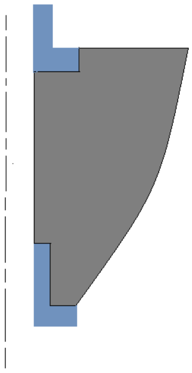


Figure 13: Schematic sketch of the test object cone.

Creation of CAD Model

To create the CAD model with the geometric data, the function mentioned in the previous section had to be incorporated first.
In the first step, the non-dimensional form function is related to the dimensions of the later model. The dimensions of the cone are limited by the maximum component size of the printer and the wave tank. The available 3D printer has a production space of 6x8x6 inches. This corresponds to about 150x200x150 mm. During the experiments with a water depth of 300 mm and various submergence steps, the cone should not be larger than 15 cm in height, taking the oscillation of the object into account. The width of the object is determined by the production dimensions of the printer, because the tank is considerably wider than the

production possibilities allow. The cone should be 15x15x15 cm. For y<0 the following equations (7) and (8) can be used. Where b and D represent the radius b of the cone and the height D as shown in figure 12.

$$\bar{x} = x/b \tag{7}$$

$$\bar{y} = y/D \tag{8}$$

However, it should be noted that D defines the height from x=b to x=0. The coordinate system is located in the rotation axis of the cone. If x is zero, a taper with a pointed tip is formed. However, as already described and shown in the sketch, a bearing should be fitted to the lower end of the cone and a shaft should pass through this bearing. The test object should therefore be a truncated cone. Therefore, in our application case, x will not be zero. The diameter of the flange of the bearing, which should rest on the lower end of the cone, is 40 mm. Therefore the lower diameter of the blunt cone should be about 41 mm and thus the radius 20.5 mm defined as $\mathrm{r_F}$. This defines the range of the form function taking into account the still valid assumption $\bar{y} < 0$ as follows:

$$[-b, -r_F) = [-b, -r_F[:= \{x \in \mathbb{R} \mid -b \leq x < -r_F\} \tag{9}$$

$$(r_F, b] =]r_F, b] := \{x \in \mathbb{R} \mid r_F < x \leq b\} \tag{10}$$

As already mentioned, D describes the height of the cone from x=b to x=0. Due to the changed interval, the cone will later have the height D* and not D. Therefore, |D*| = 23 cm and |b| = 7 cm were selected as untagging parameters. The dimensioned function is expressed by equation (11).

$$F(y) = x = \frac{b}{D} * [0.05926(y + \frac{1}{D})^2 + 3.88147(y + \frac{1}{D})^3 - 2.94074(y + \frac{1}{D})^4] \tag{11}$$

In the next step of creation, a table was designed. Starting from 0, the y value increases in 0.05 steps to y=23. With the defined increment for y and the height D the corresponding $\bar{y}$ values are calculated with formula (8) and inserted into the function (6). With the $\bar{x}$ values assigned from this and the previously defined radius b, the dimensioned x values are calculated from equation (7). With these y and x values a 6th-order approximated function (12) was designed.

The creation of the table has the background of a visual examination of the cone shape as well as modelling reasons. The form function must be integrated into CAD to generate the conical model. There are two ways to do this in the software which was used. On the one hand, functions can be modeled by coordinate points, on the other hand, curves can be specified as functions by F(x). Since the first option only displays the shape without including the shape curve in the creation of the model, the second option is used. However, the form function (6) which is used in literature is a function F(y). Instead of a complex mathematical calculation operation to adapt the function to the software requirements, the table values of the visual inspection are used to create an approximate curve. This approximation, equation 12, is a 6th-order function depending on x and the domain of equation (9). This method meets the accuracy requirements, since the manufacturing tolerances are greater than the deviations from 0.6mm resulting from the approximation.

$$F(x) = 0,0112x^6 + 0,2888x^5 + 3,0133x^4 + 16,274x^3 + 47,788x^2 + 70,088x + 27,427 \tag{12}$$

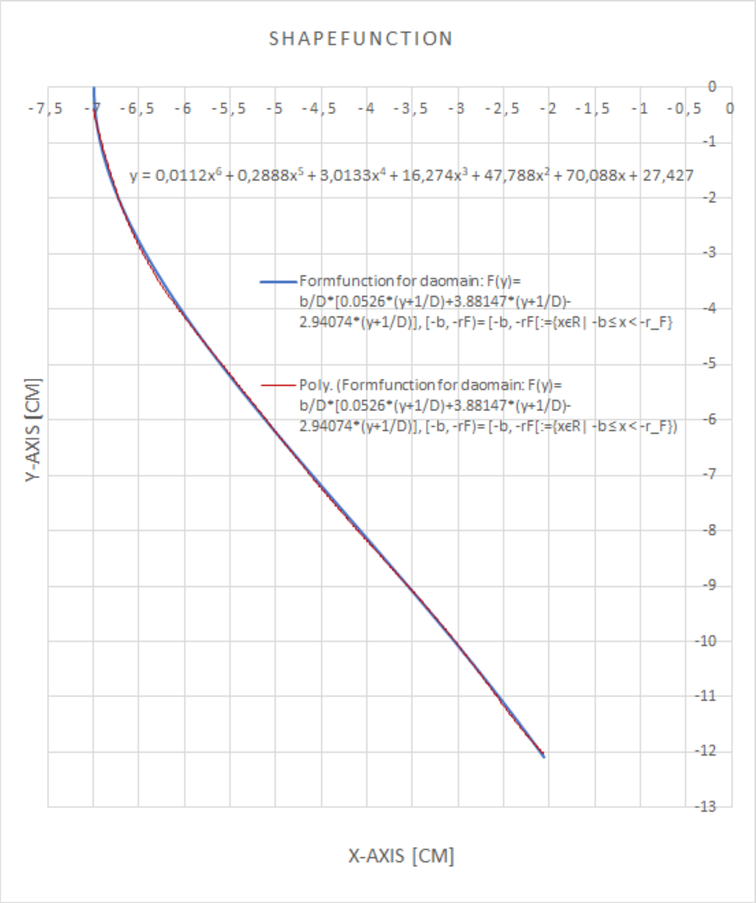


Figure 14: Graphical representation of the dimensioned form function and the approximation.

After modelling the outer shape of the cone, the wall thicknesses are modelled. Due to production reasons, the test object cannot consist entirely of material.
The wall thicknesses were chosen in such a way, that there is at each point a minimum material thickness of 11.4 mm. This thickness is sufficient to attach the bearings with the help of screws without drilling through the walls and getting

into the interior. Drilling through the hole poses a risk of water ingress.

Figure 15 and 16 show the dimensions and design of the cone.

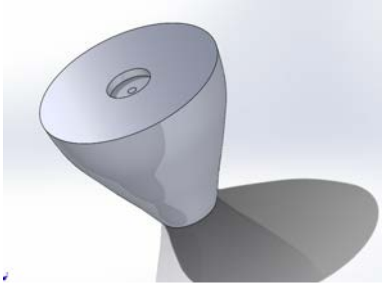


Figure 15: Representation of the cone design.

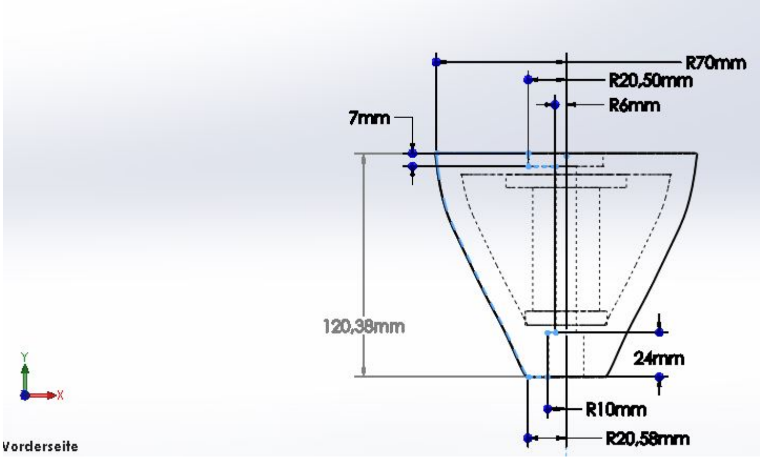


Figure 16: Sectional view of cone with dimensions.

After these steps, the created model is saved in a file format that can be used by the printer. The file information was then processed by the printer software and the model were created.

Model Preparation and Calculation of Hydrodynamic Properties

After the printing process of the cone model, the modeling of the test object is not yet complete. As explained for test object one this test object is also guided along a vertical fixed shaft with the aid of linear bearings. Due to the dimensions there are slight deviations from the previous test object. The greater height of the cone requires a second linear bearing, otherwise a tilting moment would occur. This would lead to additional, non-calculable friction. Due to the diameter of the cone, a smaller bearing is mounted.

Furthermore, it is important that the test object is submerged. It must be heavy enough for that. To determine the required weight, the following calculations are carried out. These calculations were made in the preliminary steps before printing, to ensure that the model is not too heavy and sinks. Such behaviour would prevent oscillation by the waves and is therefore an obstacle to the intended use.

Two forces normally act on a body in water, Fig. 17. The weight force, $\mathrm{F_g}$, that pulls the body down by gravity, and the buoyancy force, $\mathrm{F_A}$, created by the displacement of water, pushing the body upwards. If a body is floating in water, both attacking forces must be equal.

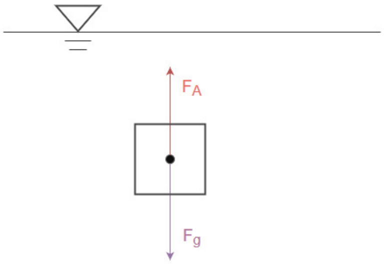


Figure 17: Attacking forces on a stationary body in water..

In our case, the body is a cone. To determine these acting forces, the density of the cone is calculated first, Eq. (13), to determine the weight later. An old test cone, called pre-cone, whose weight could be determined, was available for this purpose. The material volume of the cone was known in advance from the CAD data. All known datas and designations are listet in the table below. From equation (14) results the density for the test cone one.

Table 3: Formula symbols and values used to calculate the hydrodynamic properties of the cone.

Volume V
$\mathrm{V_{1m}}$ = 166.40 cm ^{3} = Materialvolume of pre-cone
$\mathrm{V_{2m}}$ = 601.77 cm ^{3} = Materialvolume of cone
$\mathrm{V_{2t}}$ = 1012.23 cm ^{3} = Totalvolume of cone
Weight m
$\mathrm{m_1}$= 171.18 g = Weight of pre-cone
$\mathrm{m_2}$= Weight of testobject cone
$\mathrm{m_3}$=Weight of linear bearing
$\mathrm{m_r}$= Real weight of cone with bearings and screws
$\mathrm{m_g}$=Required total weight
Density ρ
ρ_1= Material dense of pre-cone
ρ_2= Material dense of testobject cone
ρ_w= Density water

$$\rho = \frac{m}{V} \tag{13}$$

$$\rho_1 = \frac{m_1}{V_{1m=}} = \frac{0.1712kg}{166.4 * 10^{-6}m^3} = 1028\frac{kg}{m^3} \tag{14}$$

Since cone one and the test object cone two, consist of the same material, the condition (15) is assumed.

$$\rho_1 = \rho_2 \tag{15}$$

By converting equation (13) to equation (16) and inserting the values of cone two, the mass of the test object can now be calculated.

$$m = \rho * V \tag{16}$$

$$m_2 = \rho_1 * V_{2m} = 1028\frac{kg}{m^3} * 601.77 * 10^{-6}m^3 = 0.6186kg = 618.6g \tag{17}$$

As previously mentioned, the test object should not sink nor swim. In the ideal case, the body should float or sink very slightly. Thus the following force relation results from equation (18).

$$F_A = F_g \tag{18}$$

The weight force $\mathrm{F_G}$ results from the following components:

$$F_g = m * g \tag{19}$$

Because this equation takes the total mass of the test object into account, the weight of both bearings must be taken recognized in the calculation:

$$F_g = (m_2 + 2m_3) * g = (2 * 0.062kg + 0.6182kg) * 9.81\frac{m}{s^2} = 6.1N \tag{20}$$

The buoyancy force is described by equation 21. Decisive for the force is not, as in equation (14) and (17), the volume of the areas which are consisting of material, but the displacement volume of the cone. The Archimedean principle states that the buoyancy force of the test object in the medium is equal to the weight force of the medium, which is displaced by the object. The volume created by the shape of the object is displaced by the cone. Described here as $\mathrm{V_{2t}}$ and taken from the CAD data.

$$F_A = V * g * \rho_w \tag{21}$$

$$F_A = V_2t * g * \rho_w = 1.012 * 10^{-6}m^3 * 9.81\frac{m}{s^2} * 997\frac{kg}{m^3} = 9.9N \tag{22}$$

This proves, that the buoyancy force is significantly greater than the weight force. Therefore, additional weight must be applied. This is calculated in the following in which the weight force is required as in condition (18). This is set equal to the buoyancy force and equation (19) is replaced to equation (23). This results in the missing mass of 353g.

$$m_g = \frac{F_A}{g} \tag{23}$$

$$m_g = \frac{F_A}{g} = \frac{9.9N}{9.81\frac{m}{s^2}} = 1kg \tag{24}$$

$$m_a = m_g - m_r = 1000g - 647g = 353g \tag{25}$$

Trouble Shooting

After creating a first test object of the cone, another test object had to be created due to incorrect dimensions. The reason for this was an automatically activated option in the CAD program, which increases all dimensions by a factor. However, this was determined after the object was printed. This faulty cone is the pre-cone described above and the basis for the hydrodynamic calculation.
After the completion of a new test and the assembly of the other components, the first test runs were started. However, it turned out that the material which was used for the cone has an absorbing effect in the water. For this reason, the connection to the bearing was painted and sealed with silicone. This led to a slight change in surface and weight. Furthermore, the additional mass $\mathrm{m_a}$ was 357g. Therefore the weight of the cone was recorded again with $\mathrm{m_g}$= 1033g.
The measures applied also led to a positive change, but the design solution for the arrangement of the bearings is inadequate. Both bearings must be arranged very precisely on a straight line, which is difficult to achieve due to the type of assembly and the inaccuracy of the screw holes. This leads to considerable friction and distorts the measurement data.
Therefore the lower bearing was removed and the experiments were carried out with only one bearing. This causes additionally water absorption by separating the components, as well as tilting of the shaft in the bearing. The real mass used experimentally is therefore 971g. Since the duration of the experiments is short and the absorbed mass is small compared to the mass of the test object, the mass change is acceptable.

6. Springs and Bearings

In this section the friction, caused by the bearings, and the determination of the spring stiffness are explained in more detail. These are attached to the test object as already explained in the experimental assembly.

Bearings

The bearing friction is caused by horizontal forces acting in the direction of the shaft center of the vertical test arrangement. These forces are generated by the waves in the wave flume. If the forces, which are acting on the bearing in the horizontal direction, are known, the bearing friction can be calculated with the aid of the manufacturer's specifications.

This is illustrated in figure 18. The diagram shows a manufacturer-defined dependence between the friction constant μ and the force P acting "against shaft core"[4] direction. The forces causing friction in the bearing are rated at size C, dynamic basic load, with reference to the size and type of bearing, see appendix D.. These values are manufacturer's specifications and are C_1= 860 N for the bearing of the circular plate and C_2= 370 N for the bearing of the cone.[4]

In the first step of friction determination, the force which is induced by the waves in the x-direction is therefore calculated. According to the manufacturer, a change of the bearing friction with the force only occurs for a P/C ratio over 20%. Therefore an assumption is made. Due to the high basic load rating values, considerable forces must be generated by the waves in the wave flume. According to initial estimates, however, this is not the case. Therefore, the forces for the condition of a fixed test object and thus fixed bearings are calculated. The forces acting on this assumption are greater than on an oscillating object.
Equation (26) describes the horizontally acting force F_x, which acts on a fixed body with the length L_D at a wave height H and period T and immersion depth sd. All parameters used are dimensionless, marked by a crossbar.

$$\bar{F_x} = 3.60 * \bar{H}^2 * \bar{sd}^{0.11} * (1 - e^{0.09*\bar{T}})(1 - e^{-\bar{L}_D})[18] \tag{26}$$

For wave height and wave period the already dimensionless calculated values from section 4. are used. Since the largest effective force F_x is generated at the water surface by the greatest wave height and wave period, the values of H_1 and P_1 are used. Where index 1 corresponds to the force of the test object plate and index 2 corresponds to the cone.
Furthermore, this nomenclature refers the submergence depth sd in relation to the seefloor, i.e. our tank floor. This corresponds to the inertial submergence depth IS_1 from chapter 4., but IS is not standardized and must therefore still be related to the water depth, Eq. (27). Also the length L_D of the test objects must be standardized in relation to the water depth, Eqs. (28,29). Here L_{D1} represents length of the test object circular plate and L_{D2} represents length of the cone.

$$\bar{sd} = \frac{IS_1}{h} = \frac{0.27m}{0.3m} = 0.767[18] \tag{27}$$

$$\bar{L}_{D1} = \frac{L_{D1}}{h} = \frac{0.3m}{0.3m} = 1[18] \tag{28}$$

$$\bar{L}_{D2} = \frac{L_{D2}}{h} = \frac{0.14m}{0.3m} = 0.47[18] \tag{29}$$

After this normalisation, the forces, acting on the test objects, can be calculated and are then dimensioned to our experimental setup using equation (30) to (34). For this the length B_D of the test objects, perpendicular to the shaft direction, is required. Since both test objects are circular, this length corresponds to the length L_D. The height of the test object t_B corresponds to 1 t_{B1}=0.003m for the test object cone t_{B2}=0.124m.

$$F_x = \bar{F_x} * \rho * g * h * t_B * B_D[18] \tag{30}$$

$$\bar{F_{x,1}} = 3.60 * 0.2^2 * 0.767^{0.11} * (1 - e^{0.09*10})(1 - e^{-1}) = -0.1313 \tag{31}$$

$$F_{x,1} = \bar{F}_{x,1} * \rho * g * h * t_B * B_D = -0.1313 * 997 kg/m^3 * g * 0.3m * 0.003m * 0.3m = 0.35N \tag{32}$$

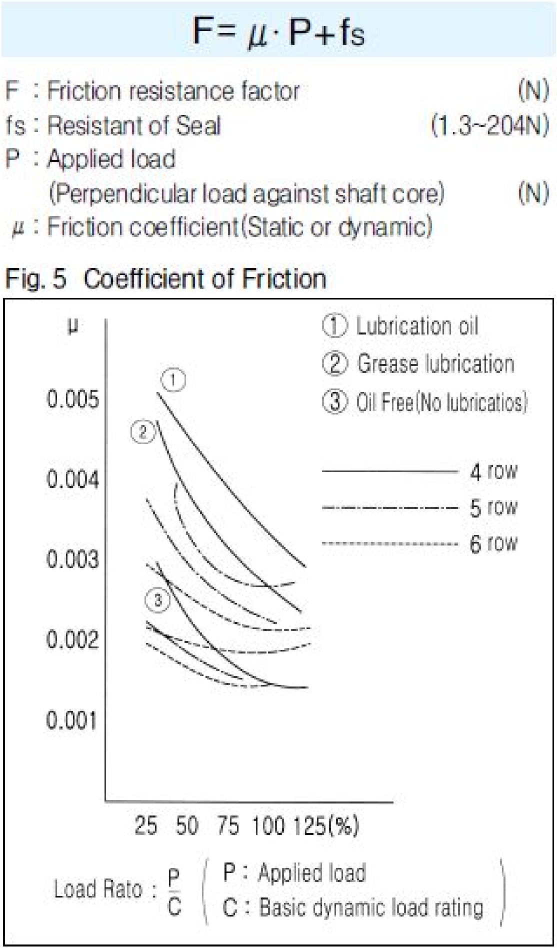
$$\bar{F_{x,2}} = 3.60 * 0.2^2 * 0.767^{0.11} * (1 - e^{0.09*10})(1 - e^{-0.47}) = -0.078 \tag{33}$$

$$F_{x,2} = \bar{F}_{x,2} * \rho * g * h * t_B * B_D = -0.78 * 997 kg/m^3 * g * 0.3m * 0.124m * 0.3m = 8.5N \tag{34}$$

After calculating the forces, these are set in relation to the dynamic basic load ratings. The ratio calculated from this, equation (35) and (36), is used to determine the coefficient of friction for the two different bearing types from equation (32) and (34). From the Graph, Fig. 18, the function two is used to taking lubrication losses caused by the release of grease by water into account. This results in a bearing friction of μ=0.0048 for both test objects.

$$\frac{F_{x,1}}{C_1} = \frac{0.33N}{860N} = 0.41\% \tag{35}$$

$$\frac{F_{x,2}}{C_2} = \frac{8.5N}{370N} = 2.3\% \tag{36}$$



$$F = \mu \cdot P + f_S$$

F : Friction resistance factor (N)
fs : Resistant of Seal (1.3~204N)
P : Applied load
(Perpendicular load against shaft core) (N)
μ : Friction coefficient(Static or dynamic)

Fig. 5 Coefficient of Friction

Figure 18: Friction coefficient diagram [4].

Springs

In order to obtain meaningful test results, knowledge of the spring stiffness is required. This has already been determined due to the use of test accessories for various experimental work. However, due to the importance of these constants for the representativeness of the results presented here, the procedure for determining them is briefly discussed.

In order to generally determine the spring constant of a spring, length measurements of the spring are carried out with different weights. Since there is a linear relationship between the change in length and the attached mass, the spring stiffness k can be expressed by the slope of this straight. The more measurements and repetitions are performed, the more accurate the results. Particular attention should be paid to a fixed reading point to record the change in length. Furthermore, the expansion of the spring must not be impaired by e.g. friction of a wall. Therefore, the measuring weights must not be attached to the spring coils. The ruler used must also be mounted parallel to the spring, otherwise the length will be distorted.
To define the spring constants used here, 13 measurements were carried out with

different weights and no repetitions. The spring constant has been determined by the gradient of a degree of compensation through the measuring points. The reading accuracy is +/-0.5mm. [19]

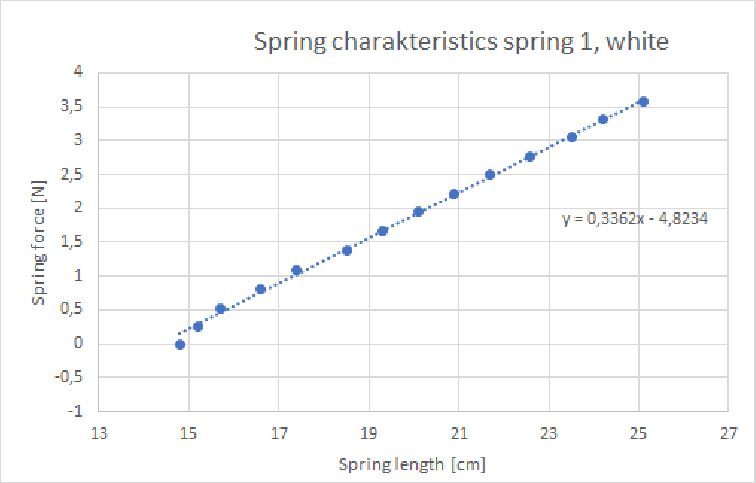


Figure 19: Spring characteristics spring 1, white[19].

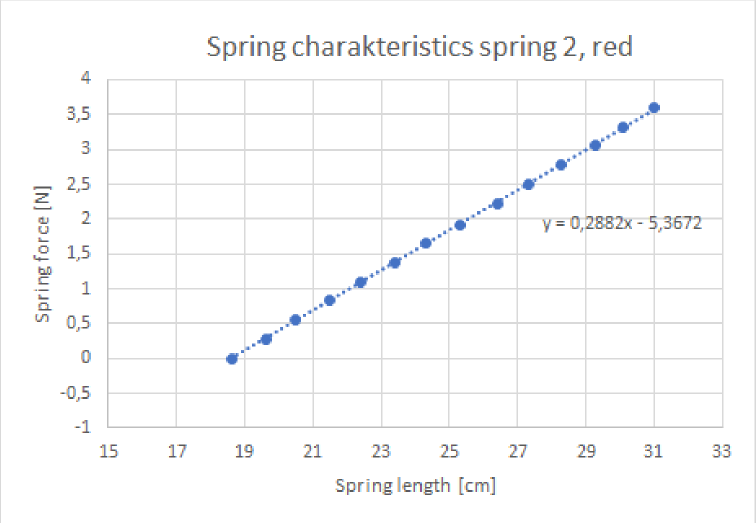


Figure 20: Spring characteristics spring 2, red[19].

After the experiment has been carried out, the weight force of the mass can be plotted against the change in length. This is shown in figures 19 and 20. The gradient of the spring characteristic determines the spring stiffness. This is for spring 1 k_1= 35.7 N/m and for spring 2 k_2= 28.7 N/m for spring.

7. Measuring Equipment

In this section, the used measuring systems are explained in more detail. During the experiments, the generated waves are recorded as well as the oscillation of the test objects. Both happend in different ways. While the test objects are detected by video analysis, the generated waves are detected by capacitive sensors.

Wave Gauges

Three measuring points of the type Wave Staff XB from Ocean Sensor are used to collect the wave data.
The measuring system is based on the principle of capacity change with change of water level. For this purpose, the system consists of two separate electrodes. The Teflon insulation of the rod serves as dielectric. While one electrode is in the water, the second electrode is in the sensor shaft as a shield conductor. If the water depth changes, the capacity changes linearly. The capacity is recorded as a function of time and can therefore be assigned to a voltage value.

The calibration is carried out by a maximum of 12 measuring points, which should be in the range of 20-80% of the expected measuring range of the water depth. Each calibration point has a tolerance of 0.05%. During calibration, a voltage value is assigned to each measuring point at a known water depth. This creates a calibration curve that is stored and called up in the system.

Depending on the length of the Shaft, here 0.5 m, water depths that do not exceed this value can be measured.

Measuring Programs

One goal of the experiments is to capture and graphically display the motion behaviour of the test objects. The position of the object should therefore be recorded over time. In order to realize this technically, the tests were recorded by means of a video camera. These videos can then be analysed using Matlab. For this purpose, two different Matlab programs were implemented.

These two programs use different ways to detect the test objects. The first presented program is based on a color detection of the object, the second program on a detection of the movement of the object. However, after the test object has been detected, both programs use the same code.
Depending on the enviromental conditions of the recordings, a different program may be advantageous. If the environmental conditions have many colors, which are very similar to the test object ist would be difficult to seperate the test object. On the other hand, if more than the test object is moving in the video, the object should be detected by color seperation.

Motion Tracker 2 was used to analyze the data presented here. This has the background that the wave movement can be seen in the videos.

Foreground Detector

The foreground detector program is based on the recognition of the affiliation of the foreground or background by means of video analysis.

In the first step, the memory location and the name of the video are recorded. The video information is then read by Matlab and information regarding duration and frames per seconds is captured in variables. This information is used to calculate the number of images and the exact time of each image.
In the next step, two matrices are created. These consist of two columns and rows according to the number of images that make up the video. Initially, the matrix consists of zeros, which are replaced later.

The essence of the program is represented by the following one-line section. With the Matlab command "forgroundDetector" several objects can be detected. This command is based on the Gaussian Mixture Model (GMM). [20]
These are probability-density-functions, which are describing the background in a GMM.[21]
The parameters of these functions are created by a number of learning images. In this program five images are used for learning to create GMMs. During the learning process, images are compared for changes. Additonaly there is a learning rate which specify how quickly the foreground is changing and the parameters change according to this. Fast objects needing a high learning rate.
In general, if the pixels do not change, they belong to the background and are described by the probability-density-functions. If individual pixels do not fit with this probability-density-function are described, they belong to the foreground. The number of GMMs in this program is four.[21]

After the command "foregroundDetector" has been defined, an order to save the image sequence from the video is created. For this purpose, the existence of an order with the name "ImageSequence" is queried and generated in case of non-existence.
The video is read and in the following loop each image is recorded step by step with the "foregroundDetector". The loop has as many repetition as the video has frames. After five loops, the number of training frames required is reached and the object is now detected. The background is represented as black and the foreground, i.e. the object, is represented by white pixels. However, in order to increase the accuracy, the images are filtered after detection. This minimizes the influence of reflections and other disturbing factors.[20]
With this complete representation of the detection, connected pixels are searched for using "vision.BlobAnalysis". If these contiguous areas consist of more than 300 pixels, the area is taken into account. The center of these areas is searched for as well as the coordinates of a the upper left edge of a rectangular box enclosing these areas.[22]
The detected areas are sorted in descending order of size. Since the objects to be detected are very large, it can be assumed that the largest found blob-area is the object. The center of the largest blob found is stored in the previously created "Cent" matrix.

In the next step, a colored box is inserted around the detected object in the original image. The previously known position of the box from the vision.blobAnalysis command is used. The edited image is saved and then used to analyze the position of a chessboard, which is visible in all videos. The chessboard is used to calibrate the camera position.

Already 35 pictures from different camera positions have been taken with the chessboard in the designated position for the experiments. These images were analyzed using the Matlap "CameraCalibrator" app and the results were stored in the CameraParams variable and the properties of the calibration panel. To perform this analysis, the size of the chess pattern must be known. It analyzes the camera distortion and camera properties that describe how the camera creates a 2D image from a three dimensional environment. This is important in order to later deduce the 3D reality from the images and thus to calculate the actual position of the test objects in the room. This is not represented realistically by changing the dimension when it is transformed into a photo.[23]
After the analysis is completed, 3D world points can be determined from 2D images using the calculated camera parameters. First, the size of the chessboard and the corner points of all fields in the chessboard are determined with the command "detectCheckerboardPoints". It is important that the chessboard has a different number of rows and columns to be able to assign whether it is a vertical or horizontal movement. In this step the chess board in the image is simultaneously compared with the chess board from the calibration. It must be the same or the program will not continue.
In the following command 'extrinics' two correction matrices are created. A 3D rotation matrix and a 3D translation vector. The purpose of these matrices is to write points from the two-dimensional camera coordinate system ("ImagePoints") into the world coordinate system. The detected points of the chessboard in the image and the world points as well as the lens distortion from the camera parameters are used for this. These correction matrices are then used to correct the stored coordinates of the objects in the "Cent" matrix and save them in "WorldCent", thus reproducing the realistic object position.[24]

Afterwards the revised images are written back into a video and saved.
Finally, the y-cordiants of the blob from the matrix "WorldCent" are written into a matrix name "position" using matrix calculations together with the time of the respective frames as x values. The reason for this, is that the matrix "WolrdCent" stores the center of the blobs. However, since there is only a vertical movement of the object, i.e. in the y-direction, the x-cordiants should not change and are of no importance. In order to be able to trace the time-dependent vertical movement of the object later and to be able to use it in other programs, it is useful to store both information, time and vertical position, in a matrix. In this arithmetic operation, the actual position of the coordinate origin in the chessboard, top left, in relation to the bottom of the shaft tank in the variable E can be taken into account, if desired. The coordinate origin is defined by the Calibration App and displayed to the user during calibration. The unit of the value for E depends on the unit of the chessboard in the Calibration App.

Finally, the course of the detected object is displayed graphically.

Motion Tracker 2

Due to the defined test conditions of the objects close to the water surface and the difficulties of the Foregound Detector program to detect only the movement of the test object, a second variant was created. This program is also based on the Matlab program "Motion Tracker", but this time it is the basic principle of detection by color seperation. The program "SimpleColorDetectionByHue" in the appendix is used as a further source.

As already explained in the previous section, this program first asks for the location, name of the video and the video information, as well as the creation of the matrices for position storage and after a negative query, the folder for saving the images.

The video is then saved in its image sequence. As this image sequence is changed in the further process to a black and white image, in which the chessboard is no longer visible for correction of the camera position, the video is read again and the corresponding image sequence is called up step by step in the following for-loop. Here, too, the loop is run through as often as the video has images.

The previously saved images are called up individually as they pass through the loop and read in with the "imread" command. At the same time, the command identifies the color of each pixel and assigns values to each color and saves them. These are described in the matrix rgbImage. Some images have a color map (storedColorMap), in which a value for the color is entered for each pixel. Each pixel in the color map is described by a numeric value.
In the next step, the size, i.e. row, columns and number of color bands of the previously created matrix are recorded. This is used to later recognize the image type of the photo, as the type of photo determines the way the values are assigned to describe a color. If one or more values are to be used to draw conclusions about the color, it must be determined which image type and therefore which color coding is concerned, as well as which image class is concerned. Depending on the class of the image (uint8, uint16, single, or double), the color values are in a range of values between 0 and 1 (double and single), 0 and 255 (uint8) and 0 to 65535 (uint16).[25]
Uint16 has such high quality for images, that it is not used for normal pictures and this program.
In general, there are three different types to describe an image by values. True color Image, indexed image and a grayscale image. A true color images discribes each color with three values. These values are descirbing the terms of the colour amount of red, green, and blue which are repesented. The second type is a grayscale image, here the color is only described by one numerical value, since this is sufficient to clearly identify and delimit the color. Therefore, this type has only one color band. This knowledge is later used to identify the image type. In the third and last type, the indexed image, the color is the only one which is described by a vector and a color map. Also in this case, the number of color bands is equal to one. As for the application of the program, the vector representation

is required by three numerical values of the true colour image, the other cases are checked in the following.[25]

Based on this background, the system first checks which image class is involved. If it is an 8 bit image, the value for uint8 is set to 1. The image type is then checked for indexed and geayscale. If the number of colour bands is equal to one, it must be one of these two types. If, in addition to this criterium, a color map exists, it is an indexed image, otherwise it is a greyscale image.
If the conditions of a grayscale image are fulfilled, in the next step the descriptive color values are converted into the format of a truecolor image with the command "cat". So instead of one value for a color into three descriptive values for a color. This command does nothing else than copy the one descriptive color value for each pixel, that at the end three descriptive numerical values with the valence of the color value in greyscale format exists. These values then describe exactly the same color in truecolor format.
If an indexed image is identified instead of a greyscale image, then this image type is also converted to truecolor image. This is done with the command "ind2rgb". However, this command only covers the indexed image to a truecolor image with the image class double or single. Unlike greyscale images, where the values are simply copied and the image class is not changed, the image class can be changed while converting an indexed uint8 image. This is the reason why it was checked before whether it is an 8 bit image. If this criterion is met, all color values are multiplied by 256 in the next step and thus returned to uint8.

If it can now be assumed that it is a truecolor image or also called rgb image, the next step is to convert the color values back to HSV format. HSV describes colors using three values for hue, saturation and value. HSV is always preferred when a colour description is required. The reason for this is that the HSV model is more similar to the human visual perception and therefore easier to define and select colour value ranges. Since the user should later be able to adjust these ranges for different recording conditions, a color description is necessary. The conversion is initiated by the command "rgb2hsv".[26]
In the following three lines of code, the HSV values are written into the corresponding matrix (hImage, sImage and vImage) according to hue, saturation and value. The user can then specify his own values for these three color characteristics. They are describing the colors that the program is looking for and detecting. In the next step, all values in the three matrices, which are in the desired color value range, are transferred to three new matrices (hueMask, saturationMask and valueMask). If values are not in the value range, they are stored as zeros in the new matrices.
Now the three matrices hueMask, saturationMask and valueMask are combined again as colorObjekctMask. If all three HSV properties have a value for one pixel, they describe the color we are looking for. If only one of these three values is zero, the pixel is not included in the detection.

In order to minimize the influence of disturbing factors, all pixel areas that fulfill the color values but do not have more than 100 neighboring such permitted pixels, they are eliminated in the following two steps. Furthermore, the edges are smoothed and individual color values for pixels that could belong to the object are included.

The following steps are the same as in ForegroundDetector and are therefore not explained here. With the help of the command blobAnalysis the contiguous pixel area is captured by coordinates and provided with a box. Then the areas are sorted by size and the largest detected area is used for further steps. In the original image a box is then inserted around the pixel area and the coordinates of the detected area are corrected with the camera parameters. The image sequence is merged into a video, the time and the y-coordinates are stored in a variable and plotted.

8. Error Analysis

To determine the analysis accuracy of the object oscillation, the wave conditions were repeated three times for test object one with spring one and for all submergence depth. It is assumed that the object oscillation must be the same due to the unchanged parameters for the repetitions.
Therefore, the oscillation height of the test object $\mathrm{H_{oz}}$ for all three repetitions has been determined.The deviation of the oscillation height of the repetitions is caculated, Eq.(37), by further analysis of the largest and smallest oscillation height $\mathrm{H_{oz,max}}$ and $\mathrm{H_{oz,min}}$,as well as the avverage heights $\mathrm{H_{oz,Averrage}}$. This is referred here as oscillations height deviation $\mathrm{H_{oz,error}}$.

$$H_{oz,error} = \frac{H_{oz,max} - H_{oz,min}}{H_{oz,Averrage}} * 100\% \tag{37}$$

As visible in table 7, the error range is very large. On the whole, no correlations of the error $\mathrm{H_{oz,error}}$ between the individual submergence depths can be observed. About 90% of the oscillation height errors are in the range of 4-20%. However, per submergence depth are about 2-4 values that exceed this range. In very few cases, the oscillation height error is twice the mean oscillation height.

Due to the large scattering range of the deviations, the empirical standard deviations for test object one, spring one, submergence step one are also calculated using equation (38). The index n, used there, denotes the wave height H of 1-6 from section 4.. The results are shown in Table 4. This means that the average oscillation heights vary between 0.1 and 6.5 mm.

$$\tilde{s} = \sqrt{\frac{\sum_{i=1}^{3}(H_{n,oz,i} - H_{n,oz,Averrage})^2}{3}} \tag{38}$$

Table 4: Erroranalysis, standart deviation of circular plate with spring one.

Standart deviation of the oscillation height $\mathrm{H_{oz,sd}}$ [mm],, spring 1, IS = 0.27 m					
T [s] / **H [mm]**	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
10	0.648	2.772	1.603	–	–
20	0.141	1.667	0.869	2.386	2.108
30	0.450	2.536	2.941	2.863	1.597
40	2.176	6.511	1.910	3.285	3.844
50	2.532	6.341	3.581	5.043	5.453
60	1.715	6.533	2.331	0.805	4.705

In general, as the wave height increases, the distances between the repetitions of the measured oscillation heights increase, figure 21, but there is no influence of the period on this behavior, figure 22.

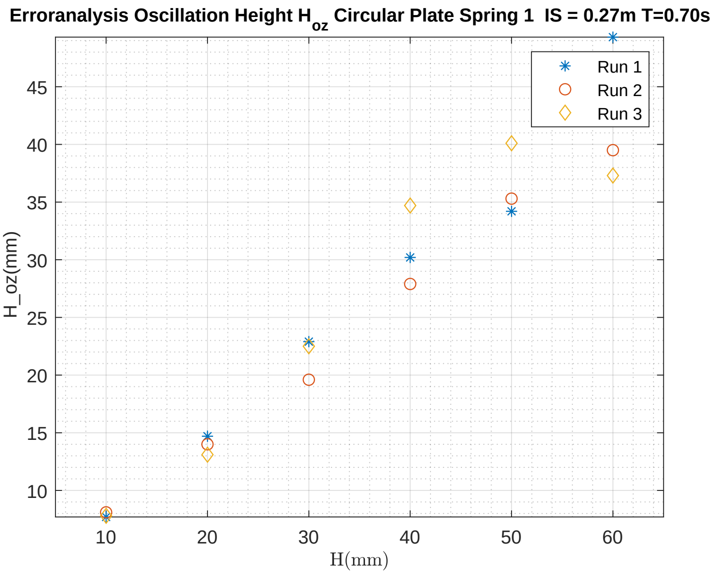


Figure 21: Erroranalysis oscillation heights of the different runs as a function of the wave height.

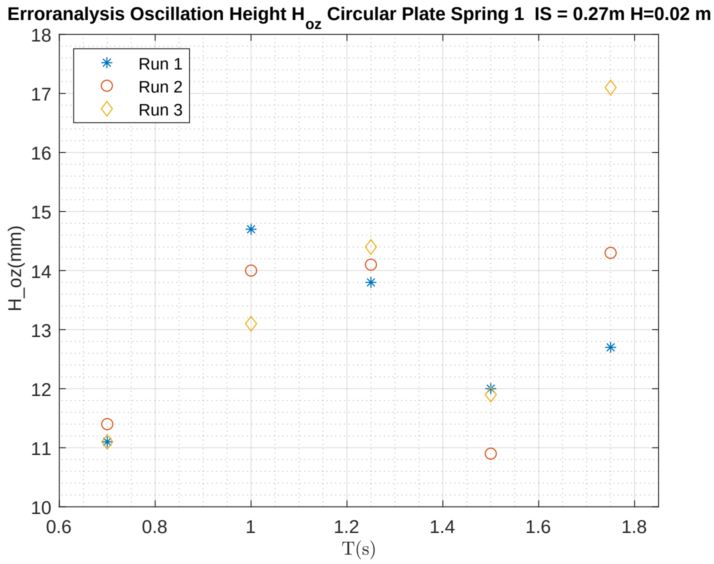


Figure 22: Erroranalysis oscillation heights of the different runs as a function of the wave period.

9. Results and Discussion

Based on the error analysis, the test data are analysed and interpreted in more detail. First, the individual test objects are analysed and compared later on. Due to the transferability to other test environments, values normalized to the water depth are also used here.

Analysis Testobject Circular Plate

The experiments with test shape circular plate have shown the following behaviour for both spring variants:
With rising submergence depth a decrease of the oscillation height is to be observed, whereby the values for s_4 deviate more clearly than the remaining submergence depth among themselves. Within a submergence depth the oscillation height increases with increasing wave height. In general, an increase in object movement can also be observed with increasing period, although some measured values deviate from this behaviour. The measured values of the period T= 1.25 s are particularly frequently affected. However, the measured values are often very close to each other, so that an influence of a measurement error cannot be excluded. But this could be an other unknown cause too. This behavior is shown in the figures 24 to 23.

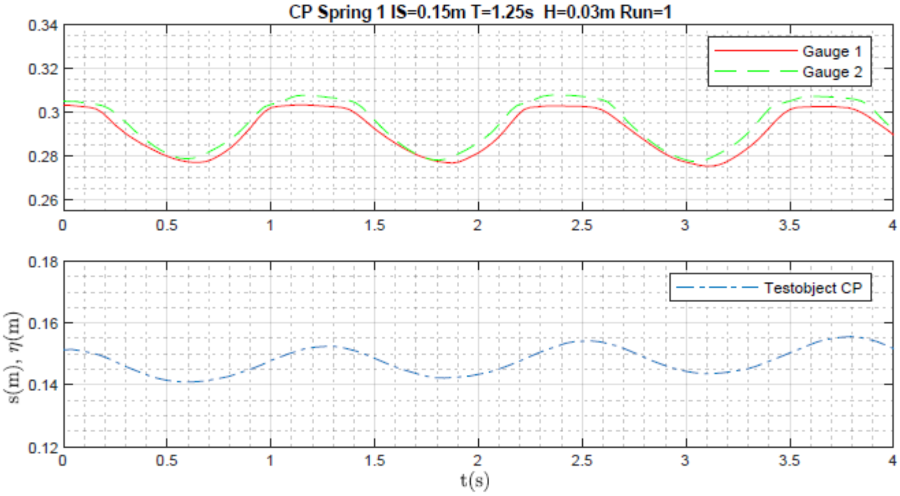


Figure 23: Illustration of test object plate spring one IS=0.27 P=1.75.

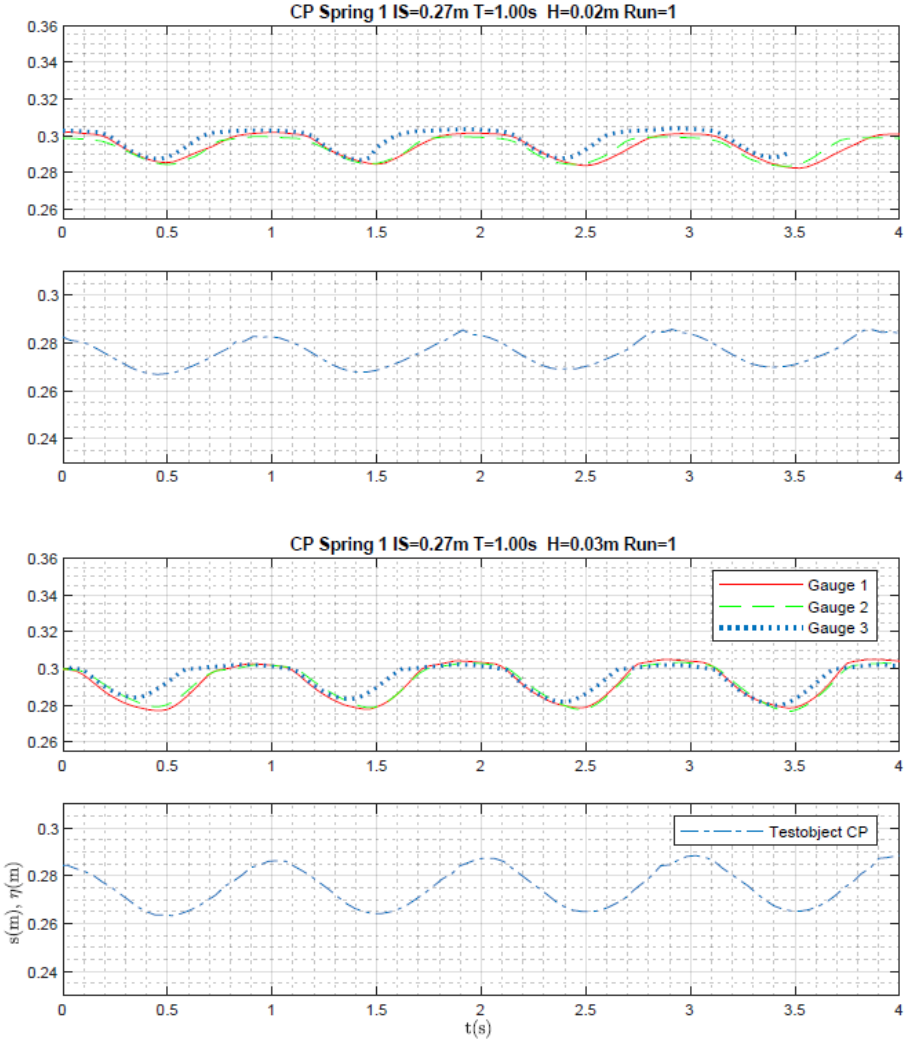


Figure 24: Illustration of test object plate spring one IS=0.27 P=1.00.

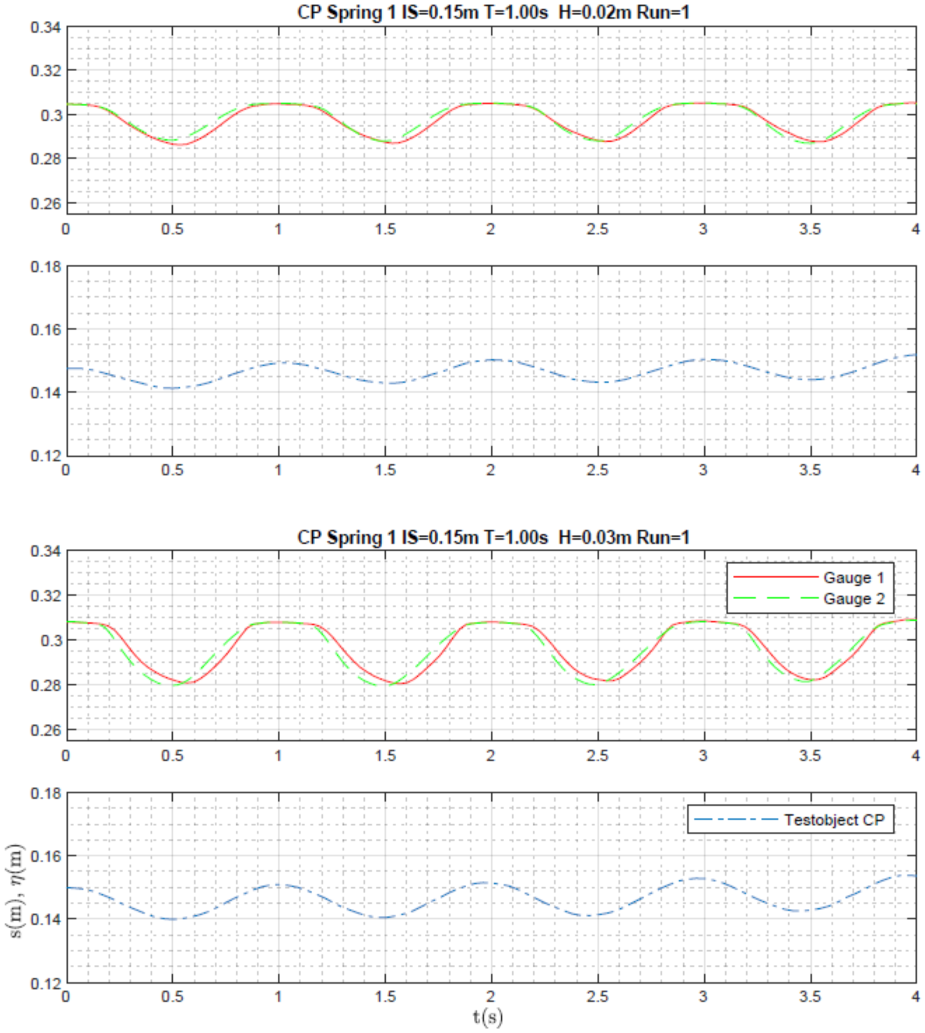


Figure 25: Illustration of the test object plate spring one IS=0.15 P=1.00.

Another peculiarity is the increasing mean value of the oscillation. This can be observed for almost all graphs, with the exception of spring one and submergence depth four. The graph drops off there. This effect is particularly visible with increasing depth of immersion and thus smaller object oscillation. This effect could not be confirmed by visual inspection of the videos. However, the change is minimal and does not need to be detected by the human eye. Therefore, in the next step, the video analysis and the programs were checked. On the one hand, already analyzed videos were checked with the identification of the detected object. The detected area is indicated by a box whose center coordinates are the basis for the graphs. If a wandering of the graph is detected, the box should also wander or change its height. This was not the case. The same behavior was found when the videos were analyzed again.

Therefore, in the next step, a video with a fixed test object was analyzed. If the analysis programs accumulate values, the graph should rise. This was not the case too. Although a jump from picture one to picture two of the video was visible in the graph, the rest of the graph was horizontal. In addition, this value was significantly higher than the increase or decrease in the graphs.

Furthermore, no constant summation within the graphs and among each other could be determined.

Another point of the oscillation behavior is that the movement behavior of the test object is gentler than the wave. This means that with a rash of the wave, the object follows this movement in attenuated and smooth.

If one compares the behaviour of the test object with different springs, similar oscillation heights are obtained:

- Spring one:
  - s_1: $H_{oz}/h=$ 0.015 - 0.15
  - s_2: $H_{oz}/h=$ 0.014 - 0.13
  - s_3: $H_{oz}/h=$ 0.007 - 0.073
  - s_4: $H_{oz}/h=$ 0.004 - 0.042
- Spring two:
  - s_1: $H_{oz}/h=$ 0.02 - 0.15
  - s_2: $H_{oz}/h=$ 0.012 - 0.098
  - s_3: $H_{oz}/h=$ 0.006 - 0.077
  - s_4: $H_{oz}/h=$ $23*10\hat{-}5$ - $41*10^{-4}$

However, if all value are considerd and not only the minimum and maximum values of the individual submergence depth, the oscillation height, with the stronger spring one, is generally higher than with spring two. You can see this behavior in comparison to Figure 26. Because the spring constants of both springs are close to each other and thus the measured values are also close to each other. Deviations of the addressed behaviour could be related to measurement errors.

Another observation is a slight variation of the average value of the object oscillation within a submergence depth by a maximum of 1 cm. This can be attributed to the large water loss of the wave flume during the experiments.

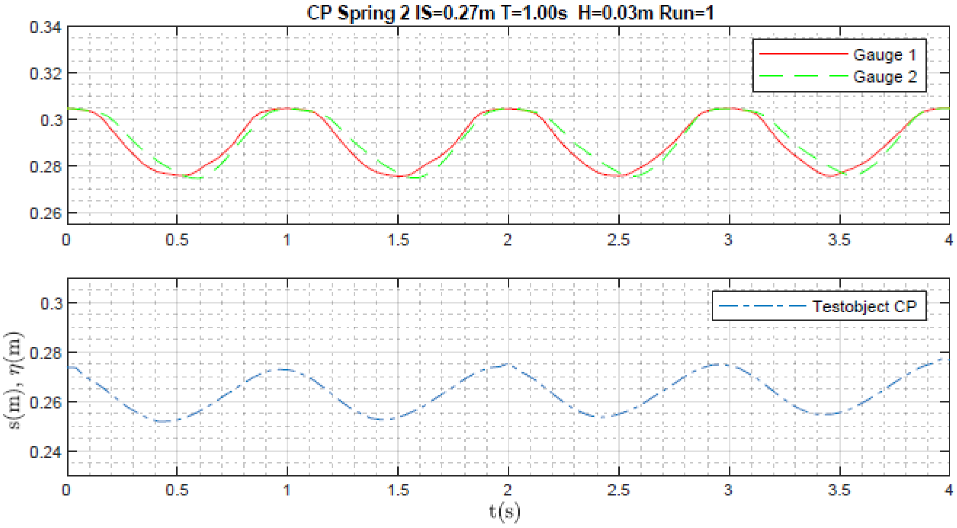


Figure 26: Illustration of test object plate spring two.

Table 5: Analysis oszillation Height Object one.

Ratio of wave height to oscillation height, H_{oz}/h of the test object 1 spring 1 IS = 0.27 m					
T [s] / **H [mm]**	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
10	0.016	0.027	0.015	–	–
20	0.037	0.049	0.047	0.040	0.057
30	0.060	0.076	0.076	0.084	0.063
40	0.067	0.093	0.100	0.102	0.097
50	0.102	0.118	0.115	0.143	0.111
60	0.109	0.124	0.122	0.143	0.154

Ratio of wave height to oscillation height, H_{oz}/h of the test object 1 spring 1 IS = 0.21 m					
T [s] / **H [mm]**	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
10	0.014	0.021	0.014	–	–
20	0.025	0.038	0.038	0.033	0.032
30	0.037	0.055	0.055	0.060	0.056
40	0.049	0.079	0.068	0.072	0.074
50	0.067	0.090	0.075	0.088	0.079
60	0.083	0.111	0.084	0.091	0.130

Ratio of wave height to oscillation height, H_{oz}/h
of the test object 1 spring 1 IS = 0.15 m

T [s] H [mm]	0.70	1.00	1.25	1.50	1.75
10	0.007	0.013	0.010	–	–
20	0.012	0.022	0.024	0.021	0.019
30	0.019	0.035	0.033	0.041	0.037
40	0.029	0.045	0.042	0.057	0.048
50	0.039	0.055	0.049	0.059	0.068
60	0.040	0.064	0.058	0.064	0.073

Ratio of wave height to oscillation height, H_{oz}/h
of the test object 1 spring 1 IS = 0.09 m

T [s] H [mm]	0.70	1.00	1.25	1.50	1.75
10	0.004	0.007	0.007	–	–
20	0.008	0.016	0.017	0.017	0.020
30	0.010	0.021	0.021	0.028	0.020
40	0.014	0.026	0.026	0.035	0.031
50	0.019	0.033	0.037	0.042	0.037
60	0.023	0.039	0.038	0.042	0.041

Ratio of wave height to oscillation height, H_{oz}/h
of the test object 1 spring 2 IS = 0.27 m

T [s] H [mm]	0.70	1.00	1.25	1.50	1.75
10	0.020	0.029	0.018	–	–
20	0.034	0.047	0.047	0.044	0.046
30	0.059	0.067	0.068	0.088	0.073
40	0.082	0.088	0.090	0.107	0.091
50	0.081	0.103	0.112	0.125	0.124
60	0.103	0.123	0.127	0.153	0.122

Ratio of wave height to oscillation height, $\mathbf{H_{oz}/h}$ of the test object 1 spring 2 IS = 0.21 m					
H [mm] \ T [s]	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
10	0.012	0.019	0.013	–	–
20	0.018	0.035	0.044	0.034	0.035
30	0.031	0.055	0.054	0.056	0.061
40	0.049	0.067	0.071	0.092	0.069
50	0.054	0.086	0.070	0.091	0.086
60	0.070	0.098	0.079	0.094	0.090

Ratio of wave height to oscillation height, $\mathbf{H_{oz}/h}$ of the test object 1 spring 2 IS = 0.15 m					
H [mm] \ T [s]	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
10	0.006	0.014	0.006	–	–
20	0.012	0.023	0.026	0.021	0.029
30	0.019	0.034	0.034	0.036	0.042
40	0.025	0.045	0.046	0.060	0.057
50	0.030	0.055	0.055	0.061	0.067
60	0.036	0.069	0.063	0.072	0.077

Ratio of wave height to oscillation height, $\mathbf{H_{oz}/h}$ of the test object 1 spring 2 IS = 0.09 m					
H [mm] \ T [s]	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
10	0.002	0.007	0.004	–	–
20	0.005	0.015	0.014	0.016	0.014
30	0.009	0.019	0.017	0.023	0.023
40	0.013	0.027	0.024	0.030	0.028
50	0.017	0.031	0.029	0.034	0.035
60	0.020	0.038	0.033	0.038	0.041

Analysis Testobject Cone

Unlike test object one, test object two was only tested with spring two and wave height H_6. Here, too, a decline in the association of objects can be observed with increasing submergence depth. How the oscillation behavior of a continuously submerged test object behaves with increasing submergence depth must be investigated by further tests.

An increase in oscillazion with the period can only be observed for s_0, however, when viewing the videos a stuttering oscillation can be observed, which is obviously caused by friction of the tilting bearing. Due to the limitted test of the object, these tests could not be repeated. It is to be assumed that these are also the cause for the strongly falling oscillation heights and thus the statement of the decreasing oscillation should be checked again. Based on primary observations of the behaviour of the test object, increasing oscillation heights are also expected for rising wave heights.

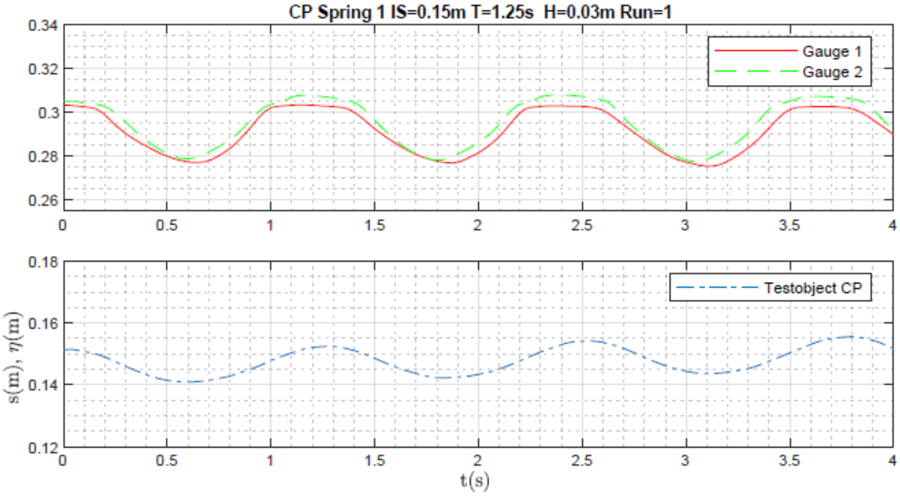


Figure 27: Illustration of the test object cone spring two.

During the experiments, weight was increased by water adsorption. The weight increase in the timeframe of the periods P_1 to P_5 from s_0 was 24 g. And in the same period from s_1 34 g.

- Spring two:
  - s_0: H_{oz}/h= 0.016 - 0.059
  - s_1: H_{oz}/h= 0.006 - 0.067

Table 6: Analysis oszillation Height Object two.

Ratio of wave height to oscillation height, H_{oz}/h of the test object 2 spring 2 IS = 0.30 m					
P [s] / **H [mm]**	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
60	.016	.057	.059	.055	.041

Ratio of wave height to oscillation height, H_{oz}/h of the test object 2 spring 2 IS = 0.27 m					
P [s] / **H [mm]**	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
60	.030	.067	.067	.018	.006

Comparison of the Test Shapes

Overall, for both test objects show the same behaviors for the same parameters. The oscillation of the cone is significantly smaller than of the circular plate. However, the surface of the two objects must be taken into account. The 30 cm diameter of the plate is more than twice as large as the 14 cm diameter of the cone. Also, it appears that the oscillation of the cone decreases more with increasing submergence depth.

Furthermore, a tilting of cone by the waves was observed in the videos and during the experiments. Thus, the friction of the object is considerably greater and considerably impairs the movement behaviour. Due to the few test data of test object two, it is difficult to make further statements because of the changing mass and friction.

However, for both objects no significant influence on the course of the waves can be observed. In addition, it must be noted at this point that from test object one, spring one, IS=0.21 m T=1.25 H=0.04 m Run 1 measuring point three has sent false signals for all subsequent tests which cannot correspond to reality. The sequence of the tests carried out corresponds to that in the appendix. This behavior can also be observed temporarily from test object one, spring one, IS = 0.21 m T=1.25 H=0.04 m Run 1 to test object one, spring one, IS=0.21 m T=1.75 H=0.06 m Run 3 for measuring point two. For the tests from test object one, spring one, IS=0.15 m T=0.70 H=0.01 m Run 1 a new calibration followed.

10. Conclusion

Due to the increasing energy consumption, the decreasing fossil fuel resources and climate change, renewable energies are becoming increasingly important. Wave energy in particular represents one of the largest and at the same time most unutilized resource and therefore offers a lot of potential. Many projects to generate wave energy are still in the R&D-stage. This thesis is intended to make a contribution to this.

The aim of this bachelor thesis was to determine the oscillation behavior of various forms of submerged primer movers, which are used as a component in heaving WEC. This was done experimentally in a wave flume and under the influence of shallow water and various submergence depth, wave period, wave height and spring stiffness. Test environment was a piston-wave-maker. The test setup was designed in such a way that the objects were mounted on a shaft with a linear bearing and a spring. This shaft was fixed in vertical position in the wave flume. Two measuring points for measuring the waves were located upwave and one downwave. Two test objects were used. A circular 3 mm thick plate and a cone. The shape of the cone is based on scientific research on asymmetric, heaving devices on the water surface.

Wireless capacitive measuring systems were used as measuring instruments to determine the water depth as a function of time. Thus, the wave movements can be captured. The motion detection of the test objects was based on video recordings which were analyzed by means of two programs. These programs use the Matlab interface and are based on color seperation and recognition of the background by mathematical Gaussian Mixture Models. Mainly the software, which is based on color seperation was used.

The error range determined during the analysis of the experiments is very large. About 90% of the test values show an error of 4 - 20%. Related to the test, this is a variation of the object oscillation of 0.1 - 6.5 mm with wave height of 1 to 6 cm. The deviations increase with increasing wave height as well as to a large extent with increasing wave period.

The following behavioural statements and correlations can be made for the test objects themselves:
Both, for the circular plate and for the cone, the object oscialltion increases with increasing wave height and, for the most parts, with increasing wave period. Whereas with increasing submergence depth the object oscillation decreases. Furthermore, the oscillation seems to be larger with greater spring stiffness. The oscillation of the test objects does not seem to have any significant influence on the progression of the waves.
In addition, it can be observed that the oscillation of the circular plate is greater for the same test parameters than the oscialltion of the cone. However, the cone is significantly smaller in relation to the relevant surfaces for the forces than the plate.

It is recommended to carry out further tests with the test object cone in order to have more comparable data with the plate. In relation to the cone, the basic behavior should also be observed with a larger dimensioning and a larger difference between the upper and the bottom diameter.

References

[1] MathWorks. Detecting cars using gaussian mixture models. https://de.mathworks.com/help/vision/examples/detecting-cars-using-gaussian-mixture-models.html, 2018. [acess:27.06.2018].

[2] Dr. Masoud Hayatdavoodi. Motion tracker. Unpublished, March 2018. Matlab Program.

[3] MathWorks. Simple color detection by hue. https://de.mathworks.com/matlabcentral/fileexchange/28512-simplecolordetectionbyhue?focused=5807952&tab=function, 2018. [acess:02.07.2018].

[4] WON Linear Motion System. *Linear Ball Bushing*. WON Linear Motion System, http://www.lunabearings.com/new/Linear_Ball_Bushing.pdf edition. [Online; accessed 09-June-2018].

[5] Klaus Gebhardt. Das umweltlexikon–erneuerbare energien. https://www.umweltdatenbank.de/cms/lexikon/31-lexikon-e/1690-erneuerbare-energie.html, 2018. [Online; accessed 06-August-2018].

[6] H.C.Soerensen Erik Friis-Madsen and S. Parmeggiani. Wave dragon. Presentation, 2012.

[7] Cibran Camba Rey Augusto Barata da Rocha, John Dalen Marc Le Boulluec (Chair), Finn Gunnar Nielsen Henry Jeffrey, Nathalie Rousseau Geoffrey O'Sullivan, Eoin Sweeney, and Judith Wolf . Marine renewable energy research challenges and opportunities for a new energy era in europe. printed, online, October 2010.

[8] Gail Tverberg. The energy collective. http://www.theenergycollective.com/gail-tverberg/2396261/2017-the-year-when-the-world-economy-starts-coming-apart, 2018. [Online; accessed 28-May-2018].

[9] Robin Pelc and Rod M. Fujita. Renewable energy from the ocean. *Marine Policy*, 26(6):471–479, November 2002.

[10] Alain Clément, Pat McCullen, Antonio Falcao, Antonio Fiorentino, Fred Gardner, Karin Hammarlund, Georgios Lemonis, Tony Lewis, Kim Nielsen, Simona Petroncini, M.-Teresa Pontes, Phillippe Schild, Bengt-Olov Sjöström, Hans Christian Sørensen, and Tom Thorpe. Wave energy in europe: Current status and perspectives. 6:405–431, 10 2002.

[11] Frank Neumann (IMIEU) Ruud Kempener (IRENA). Wave energy technology brief. Printed, IRENA Innovation and Technology Centre Robert-Schuman-Platz 3 53175 Bonn Germany, June 2014.

[12] Simona Negro, Floortje Alkemade, and Marko Hekkert. Renewabale and sustainable energy reviews. *Elsevier*, 16, January 2012.

[13] Shuijin Li. Structual analysis of a wave energy device. Master's thesis, University of Dundee, School of Science and Engineering, March 2017.

[14] Malin Goteman Liguo Wang, Jens Engstrom and Jan Isberg. Constrained optimal control of a point absorber wave energy converter with linear generator. *JOURNAL OF RENEWABLE AND SUSTAINABLE ENERGY*, Volume 7, 2015.

[15] Prof. S. A. Sannasiraj. Module ii laboratory wave generation.

[16] Turgut Sarpkaya and Michael Isaacson. *Mechanics of wave forces on offshore structures*. Van Nostrand Reinhold Co New York, 1981.

[17] Farshad Madhi, M.E. Sinclair, and Ronald Yeung. The "berkeley wedge": an asymmetrical energy-capturing floating breakwater of high performance. 9:5–16, 01 2014.

[18] M. Hayatdavoodi, K. Treichel, and R. C. Ertekin. Parametric study of nonlineae wave loads on submerged decks in shallow water. Preprint submitted, manuscript obtained through personal communication with Dr. Hayatdavoodi., 2018.

[19] Xiachen Zhang. Laboratory test of a submerged wave energy device. Master's thesis, School of science and engineering, August 2018.

[20] MathWorks. Foreground detection using gaussian mixture models - matlab - mathworks deutschland. `https://de.mathworks.com/help/vision/ref/vision.foregrounddetector-system-object.html`, 2018. [Online; accessed 09-June-2018].

[21] Yannick Benezeth, Pierre-Marc Jodoin, Bruno Emile, Hélène Laurent, and Christophe Rosenberger. Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, 2010.

[22] MathWorks. vision.blobanalysis system object- properties of connected regions - matlab - mathworks deutschland. `https://de.mathworks.com/help/vision/ref/vision.blobanalysis-system-object.html`, 2018. [Online; accessed 01-July-2018].

[23] MathWorks. Single camera calibrator app - matlab & simulink - mathworks deutschland. `https://de.mathworks.com/help/vision/ug/single-camera-calibrator-app.html`, 2018. [Online; accessed 01-June-2018].

[24] MathWorks. Determine world coordinates of image points - matlab pointstoworld - mathworks deutschland. `https://de.mathworks.com/help/vision/ref/cameraparameters.pointstoworld.html`, 2018. [Online; accessed 01-June-2018].

[25] MathWorks. Image types - matlab & simulink - mathworks deutschland. `https://de.mathworks.com/help/vision/ref/vision.blobanalysis-system-object.html`, 2018. [Online; accessed 01-July-2018].

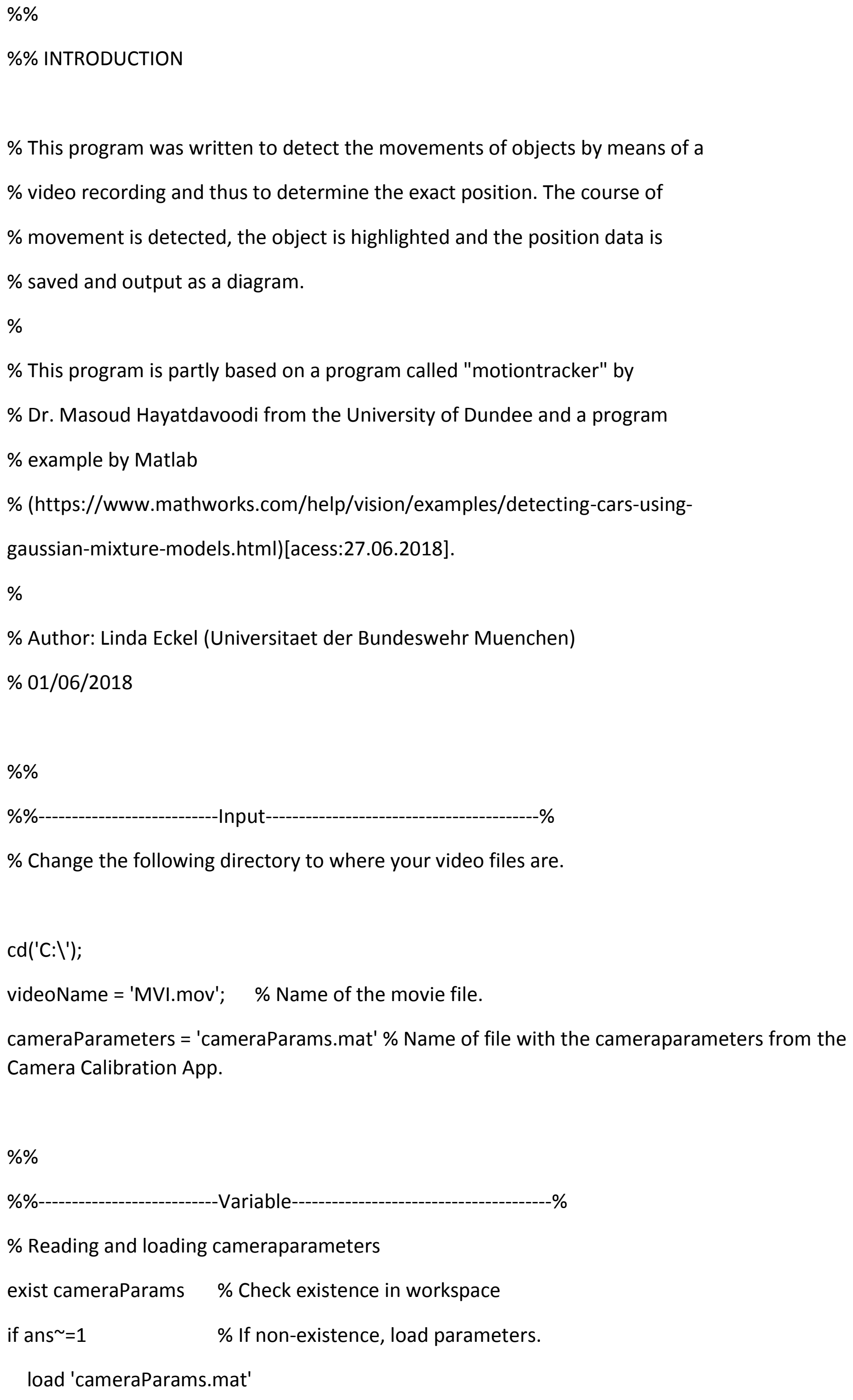
[26] MathWorks. Convert rgb colors to hsv - matlab rgb2hsv - mathworks deutschland. `https://de.mathworks.com/help/matlab/ref/rgb2hsv.html`, 2018. [Online; accessed 11-June-2018].

Appendix

List of Appendix

A. Messuring Program Foreground Detector

Figure 28: Messuring Program Foreground Detector[1],[2].



```
%%

%% INTRODUCTION


% This program was written to detect the movements of objects by means of a

% video recording and thus to determine the exact position. The course of

% movement is detected, the object is highlighted and the position data is

% saved and output as a diagram.

%

% This program is partly based on a program called "motiontracker" by

% Dr. Masoud Hayatdavoodi from the University of Dundee and a program

% example by Matlab

% (https://www.mathworks.com/help/vision/examples/detecting-cars-using-

gaussian-mixture-models.html)[acess:27.06.2018].

%

% Author: Linda Eckel (Universitaet der Bundeswehr Muenchen)

% 01/06/2018


%%

%%---------------------------Input-----------------------------------------%

% Change the following directory to where your video files are.


cd('C:\');

videoName = 'MVI.mov';      % Name of the movie file.

cameraParameters = 'cameraParams.mat' % Name of file with the cameraparameters from the
Camera Calibration App.


%%

%%---------------------------Variable---------------------------------------%

% Reading and loading cameraparameters

exist cameraParams      % Check existence in workspace

if ans~=1               % If non-existence, load parameters.

   load 'cameraParams.mat'
```

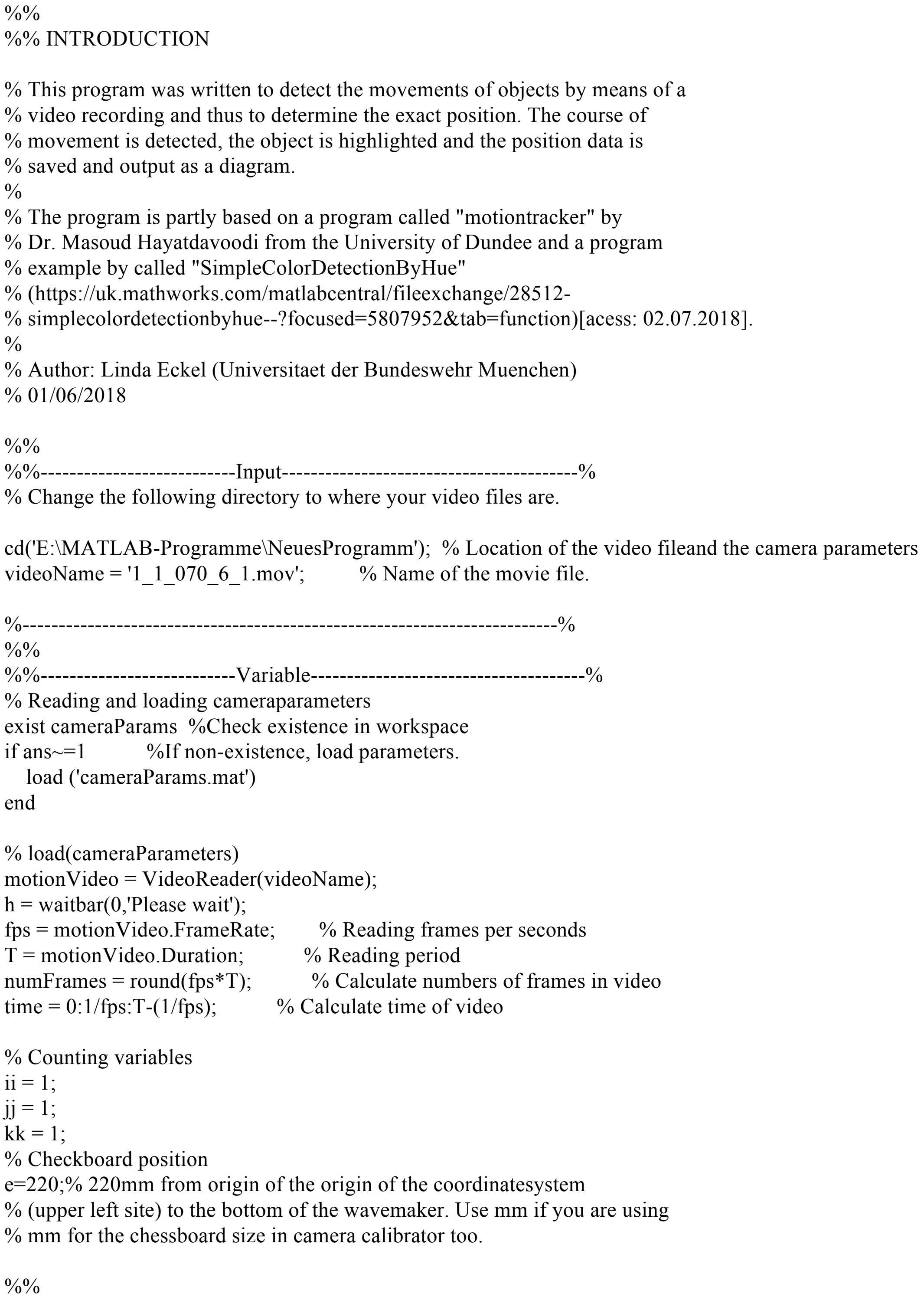
```
end



motionVideo = VideoReader(videoName);
h = waitbar(0,'Please wait');
fps = motionVideo.FrameRate;  %Reading frames per seconds
T = motionVideo.Duration;     %Reading period
numFrames = round(fps*T);     %Calculate numbers of frames in video
time = 0:1/fps:T-(1/fps);     %Calculate time of video

% Counting variables
ii = 1;
jj = 1;
k = 1;
% Checkboard position
e=220;% 220mm from lowest point of chessboard (origin of the
% coordinatesystem of the plot) to the bottom of the wavemaker.
% Use mm if you are using mm for the chessboard size in camera calibrator too.

%%
%%----------------------------Matrix-----------------------------------------%
cent = zeros(numFrames,2);
worldCent = zeros(numFrames,2);

%%
%%--------------------------Start--Programm----------------------------------%
foregroundDetector = vision.ForegroundDetector('NumGaussians', 4, ...
    'MinimumBackgroundRatio', 0.09, 'NumTrainingFrames', 5, 'LearningRate', 0.0001 );

% Proof existing file to safe images in it. If non-existence, create file.
exist Image_Sequence
if ans~=7
```

```
 mkdir Image_Sequence % Creates folder for the extracted frames
end



videoReader = vision.VideoFileReader(videoName);
for jj = 1:numFrames;
    waitbar(jj/numFrames,h,'Waiting for Detecting Foreground');
    % Detect foreground: Moving object will be in white pixels, background
    % in black
    frame = step(videoReader); %Read the next video frame
    foreground = step(foregroundDetector, frame);%Detect foreground

    % Filtered detected foreground for better quality
    se = strel('square', 3);
    filteredForeground = imopen(foreground, se);

    % Find areas of white pixels which are bigger then 300 pixels
    blobAnalysis = vision.BlobAnalysis('AreaOutputPort', true,...
    'CentroidOutputPort', true,...
    'BoundingBoxOutputPort', true,...
    'MinimumBlobArea', 300, 'ExcludeBorderBlobs', false);
    [areas, centroids , boxes] = step(blobAnalysis, foreground);

    % Sort areas with white pixels in descending rder
    [~, idx] = sort(areas, 'Descend');

    % Get the two largest areas.
    if numel(boxes) > 0
        boxes = double(boxes(idx(1), :));
        cent(jj,:) = double(centroids(idx(1),:));
    end
```

```
   % Insert box in coloured Frames
   imDetectedObj = insertObjectAnnotation(frame, 'rectangle', ...
    boxes, 'Object');

   % Save Frames with Boxes
   filename=strcat('frame',num2str(jj),'.jpg');
   folder = 'Image_Sequence';
   fullname = fullfile(folder,filename);
   imwrite(imDetectedObj,fullname);

   % Detect the checkerboard.
   [imagePoints, boardSize] = detectCheckerboardPoints(frame);

   % Compute rotation and translation of the camera.
   [R, t] = extrinsics(imagePoints, cameraParams.WorldPoints, cameraParams);
   % Get the world coordinates of the corners
   worldCent(jj,:) = pointsToWorld(cameraParams, R, t, cent(jj,:));

   jj = jj+1;
end
%%
% Write the processed frames back to a video file

imageNames = dir(fullfile('Image_Sequence','*.jpg'));
imageNames = {imageNames.name}';
outputVideo = VideoWriter(fullfile('motiontrack66.avi'));
outputVideo.FrameRate = motionVideo.FrameRate;
open(outputVideo)
for ii = 1:length(imageNames)
  img = imread(fullfile('Image_Sequence',imageNames{ii}));
  writeVideo(outputVideo,img)
  waitbar(ii/numFrames,h,'Reconstructing video');
```

B. Messuring Program Motion Track 2

Figure 29: Messuring Program Motion Track 2[3],[2].



```
%%
%% INTRODUCTION

% This program was written to detect the movements of objects by means of a
% video recording and thus to determine the exact position. The course of
% movement is detected, the object is highlighted and the position data is
% saved and output as a diagram.
%
% The program is partly based on a program called "motiontracker" by
% Dr. Masoud Hayatdavoodi from the University of Dundee and a program
% example by called "SimpleColorDetectionByHue"
% (https://uk.mathworks.com/matlabcentral/fileexchange/28512-
% simplecolordetectionbyhue--?focused=5807952&tab=function)[acess: 02.07.2018].
%
% Author: Linda Eckel (Universitaet der Bundeswehr Muenchen)
% 01/06/2018

%%
%%--------------------------Input----------------------------------------%
% Change the following directory to where your video files are.

cd('E:\MATLAB-Programme\NeuesProgramm');  % Location of the video fileand the camera parameters
videoName = '1_1_070_6_1.mov';          % Name of the movie file.

%-----------------------------------------------------------------------%
%%
%%--------------------------Variable-------------------------------------%
% Reading and loading cameraparameters
exist cameraParams  %Check existence in workspace
if ans~=1           %If non-existence, load parameters.
   load ('cameraParams.mat')
end

% load(cameraParameters)
motionVideo = VideoReader(videoName);
h = waitbar(0,'Please wait');
fps = motionVideo.FrameRate;        % Reading frames per seconds
T = motionVideo.Duration;           % Reading period
numFrames = round(fps*T);           % Calculate numbers of frames in video
time = 0:1/fps:T-(1/fps);           % Calculate time of video

% Counting variables
ii = 1;
jj = 1;
kk = 1;
% Checkboard position
e=220;% 220mm from origin of the origin of the coordinatesystem
% (upper left site) to the bottom of the wavemaker. Use mm if you are using
% mm for the chessboard size in camera calibrator too.

%%
```

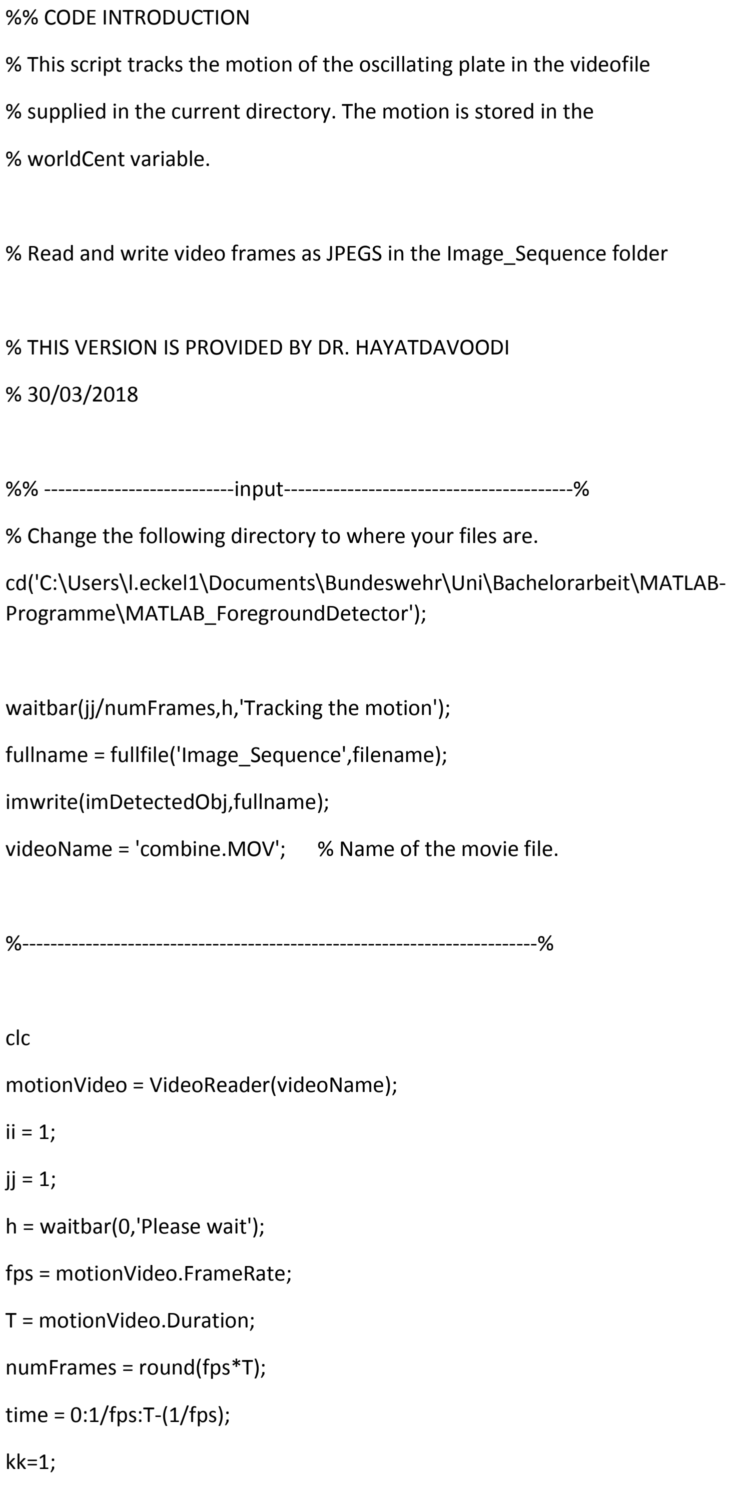
```
%%----------------------------Matrix-----------------------------------------%
cent = zeros(numFrames,2);
worldCent = zeros(numFrames,2);

%%
%%-------------------------Start--Programm----------------------------------%

%Proof existing file to safe images in it. If non-existence, create file.
exist Image_Sequence
if ans~=7
 mkdir Image_Sequence %Creates folder for the extracted frames
end

% Cut video into picture squence and save them.
while kk<=numFrames
   waitbar(kk/numFrames,h,'Extracting frames from video');
   img = readFrame(motionVideo);
   filename = [sprintf('%03d',kk) '.jpg'];  % Filname (001, 002, ...)
   folder = 'Image_Sequence';              % Storage location
   fullname = fullfile(folder,filename);
   imwrite(img,fullname)  % Write out to a JPEG file with filname in folder
   kk = kk+1;
end

% Read video again.Reads the video again, because the generated images of
% the folder "Image_Sequence" are changed to black and white images when the
% object is detected. Only the object which is detected is displayed as
% white pixels. For better visual inspection of which object has been
% identified, the yellow box, that marks the found object, is inserted into
% original images.
videoReader = vision.VideoFileReader(videoName);

% Detecing Object
for jj = 1:numFrames;

 waitbar(jj/numFrames,h,'Detecting and Tracking Movement of Object');

 filename = [sprintf('%03d',jj) '.jpg'];
 folder = 'Image_Sequence'
 fullname = fullfile(folder,filename);

 % Identifies the color of each pixel and assigns values to each colour.
 % These arrays are describing. Some pictures have a colour chart where is
 % inetered in each pixel one value for the colour. Thus, every pixel in
 % the colour map has a numerical value that describes it.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
% This section is subject to the following copyright laws:
%Copyright (c) 2015, Image Analyst
%Copyright (c) 2018, Image Analyst
%All rights reserved.
```

```
%Redistribution and use in source and binary forms, with or without
%modification, are permitted provided that the following conditions are met:

%* Redistributions of source code must retain the above copyright notice, this
%  list of conditions and the following disclaimer.

%* Redistributions in binary form must reproduce the above copyright notice,
%  this list of conditions and the following disclaimer in the documentation
%  and/or other materials provided with the distribution
%THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
%AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
%IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE
%DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
%FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
%DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
%SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
%CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
%OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
%OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



 [rgbImage, storedColorMap] = imread(fullname);

 [rows, columns, numberOfColorBands] = size(rgbImage);

 % The image can be of class uint8, uint16, single, or double.
 % If it's monochrome (indexed), convert it to color.
 % Check to see if it's an 8-bit image needed later for scaling.
 if strcmpi(class(rgbImage), 'uint8')
  % Flag for 256 gray levels.
  eightBit = true;
 else
  eightBit = false;
    end


% There are three image Types. True color Image, indexed image and a
% grayscale image. A true color images discribes each color with three
% values. These values are descirbing the terms of the colour amount of
% red, green, and blue which are repesented.  Depending on the class of the
% array (uint8, uint16, single, or double), the values are in a range of
% values between 0 and 1 (double and single), 0 and 255 (uint8) and 0 to
% 65535 (uint16). Uint16 has such high quality, that it will not use for
% normal pictures and this program.  The second type is a grayscale image,
% here the color is only described by a numerical value, since this is
% sufficient to clearly identify and delimit the color. In the third and
% last type, the indexed image, the color is the only one described by a
% vector and a color map. As for the application of the program the vector
% representation is required by three numerical values of the true colour
% image, the other cases are checked.


% Check if it is a grayscale image. If this is the case, the color has only
% one value, this value is sufficient to describe the color. However, since
% three values are required everywhere, the missing two numerical values are
% added by using the first value. This discribes the same color.
 if numberOfColorBands == 1
  if isempty(storedColorMap)
   % Just a simple gray level image, not indexed with a stored color map.
   % Create a 3D true color image where we copy the monochrome image into all 3 (R, G, & B) color planes.
   rgbImage = cat(3, rgbImage, rgbImage, rgbImage);
  else
   % It's an indexed image because the color map is not empty.
   rgbImage = ind2rgb(rgbImage, storedColorMap);
   % ind2rgb() will convert it to double and normalize it to the range 0-1.
   % Convert back to uint8 in the range 0-255, if needed.
   if eightBit
    rgbImage = uint8(255 * rgbImage);
         end

  end
 end
 % Convert RGB image to HSV. HSV describes the properties hue,
    % saturation and value using three values.It the color description of
    % importance, HSV model is preferred to RGB model. The reason for this
    % is that the HSV model is more similar to the human visual perception
    % and therefore easier to define and select colour value ranges.
 hsvImage = rgb2hsv(rgbImage);
 % Extract out the H, S, and V images individually
 hImage = hsvImage(:,:,1);
 sImage = hsvImage(:,:,2);
 vImage = hsvImage(:,:,3);

    % Assign the low and high thresholds for each color band.

  % Take a guess at the values that might work for the image.
  hueThresholdLow = 0.09;
  hueThresholdHigh = 0.18;
  saturationThresholdLow = 0;
  saturationThresholdHigh = 0.7;
  valueThresholdLow = 0.3;
  valueThresholdHigh = 1.0;



    % Now apply each color band's particular thresholds to the color band
 hueMask = (hImage >= hueThresholdLow) & (hImage <= hueThresholdHigh);
 saturationMask = (sImage >= saturationThresholdLow) & (sImage <= saturationThresholdHigh);
 valueMask = (vImage >= valueThresholdLow) & (vImage <= valueThresholdHigh);
    % Combine the masks to find where all 3 are "true."
 coloredObjectsMask = uint8(hueMask & saturationMask & valueMask);

 % Tell user that we're going to filter out small objects.
 smallestAcceptableArea = 100;

   % Get rid of small objects.
 coloredObjectsMask = uint8(bwareaopen(coloredObjectsMask, smallestAcceptableArea));

   % Smooth the border using a morphological closing operation, imclose().
 structuringElement = strel('disk', 4);
 coloredObjectsMask = imclose(coloredObjectsMask, structuringElement);

 % Fill in any holes in the regions.
 coloredObjectsMask = imfill(logical(coloredObjectsMask), 'holes');

%The section of the copyrights above end here.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % Findig connecting components which have more than 300 pixels.
   blobAnalysis = vision.BlobAnalysis('AreaOutputPort', true,...
   'CentroidOutputPort', true,...
   'BoundingBoxOutputPort', true,...
   'MinimumBlobArea',300, 'ExcludeBorderBlobs', false);
   [areas, centroids , boxes] = step(blobAnalysis, coloredObjectsMask);


   % Sort connected components in descending order by area
   [~, idx] = sort(areas, 'Descend');

   % Get the two largest components.
   if numel(boxes) > 0
      boxes = double(boxes(idx(1), :));
      cent(jj,:) = double(centroids(idx(1),:));
   end

   % Use original images which has not been changed by color recognition.
   frame = step(videoReader);

   % Insert box to show detected area.
   imDetectedObj = insertObjectAnnotation(frame, 'rectangle', ...
   boxes, 'Object');

   % Save original image with tagged object
   filename2 = [sprintf('%03d',jj) '.jpg'];
   folder = 'Image_Sequence';
   fullname2 = fullfile(folder,filename2);
   imwrite(imDetectedObj,fullname2);

   % Detect the checkerboard.
   [imagePoints, boardSize] = detectCheckerboardPoints(frame);
```

```
    % Compute rotation and translation of the camera.
    [R, t] = extrinsics(imagePoints, cameraParams.WorldPoints, cameraParams);
    % Get the world coordinates of the corners
    worldCent(jj,:) = pointsToWorld(cameraParams, R, t, cent(jj,:));

    jj = jj+1;

end

% Write the processed frames back to a video file

imageNames = dir(fullfile('Image_Sequence','*.jpg'));
imageNames = {imageNames.name}';
outputVideo = VideoWriter(fullfile('Motiontrack.avi'));
outputVideo.FrameRate = motionVideo.FrameRate;
open(outputVideo)
for ii = 1:length(imageNames)
   img = imread(fullfile('Image_Sequence',imageNames{ii}));
   writeVideo(outputVideo,img)
   waitbar(ii/numFrames,h,'Reconstructing video');
end
close(outputVideo)
motiontrack = VideoReader(fullfile('Motiontrack.avi'));
ii = 1;
while hasFrame(motiontrack)
   mov(ii) = im2frame(readFrame(motiontrack));
   ii = ii+1;
   waitbar(ii/numFrames,h,'Creating new video');
end

%%
%Creating matrix only with time and y-values in consideration of the
%position of the chessboard.

worldCent=worldCent*(-1);
a=ones(numFrames,1);
b=[0 e];
c=a*b;
d=worldCent+c;
e=[0 0; 0 1];
x=d*e;
position=transpose(time)+x; % Matrix with time and position

%%
%Save matrix with time and position of the object
%inform about saving file
message = sprintf('Save Object Position?\nContinue?');
 reply = questdlg(message, 'Continue', 'OK', 'Cancel', 'OK');
 if strcmpi(reply, 'Cancel')
  % User said cancel, so exit.
    else
        [file, path] = uiputfile('test.xls','Speichern');
```

C. Submittal Messuring Program Motion Tracker

Figure 30: Submittal Messuring Program Motion Tracker[2].



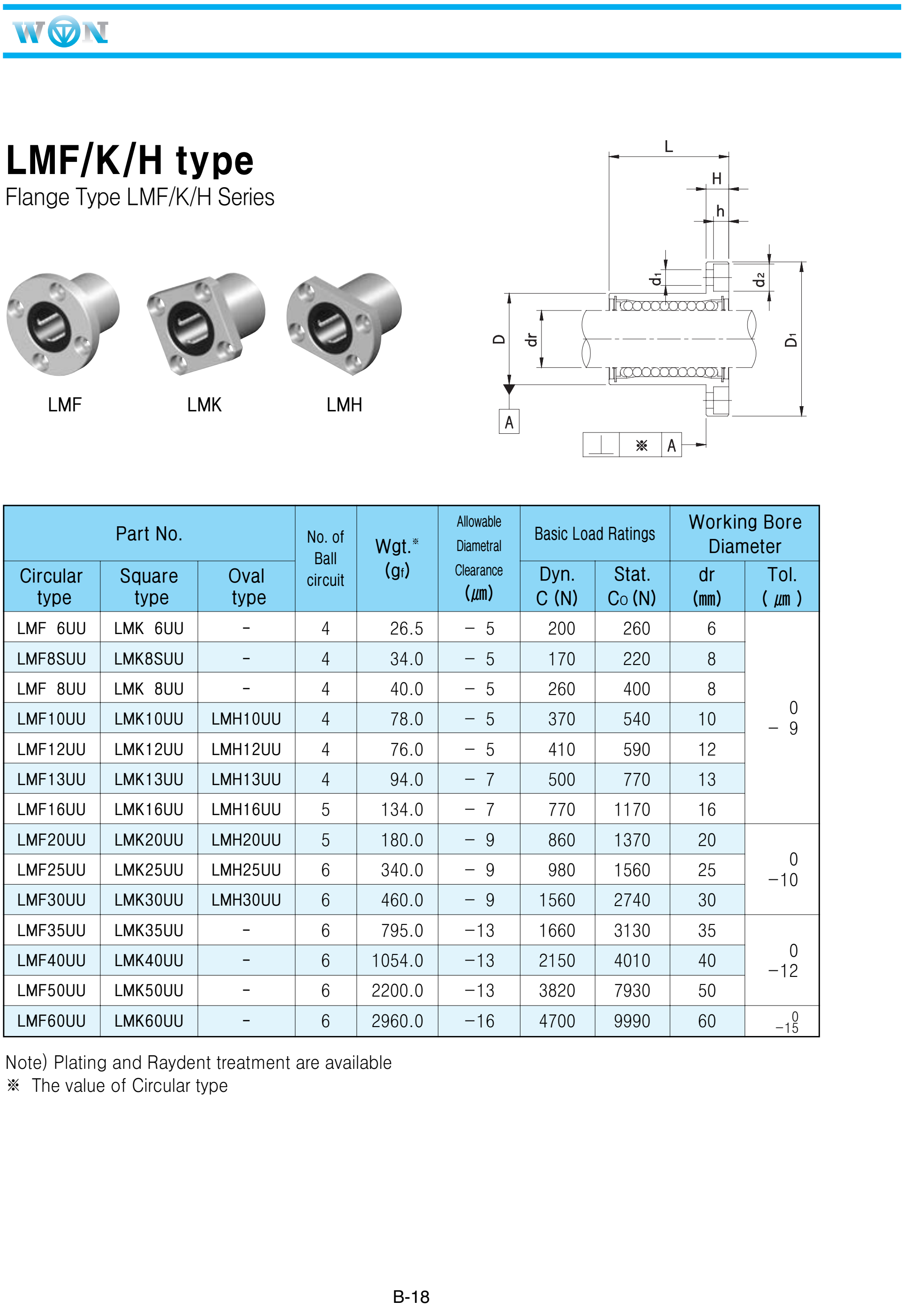
```
%% CODE INTRODUCTION

% This script tracks the motion of the oscillating plate in the videofile

% supplied in the current directory. The motion is stored in the

% worldCent variable.


% Read and write video frames as JPEGS in the Image_Sequence folder


% THIS VERSION IS PROVIDED BY DR. HAYATDAVOODI

% 30/03/2018


%% --------------------------input----------------------------------------%

% Change the following directory to where your files are.

cd('C:\Users\l.eckel1\Documents\Bundeswehr\Uni\Bachelorarbeit\MATLAB-
Programme\MATLAB_ForegroundDetector');


waitbar(jj/numFrames,h,'Tracking the motion');

fullname = fullfile('Image_Sequence',filename);

imwrite(imDetectedObj,fullname);

videoName = 'combine.MOV';      % Name of the movie file.


%----------------------------------------------------------------------%


clc

motionVideo = VideoReader(videoName);

ii = 1;

jj = 1;

h = waitbar(0,'Please wait');

fps = motionVideo.FrameRate;

T = motionVideo.Duration;

numFrames = round(fps*T);

time = 0:1/fps:T-(1/fps);

kk=1;
```

```
while kk<=numFrames
  img = readFrame(motionVideo);
  filename = [sprintf('%03d',kk) '.jpg'];
  folder = 'Image_Sequence';
  fullname = fullfile(folder,filename);
  imwrite(img,fullname)    % Write out to a JPEG file (img1.jpg, img2.jpg, etc.)
  kk = kk+1;
  waitbar(kk/numFrames,h,'Extracting frames from video');
end

cent = zeros(numFrames,2);
worldCent = zeros(numFrames,2);

while jj <= numFrames

% Undistort current frame

filename = [sprintf('%03d',jj) '.jpg'];

imOrig = imread(fullfile('Image_Sequence',filename));
%[im, newOrigin] = undistortImage(imOrig, cameraParams, 'Outputview', ...
    'full');

% Segment out foreground based on color space
imHSV = rgb2hsv(imOrig);
saturation = imHSV(:,:,2);

g = graythresh(saturation);
imObj = (saturation > g);

% Find connected components.
```

```
blobAnalysis = vision.BlobAnalysis('AreaOutputPort', true,...
   'CentroidOutputPort', true,...
   'BoundingBoxOutputPort', true,...
   'MinimumBlobArea', 300, 'ExcludeBorderBlobs', false);

[areas, centroids , boxes] = step(blobAnalysis, imObj);

% Sort connected components in descending order by area
[~, idx] = sort(areas, 'Descend');

% Get the two largest components.
if numel(boxes) > 0
boxes = double(boxes(idx(1), :));
cent(jj,:) = double(centroids(idx(1),:));
end

imDetectedObj = insertObjectAnnotation(im, 'rectangle', ...
    boxes, 'Object');

waitbar(jj/numFrames,h,'Tracking the motion');
fullname = fullfile('Image_Sequence',filename);
imwrite(imDetectedObj,fullname)

% Convert image space to real world Euclidean space

% Detect the checkerboard.
[imagePoints, boardSize] = detectCheckerboardPoints(im);



% Compute rotation and translation of the camera.
[R, t] = extrinsics(imagePoints, cameraParams.WorldPoints, cameraParams);
% Get the world coordinates of the corners
```

D. Basic Load Ratings Linear Bearings

Figure 31: Basic Load Ratings Linear Bearings[4].



WON
L
H
h
d1
d2
D
dr
D1
A
⊥ ※ A
LMF
LMK
LMH

LMF/K/H type

Flange Type LMF/K/H Series

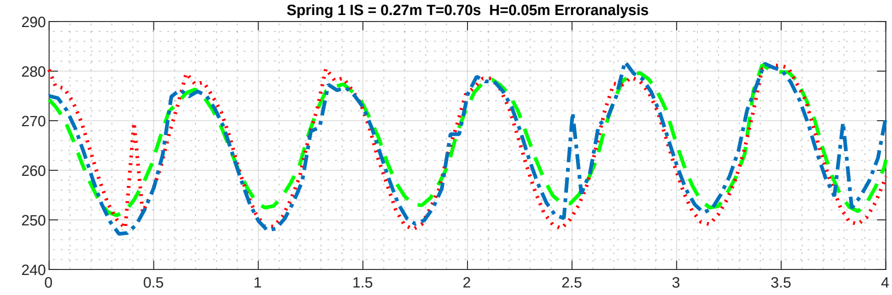
Part No.			No. of Ball circuit	Wgt.※ (gf)	Allowable Diametral Clearance (㎛)	Basic Load Ratings		Working Bore Diameter	
Circular type	Square type	Oval type				Dyn. C (N)	Stat. Co (N)	dr (㎜)	Tol. (㎛)
LMF 6UU	LMK 6UU	-	4	26.5	− 5	200	260	6	0 − 9
LMF8SUU	LMK8SUU	-	4	34.0	− 5	170	220	8	
LMF 8UU	LMK 8UU	-	4	40.0	− 5	260	400	8	
LMF10UU	LMK10UU	LMH10UU	4	78.0	− 5	370	540	10	
LMF12UU	LMK12UU	LMH12UU	4	76.0	− 5	410	590	12	
LMF13UU	LMK13UU	LMH13UU	4	94.0	− 7	500	770	13	
LMF16UU	LMK16UU	LMH16UU	5	134.0	− 7	770	1170	16	
LMF20UU	LMK20UU	LMH20UU	5	180.0	− 9	860	1370	20	0 −10
LMF25UU	LMK25UU	LMH25UU	6	340.0	− 9	980	1560	25	
LMF30UU	LMK30UU	LMH30UU	6	460.0	− 9	1560	2740	30	
LMF35UU	LMK35UU	-	6	795.0	−13	1660	3130	35	0 −12
LMF40UU	LMK40UU	-	6	1054.0	−13	2150	4010	40	
LMF50UU	LMK50UU	-	6	2200.0	−13	3820	7930	50	
LMF60UU	LMK60UU	-	6	2960.0	−16	4700	9990	60	0 −15

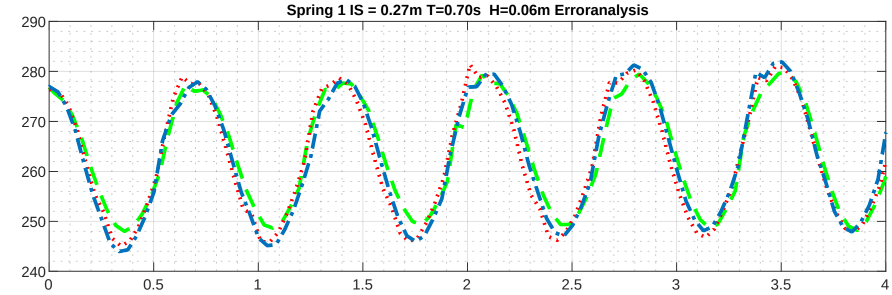
Note) Plating and Raydent treatment are available

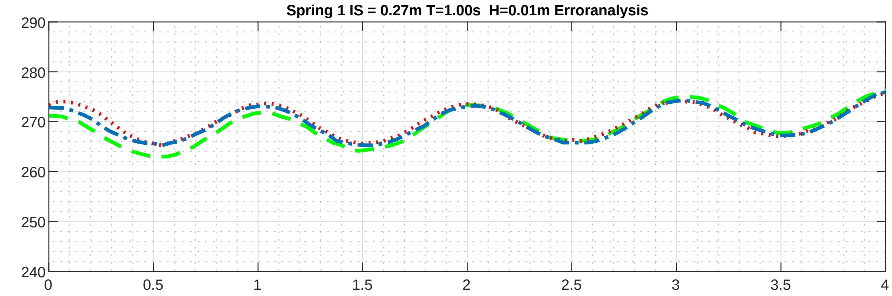
※ The value of Circular type

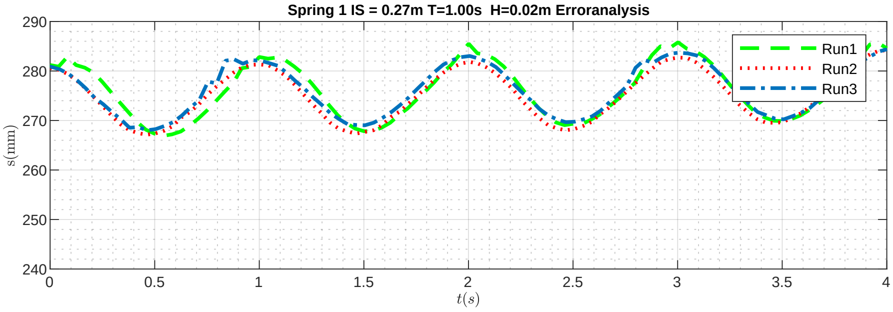
B-18

E. Example Excerpt of Plots Erroranalysis oscillation Height


Spring 1 IS = 0.27m T=0.70s H=0.05m Erroranalysis


Spring 1 IS = 0.27m T=0.70s H=0.06m Erroranalysis


Spring 1 IS = 0.27m T=1.00s H=0.01m Erroranalysis


Spring 1 IS = 0.27m T=1.00s H=0.02m Erroranalysis

F. Erroranalysis Oszillation Height

Table 7: Erroranalysis Oszillation Height

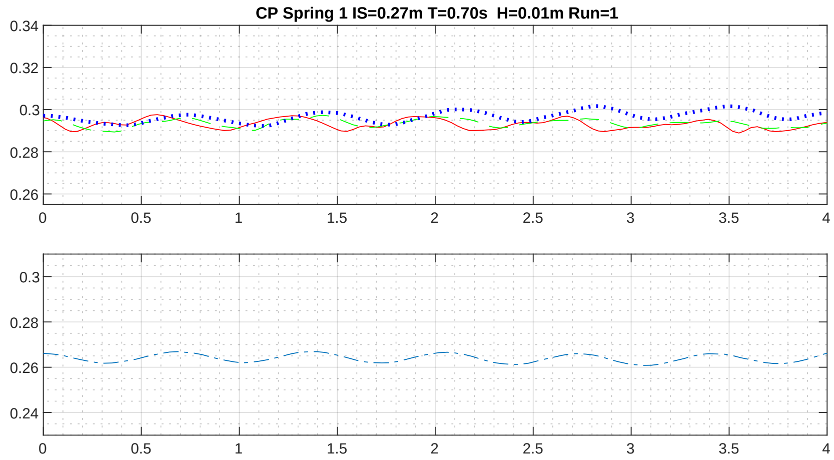
Oscillation height error $H_{oz,error}$ [%] spring 1, IS = 0.27 m					
H [mm] \ **P [s]**	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
10	36.585	7.843	3.158	–	–
20	2.679	12.903	4.523	7.895	32.274
30	6.134	16.837	2.504	14.308	9.722
40	21.708	27.128	12.635	14.224	19.135
50	22.600	19.197	17.010	9.307	6.608
60	12.460	31.496	10.855	4.282	25.445

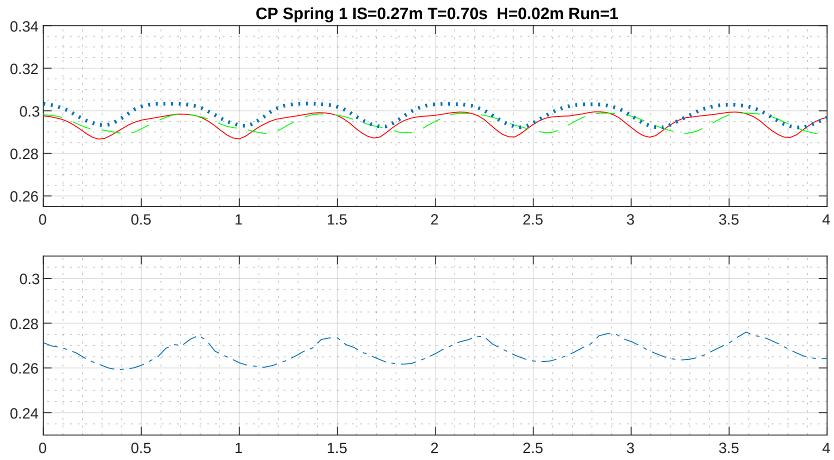
Oscillation height error $H_{oz,error}$ [%] spring 1, IS = 0.21 m					
H [mm] \ **P [s]**	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
10	39.669	4.225	15.190	–	–
20	11.588	128.571	7.031	3.896	10.881
30	5.590	6.250	10.297	15.854	4.810
40	1.342	6.742	2.857	10.955	7.500
50	40.244	3.301	5.488	36.720	19.041
60	13.500	5.849	4.324	18.847	21.951

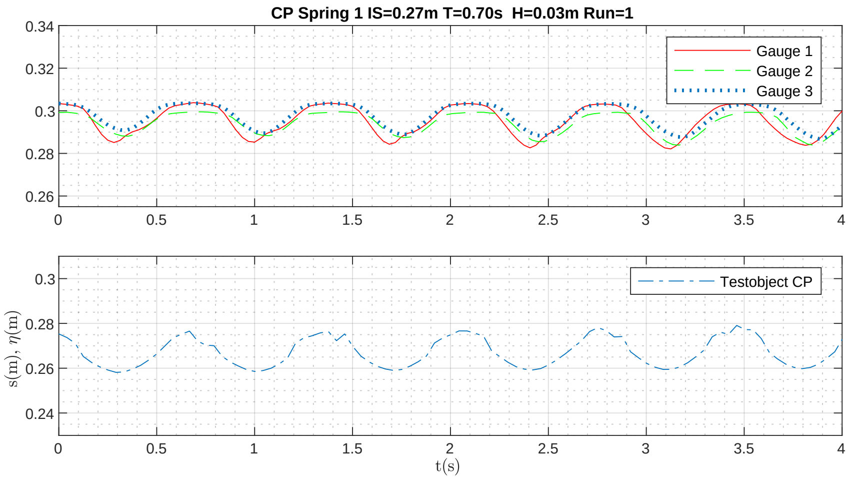
Oscillation height error $H_{oz,error}$ [%] spring 1, IS = 0.15 m					
H [mm] \ **P [s]**	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
10	178.378	14.286	11.413	–	–
20	27.778	58.696	11.765	14.907	31.214
30	7.229	8.411	11.407	6.774	18.689
40	8.434	6.040	6.818	14.005	21.053
50	14.201	3.896	6.027	4.200	29.876
60	4.945	6.865	12.233	11.205	25.789

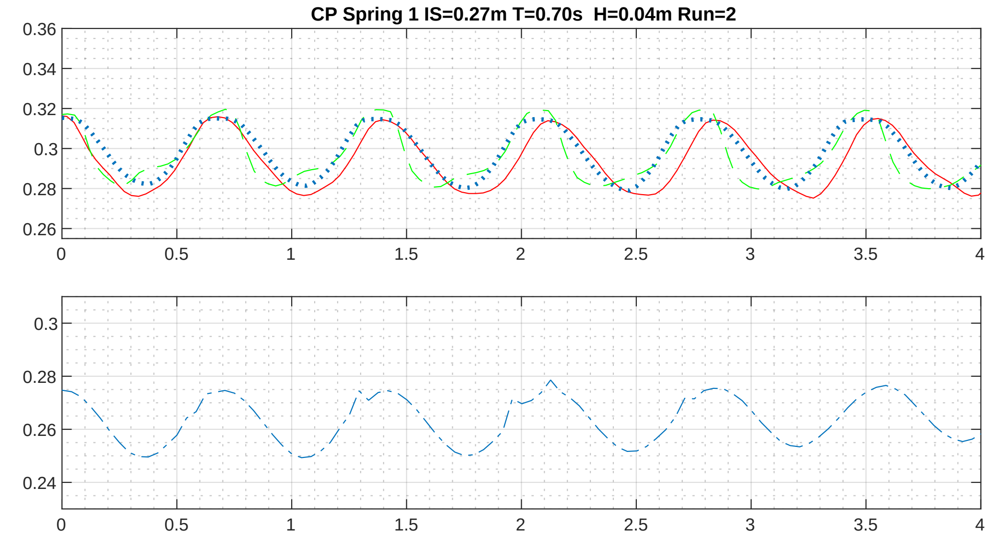
Oscillation height error $\mathbf{H_{oz,error}}$ [%] spring 1, IS = 0.09 m					
P [s] **H [mm]**	**0.70**	**1.00**	**1.25**	**1.50**	**1.75**
10	20.000	45.652	95.455	–	–
20	54.207	20.594	14.596	10.791	23.388
30	27.433	23.909	13.421	31.479	30.710
40	6.462	14.414	18.796	3.891	9.935
50	9.661	3.792	12.623	10.762	9.378
60	2.290	8.999	3.295	12.415	15.228

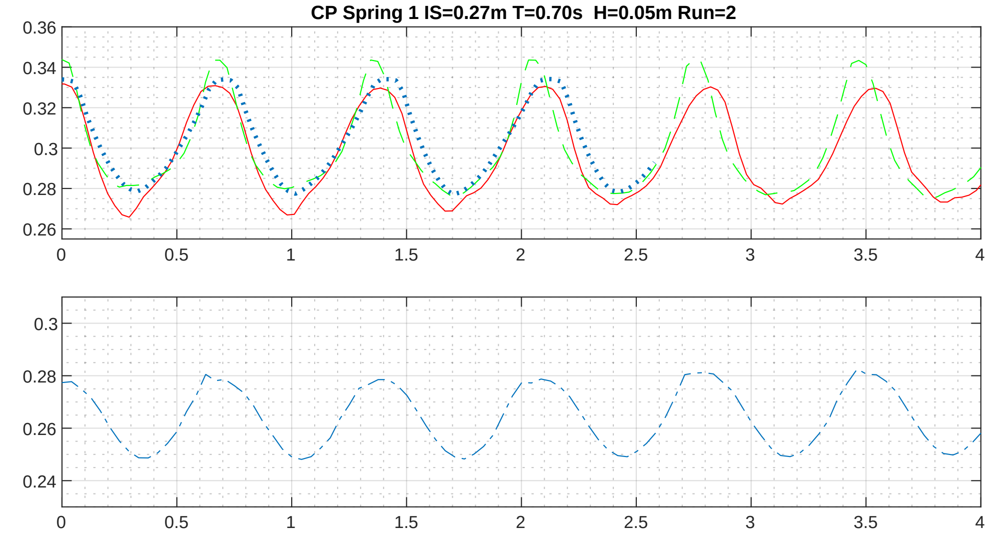
G. Plots Oscillation Height Testobject Circular Plate

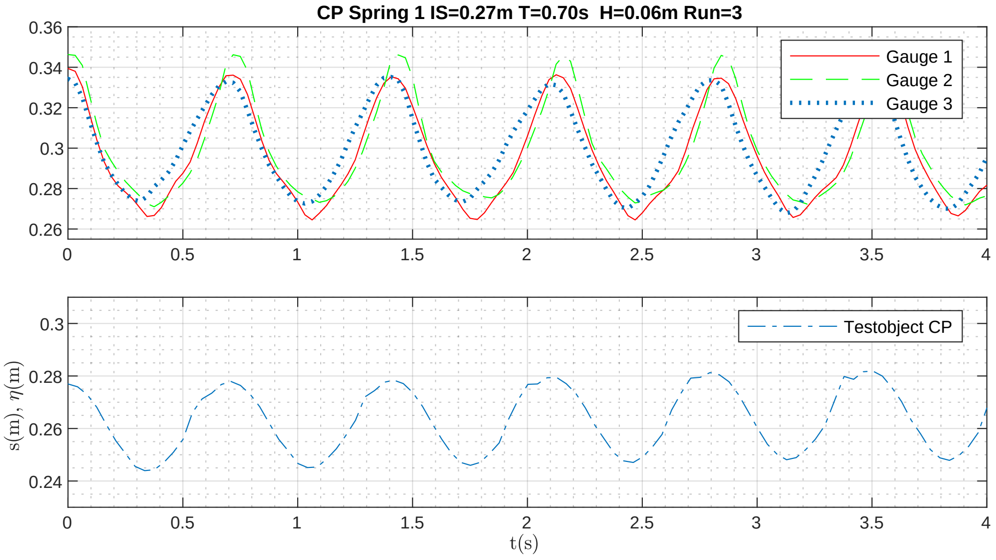


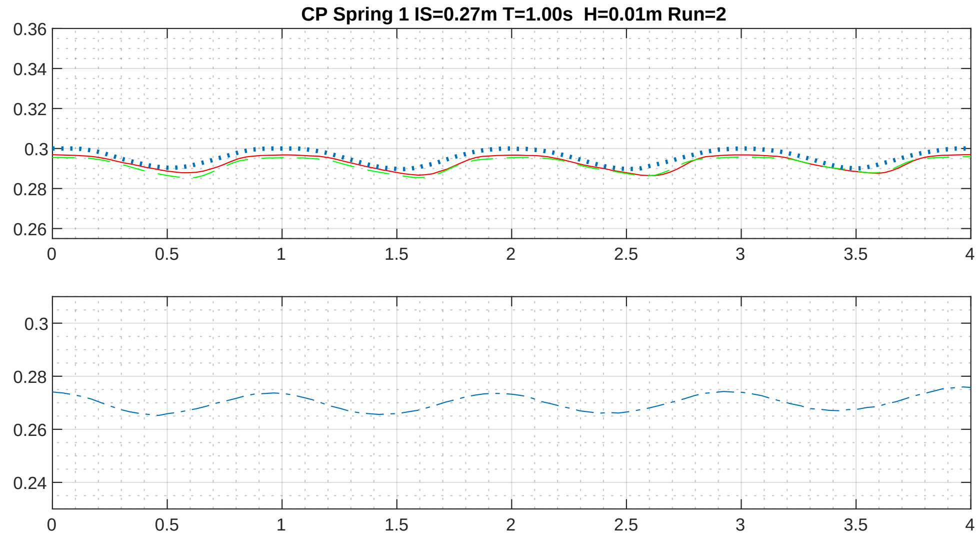


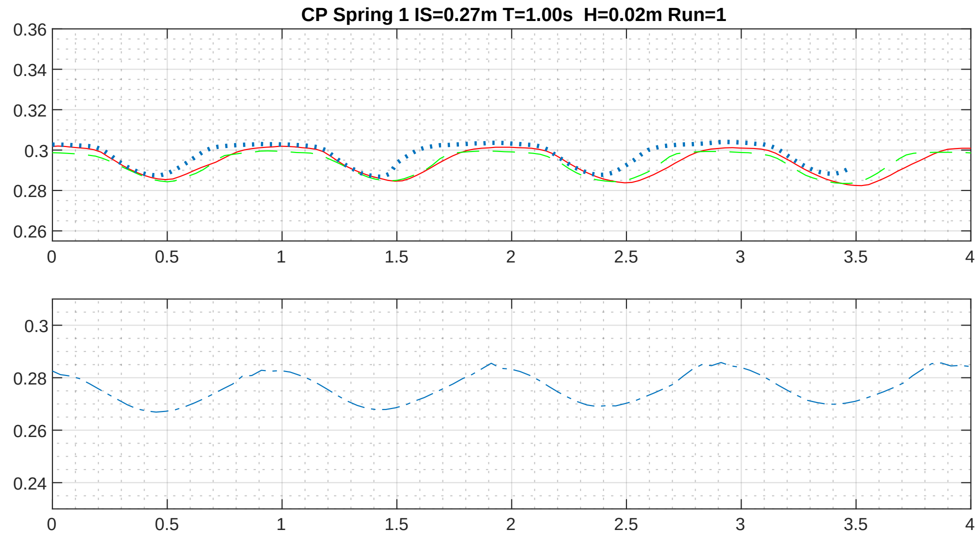


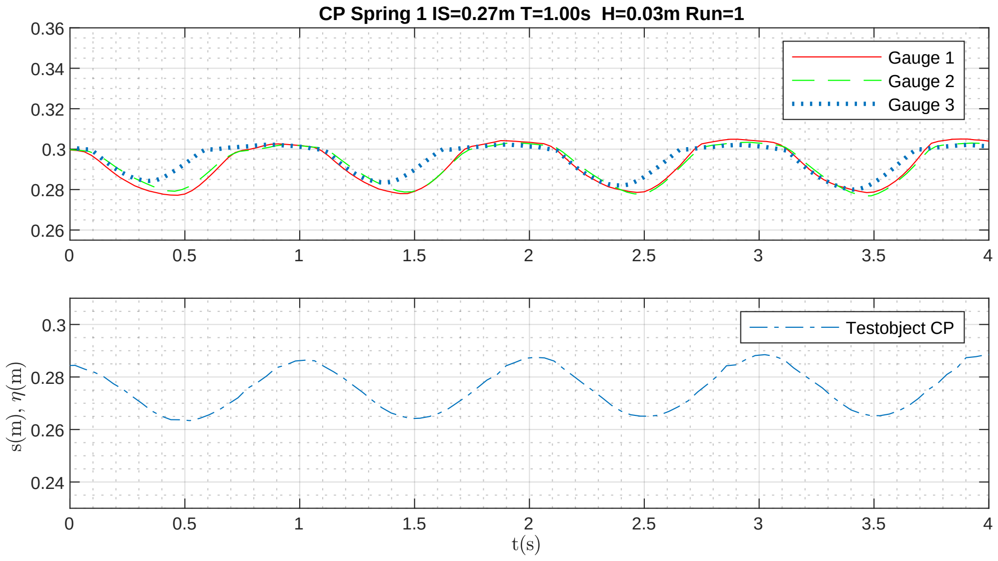

CP Spring 1 IS=0.27m T=0.70s H=0.04m Run=2
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4

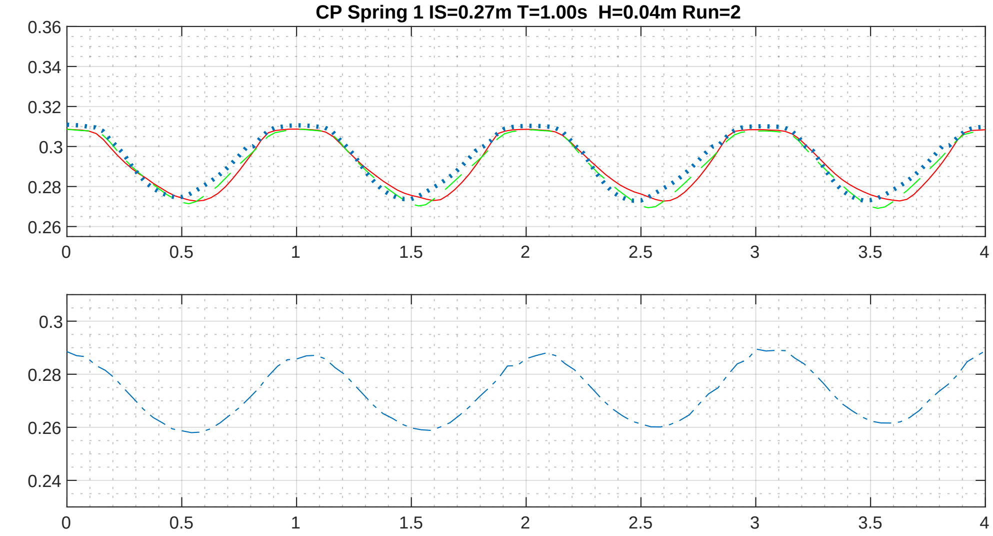

CP Spring 1 IS=0.27m T=0.70s H=0.05m Run=2
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4

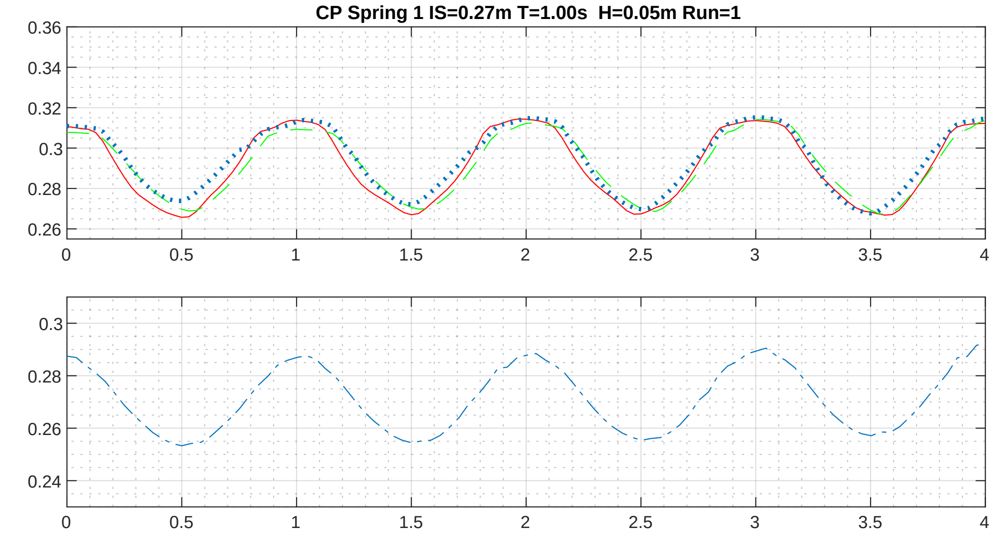

CP Spring 1 IS=0.27m T=0.70s H=0.06m Run=3
Gauge 1
Gauge 2
Gauge 3
Testobject CP
s(m), η(m)
t(s)
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4

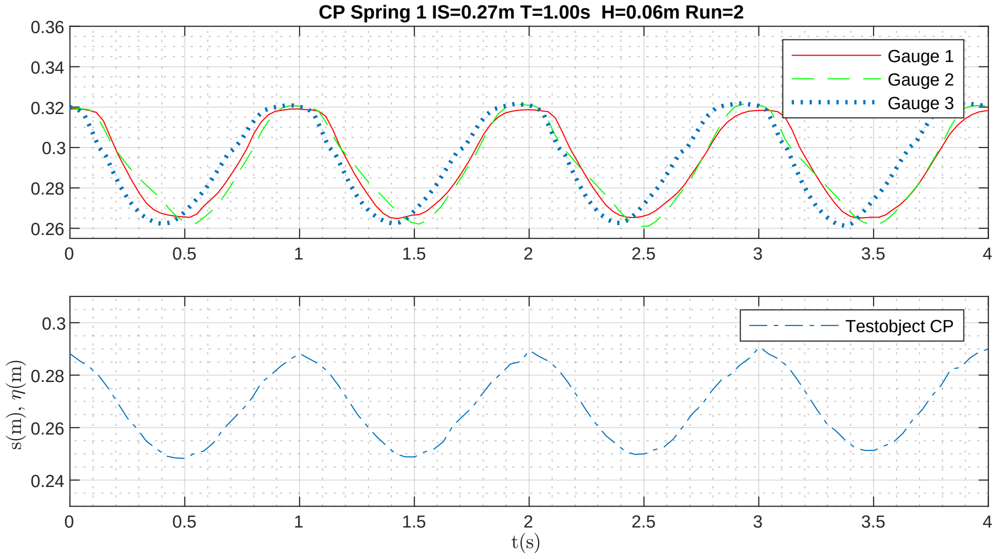

CP Spring 1 IS=0.27m T=1.00s H=0.01m Run=2
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4

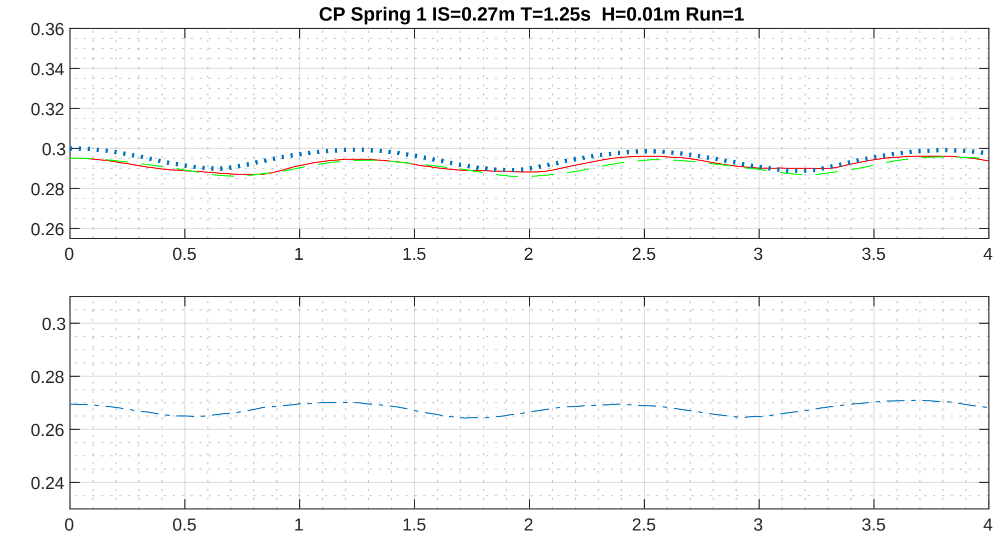

CP Spring 1 IS=0.27m T=1.00s H=0.02m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4

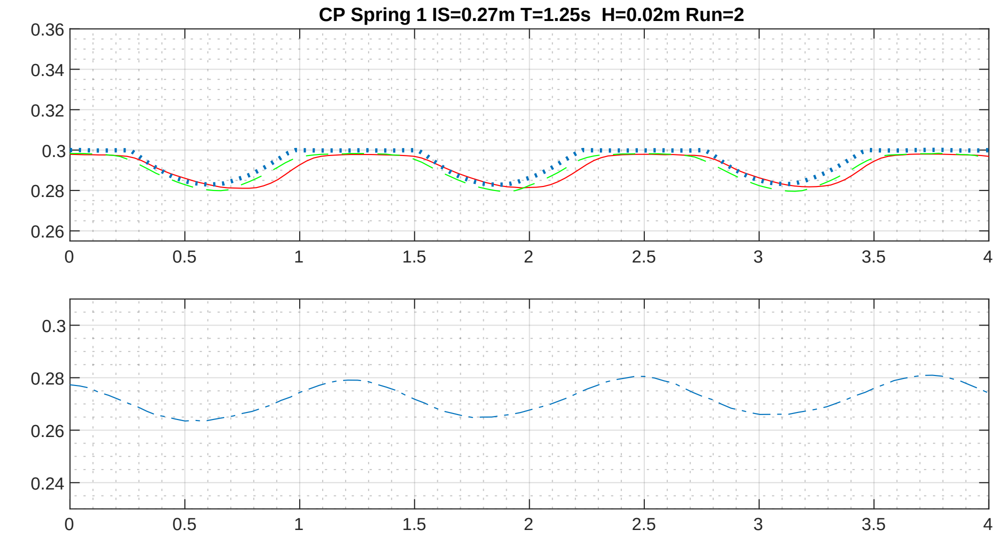

CP Spring 1 IS=0.27m T=1.00s H=0.03m Run=1
Gauge 1
Gauge 2
Gauge 3
Testobject CP
s(m), η(m)
t(s)
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4

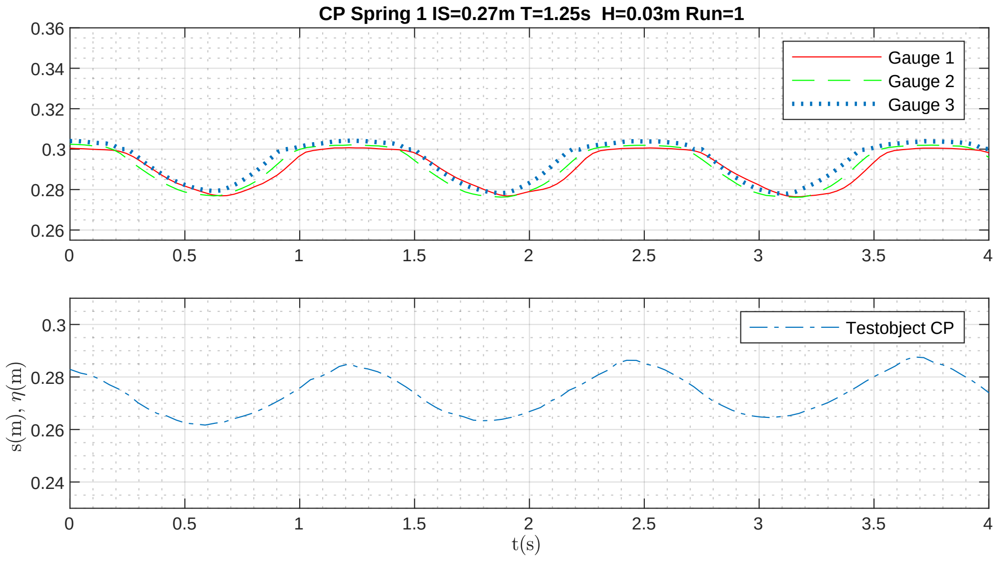

CP Spring 1 IS=0.27m T=1.00s H=0.04m Run=2

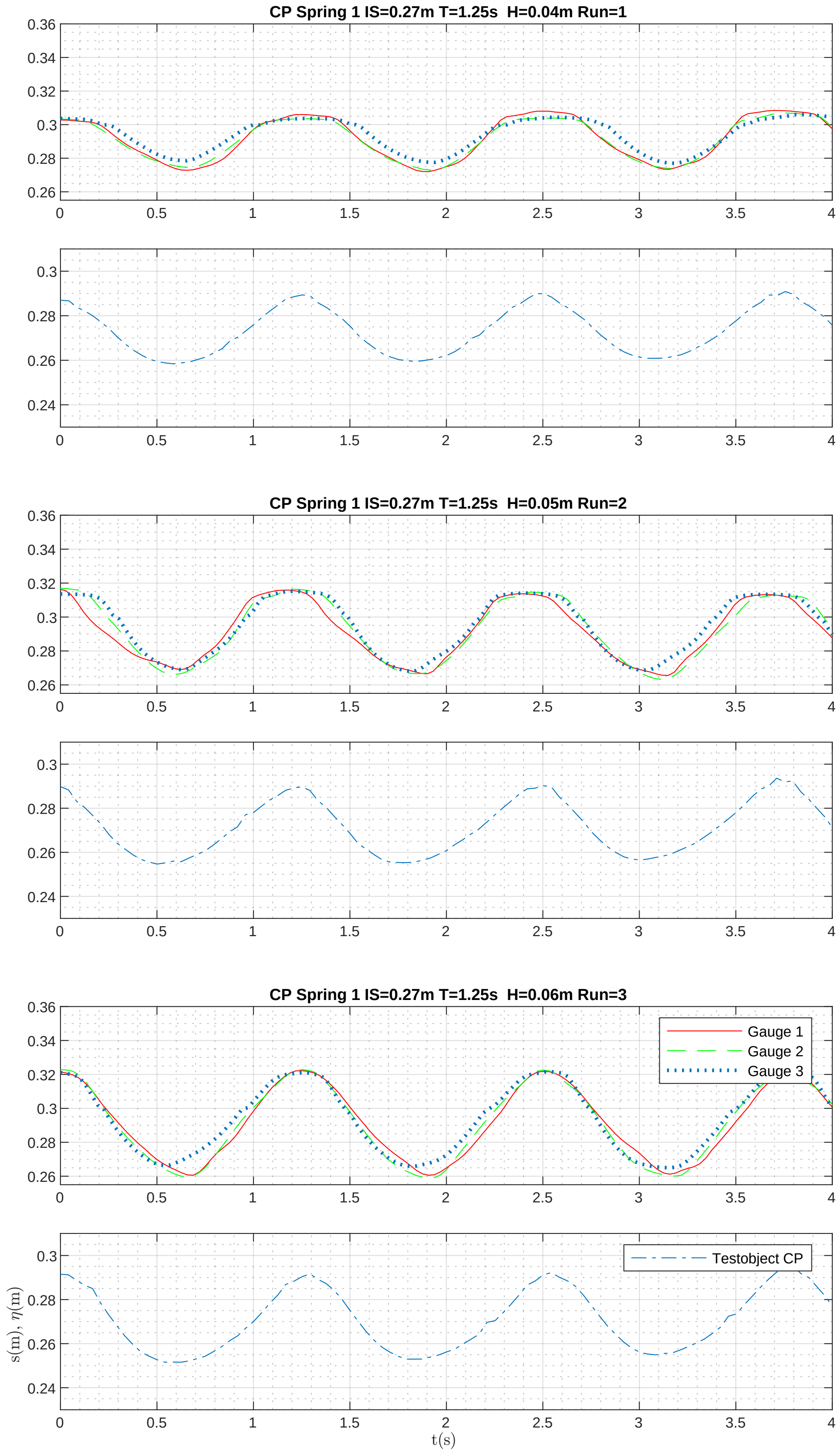

CP Spring 1 IS=0.27m T=1.00s H=0.05m Run=1

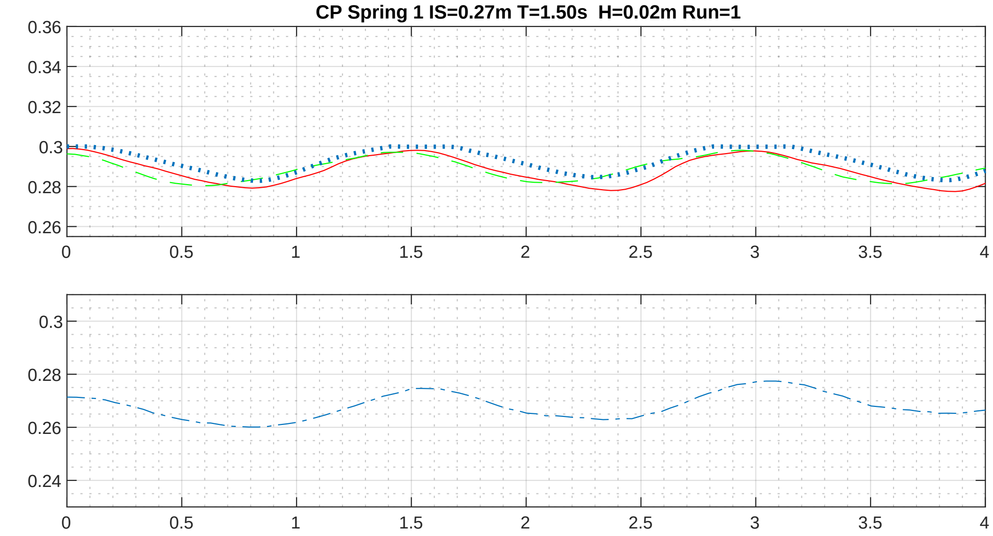

CP Spring 1 IS=0.27m T=1.00s H=0.06m Run=2

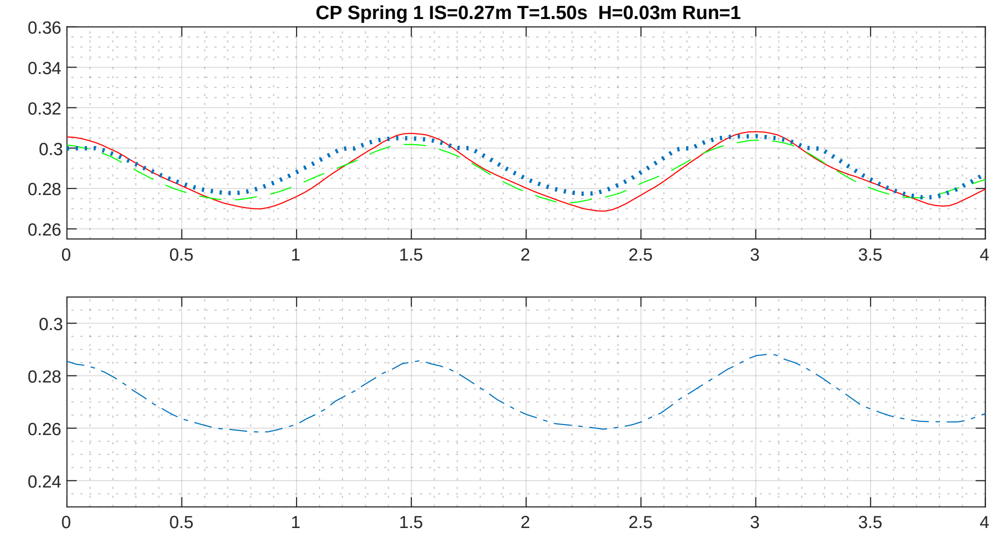

CP Spring 1 IS=0.27m T=1.25s H=0.01m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4

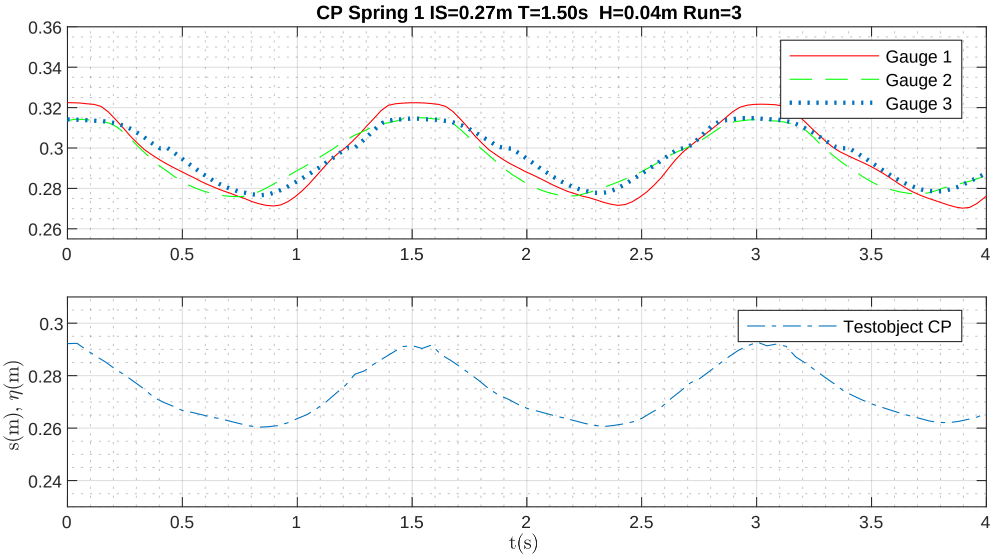

CP Spring 1 IS=0.27m T=1.25s H=0.02m Run=2
0.36
0.34
0.32
0.3
0.28
0.26
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4

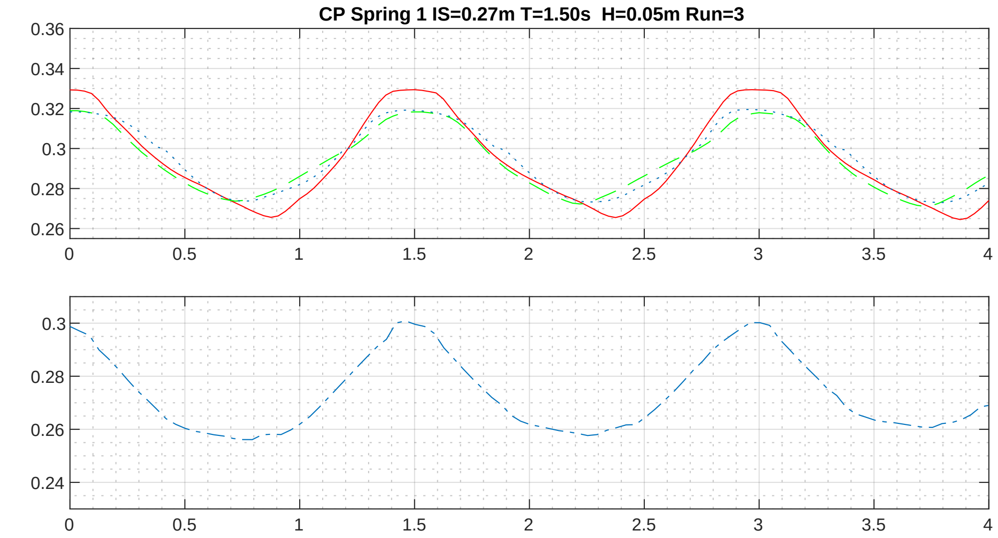

CP Spring 1 IS=0.27m T=1.25s H=0.03m Run=1
Gauge 1
Gauge 2
Gauge 3
Testobject CP
s(m), η(m)
t(s)
0.36
0.34
0.32
0.3
0.28
0.26
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4

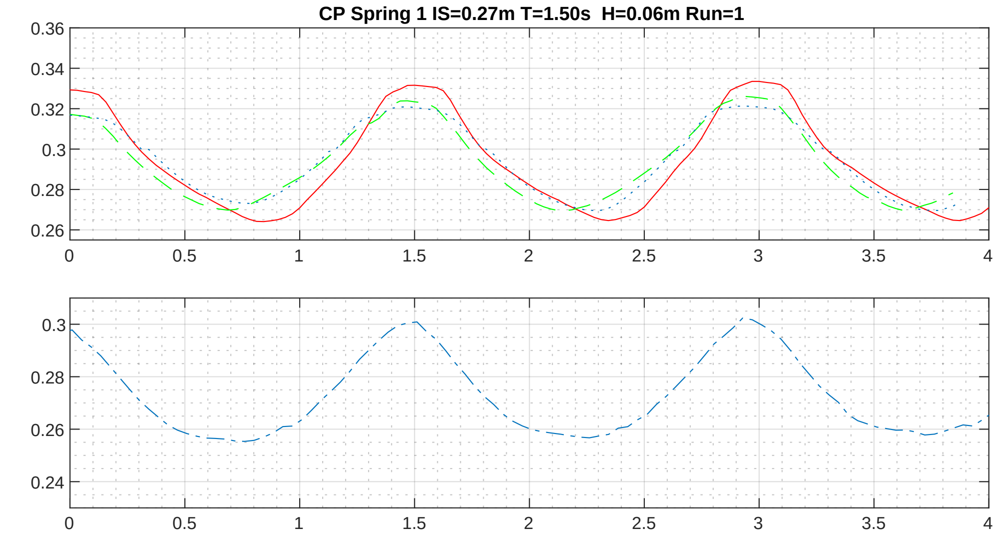

CP Spring 1 IS=0.27m T=1.25s H=0.04m Run=1
CP Spring 1 IS=0.27m T=1.25s H=0.05m Run=2
CP Spring 1 IS=0.27m T=1.25s H=0.06m Run=3
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Gauge 3
Testobject CP
s(m), η(m)
t(s)

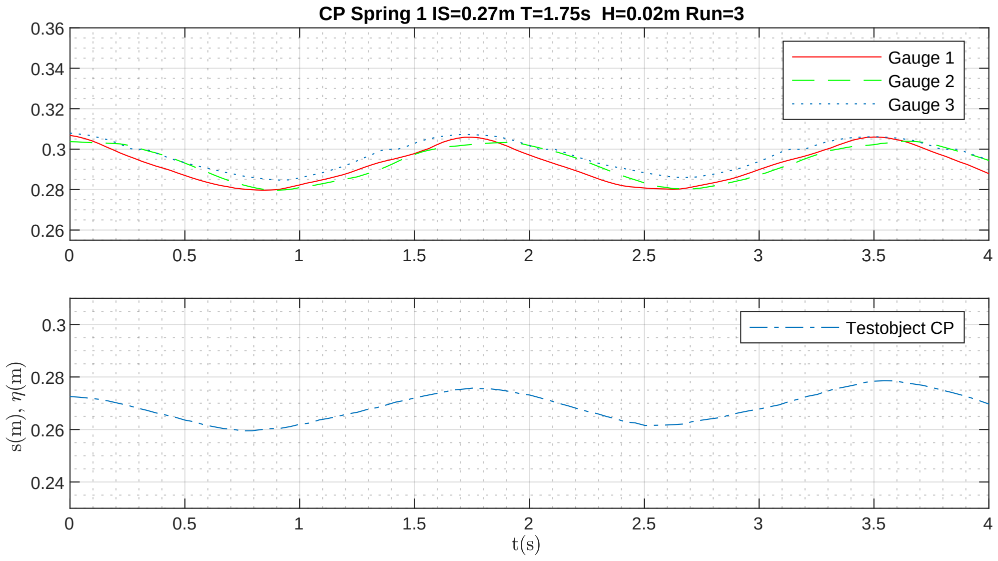

CP Spring 1 IS=0.27m T=1.50s H=0.02m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.3
0.28
0.26
0.24

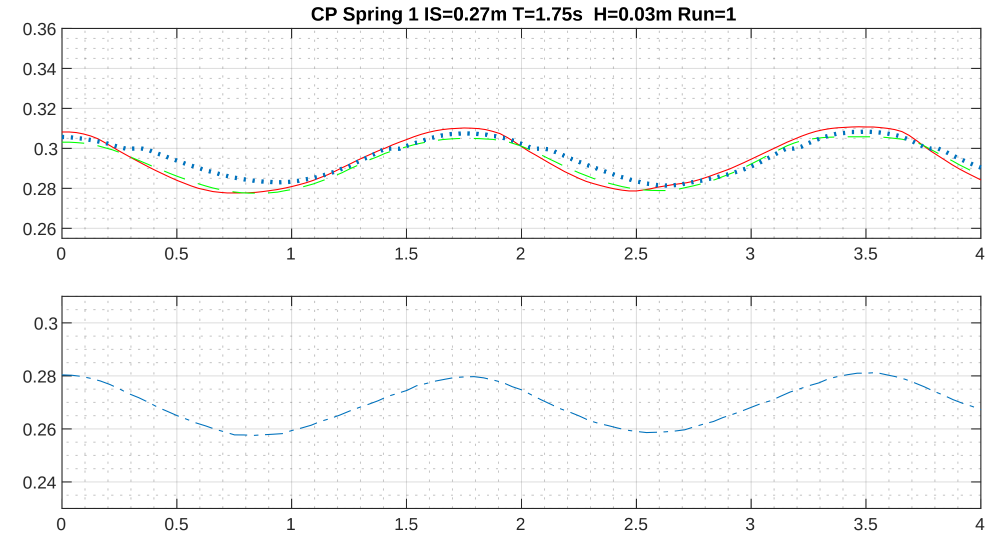

CP Spring 1 IS=0.27m T=1.50s H=0.03m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.3
0.28
0.26
0.24


CP Spring 1 IS=0.27m T=1.50s H=0.04m Run=3
Gauge 1
Gauge 2
Gauge 3
Testobject CP
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.3
0.28
0.26
0.24
s(m), η(m)
t(s)

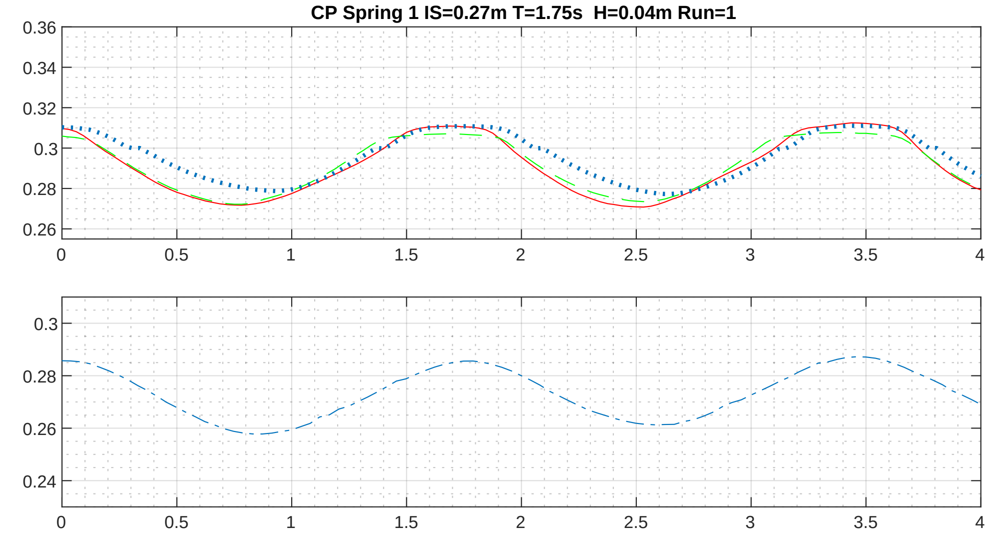

CP Spring 1 IS=0.27m T=1.50s H=0.05m Run=3


CP Spring 1 IS=0.27m T=1.50s H=0.06m Run=1

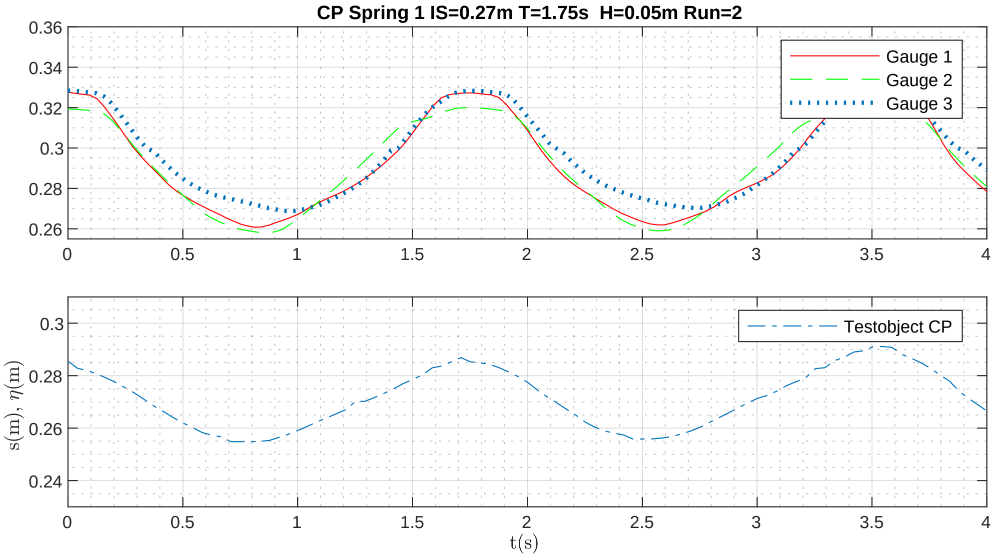

CP Spring 1 IS=0.27m T=1.75s H=0.02m Run=3



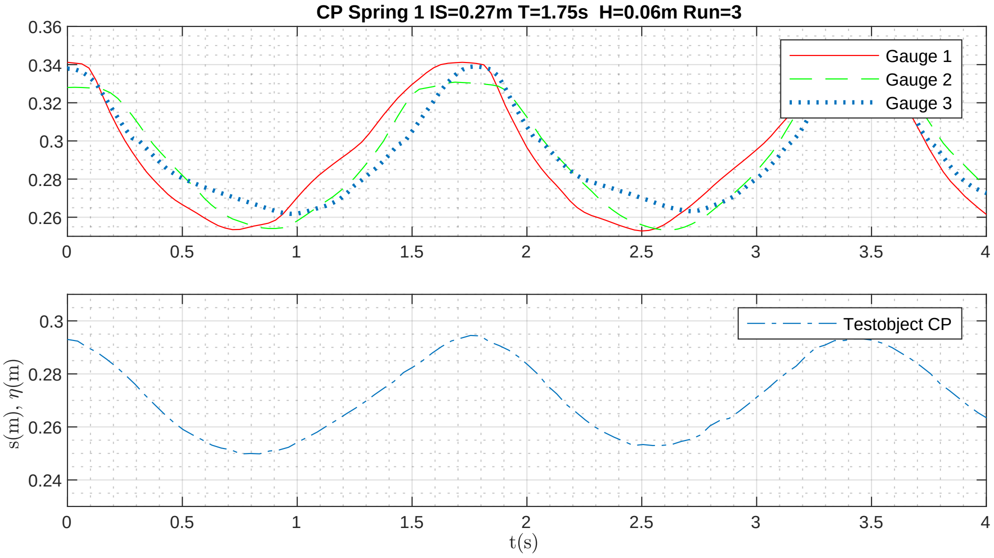
CP Spring 1 IS=0.27m T=1.75s H=0.03m Run=1

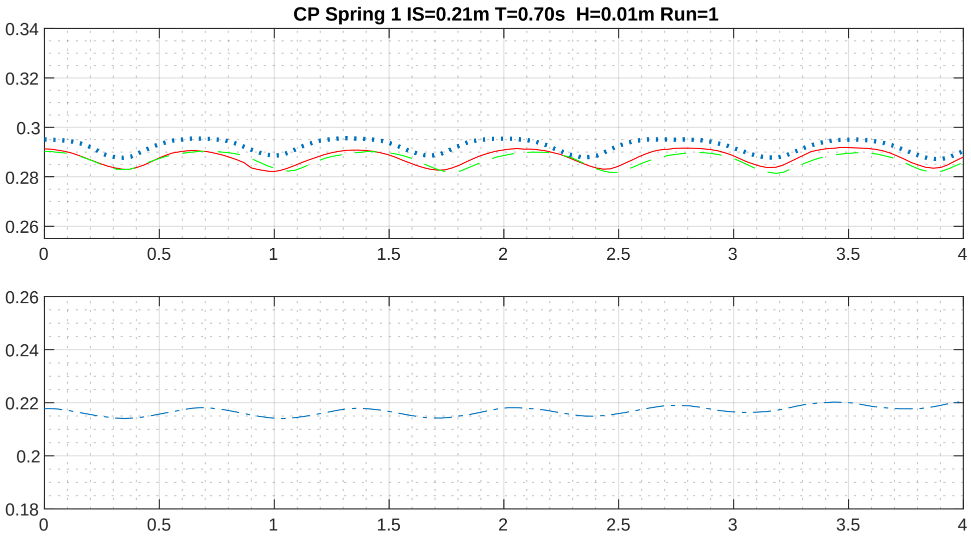


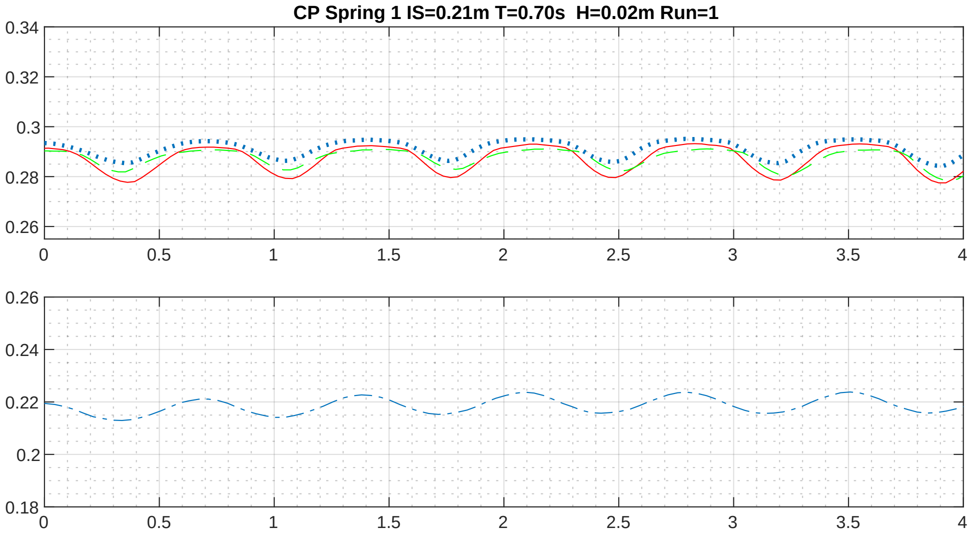
CP Spring 1 IS=0.27m T=1.75s H=0.04m Run=1

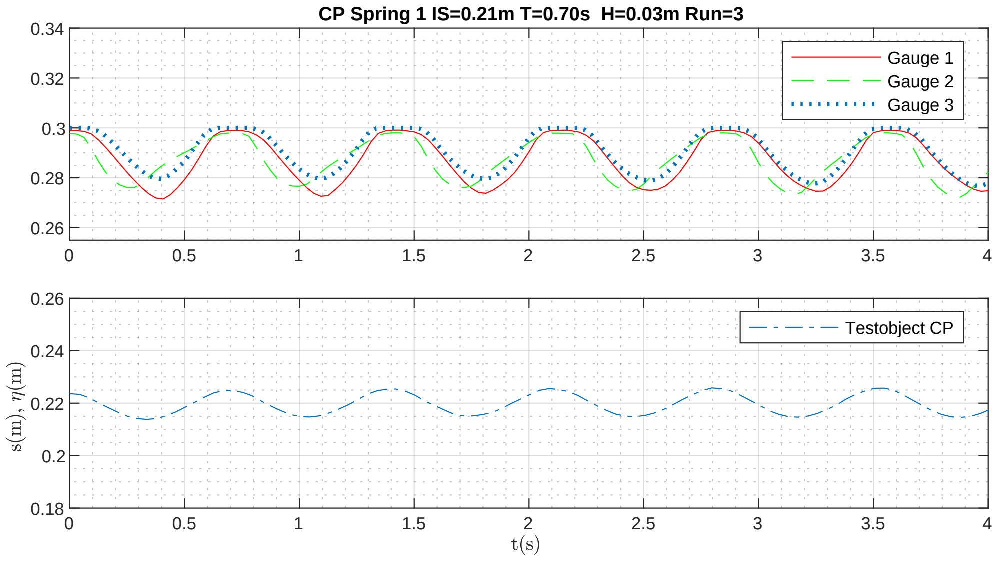


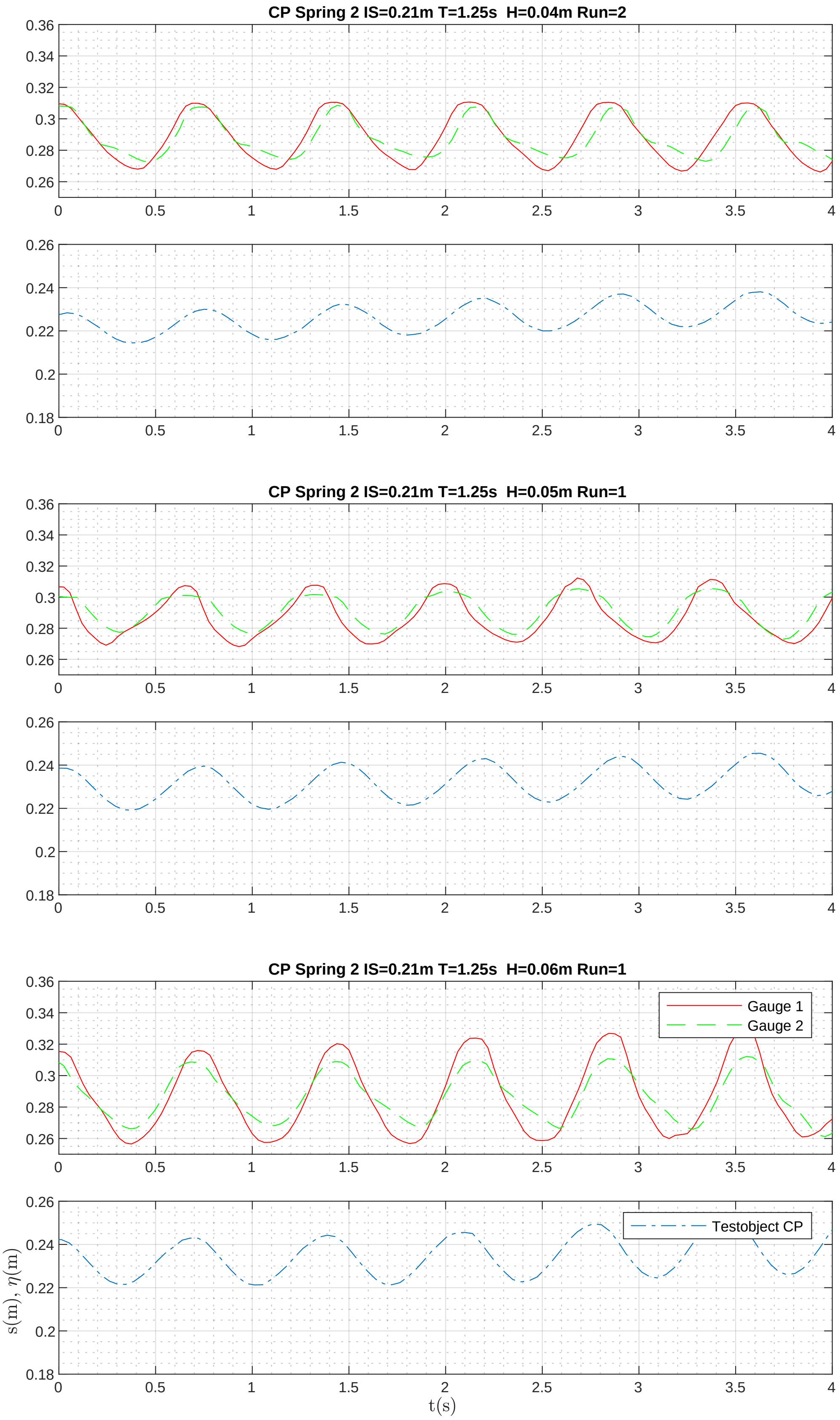
CP Spring 1 IS=0.27m T=1.75s H=0.05m Run=2

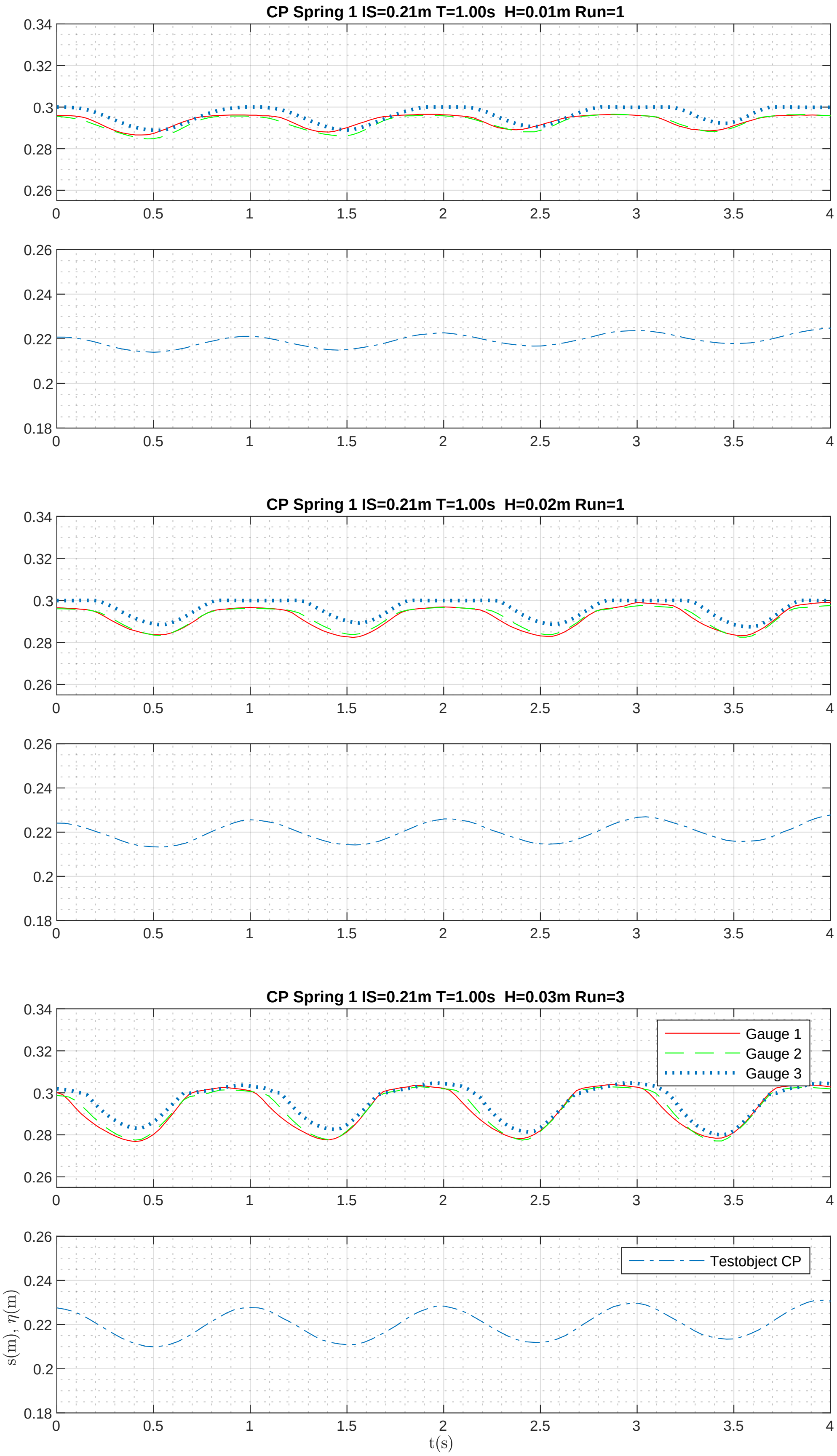

CP Spring 1 IS=0.27m T=1.75s H=0.06m Run=3
Gauge 1
Gauge 2
Gauge 3
Testobject CP
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4
s(m), η(m)
t(s)

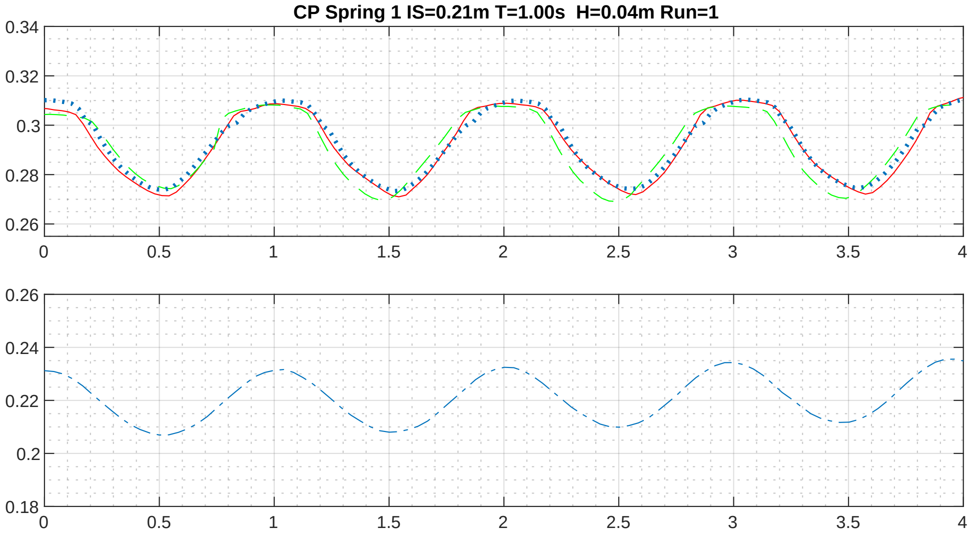

CP Spring 1 IS=0.21m T=0.70s H=0.01m Run=1
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

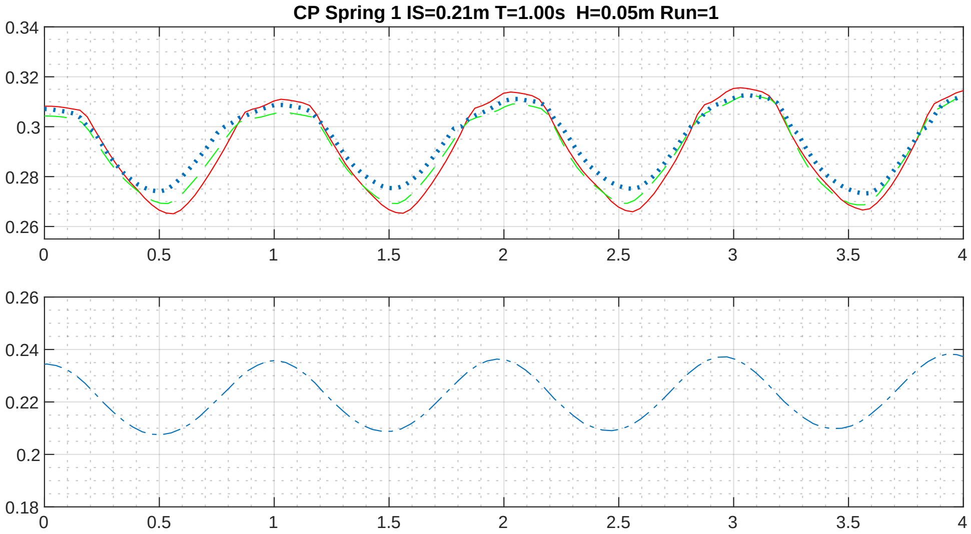

CP Spring 1 IS=0.21m T=0.70s H=0.02m Run=1
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

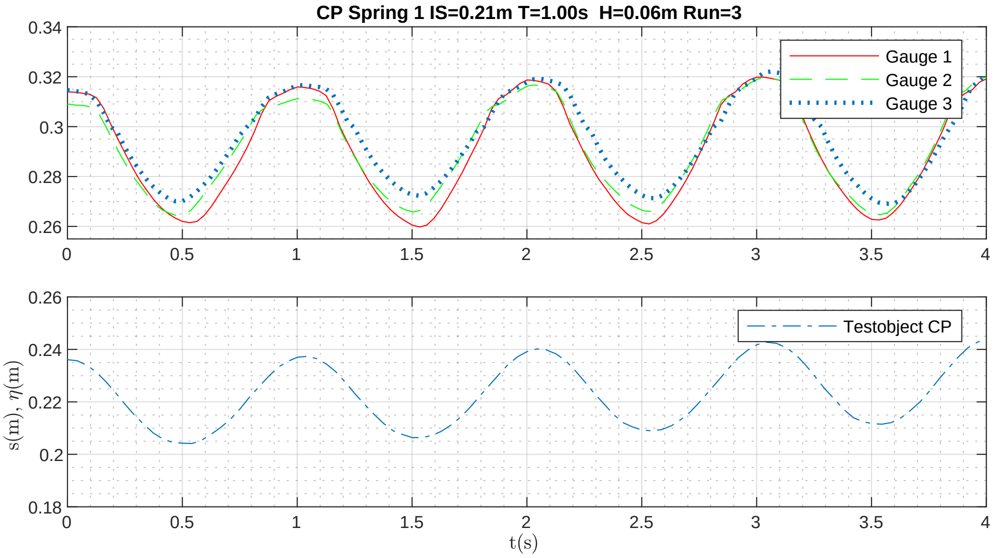

CP Spring 1 IS=0.21m T=0.70s H=0.03m Run=3
Gauge 1
Gauge 2
Gauge 3
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
Testobject CP
0.26
0.24
0.22
0.2
0.18
s(m), η(m)
0
0.5
1
1.5
2
2.5
3
3.5
4
t(s)

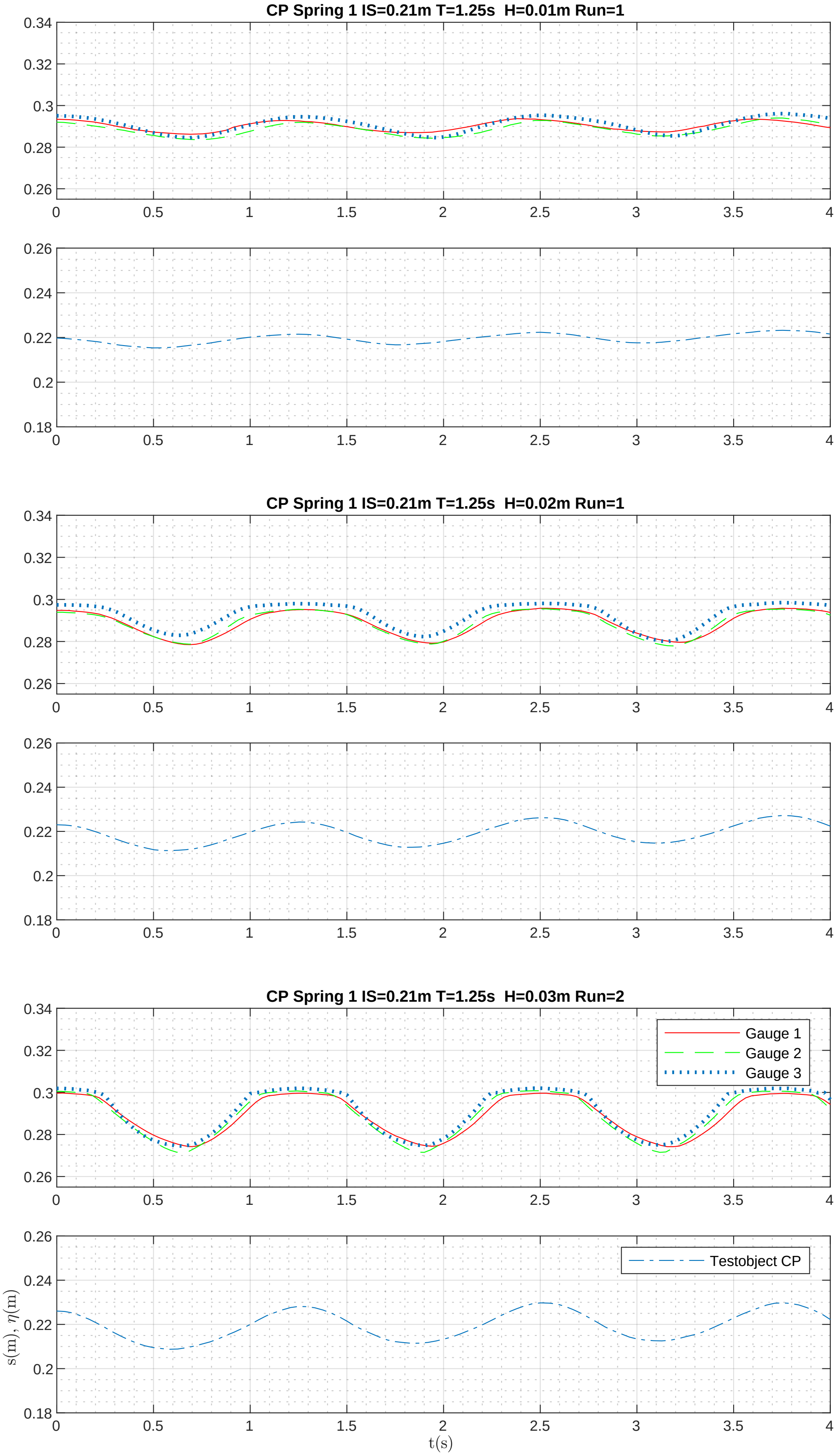

CP Spring 2 IS=0.21m T=1.25s H=0.04m Run=2
CP Spring 2 IS=0.21m T=1.25s H=0.05m Run=1
CP Spring 2 IS=0.21m T=1.25s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

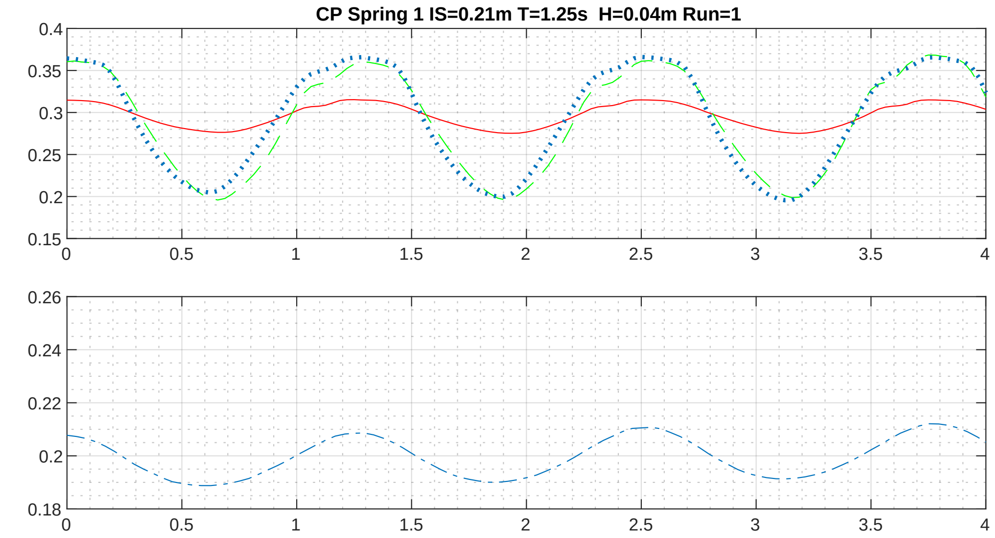

CP Spring 1 IS=0.21m T=1.00s H=0.01m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4
CP Spring 1 IS=0.21m T=1.00s H=0.02m Run=1
CP Spring 1 IS=0.21m T=1.00s H=0.03m Run=3
Gauge 1
Gauge 2
Gauge 3
Testobject CP
s(m), η(m)
t(s)

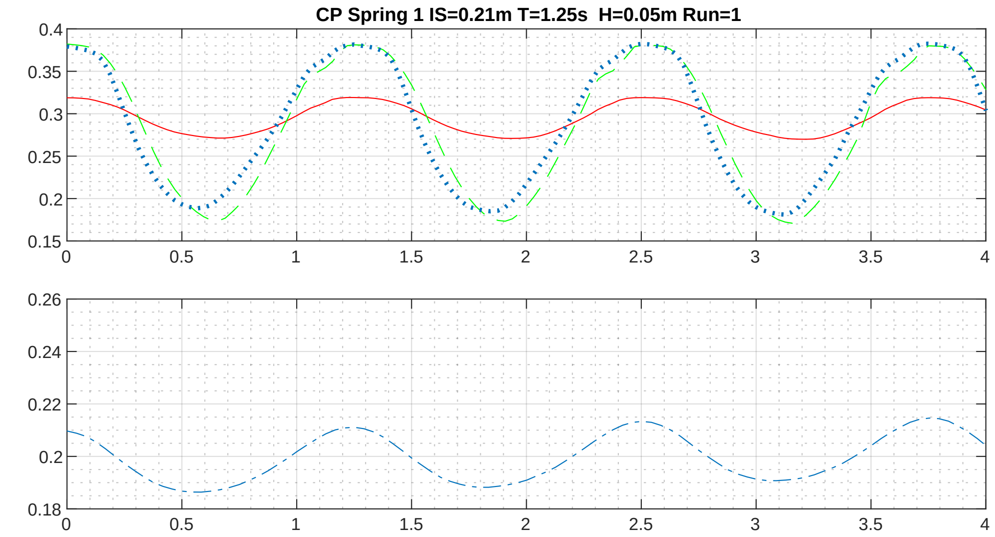

CP Spring 1 IS=0.21m T=1.00s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

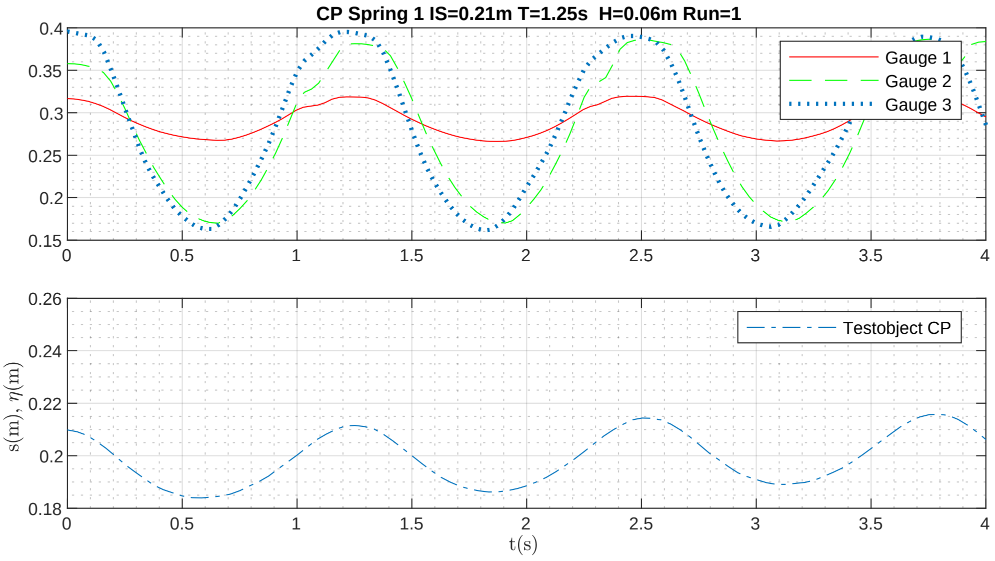

CP Spring 1 IS=0.21m T=1.00s H=0.05m Run=1
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

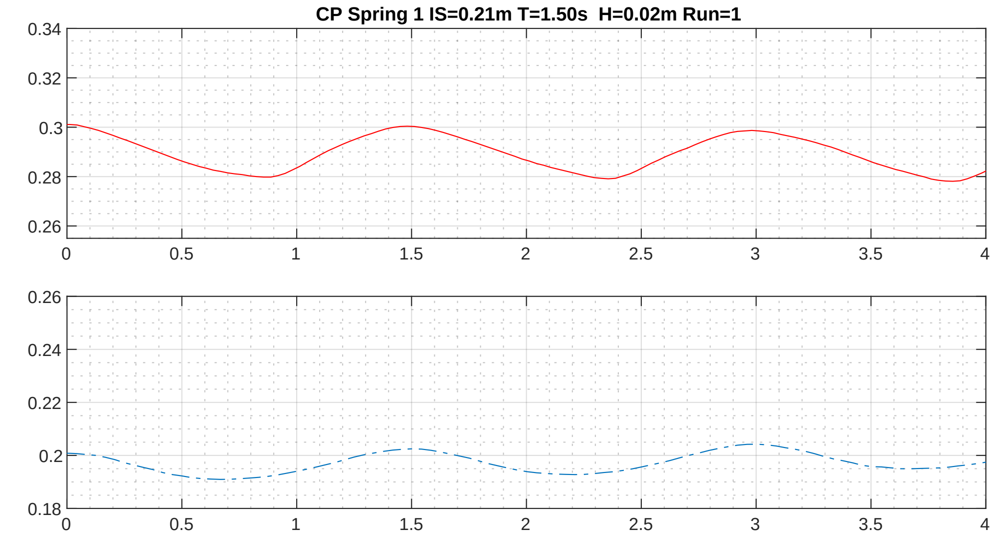

CP Spring 1 IS=0.21m T=1.00s H=0.06m Run=3
Gauge 1
Gauge 2
Gauge 3
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
Testobject CP
s(m), η(m)
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4
t(s)

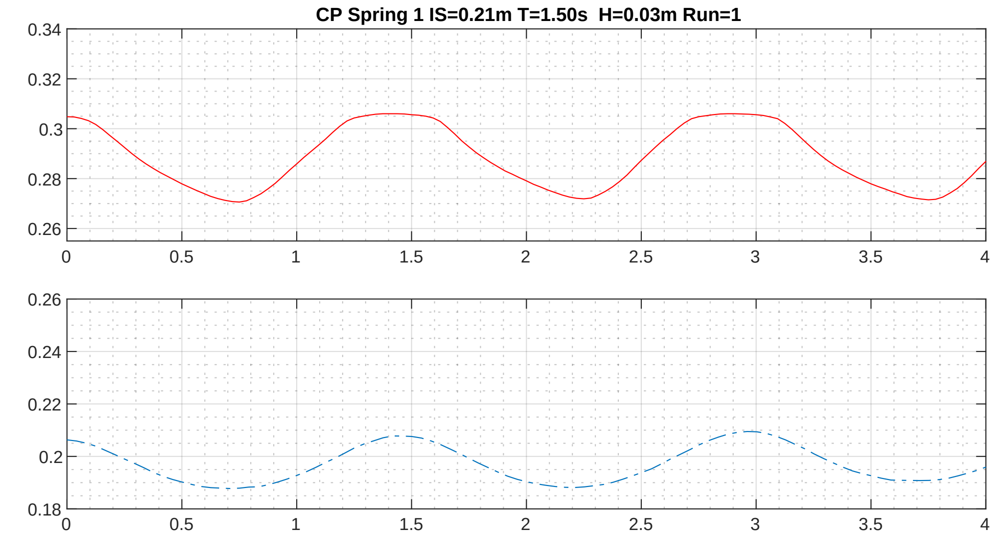

CP Spring 1 IS=0.21m T=1.25s H=0.01m Run=1
CP Spring 1 IS=0.21m T=1.25s H=0.02m Run=1
CP Spring 1 IS=0.21m T=1.25s H=0.03m Run=2
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Gauge 3
Testobject CP
s(m), η(m)
t(s)

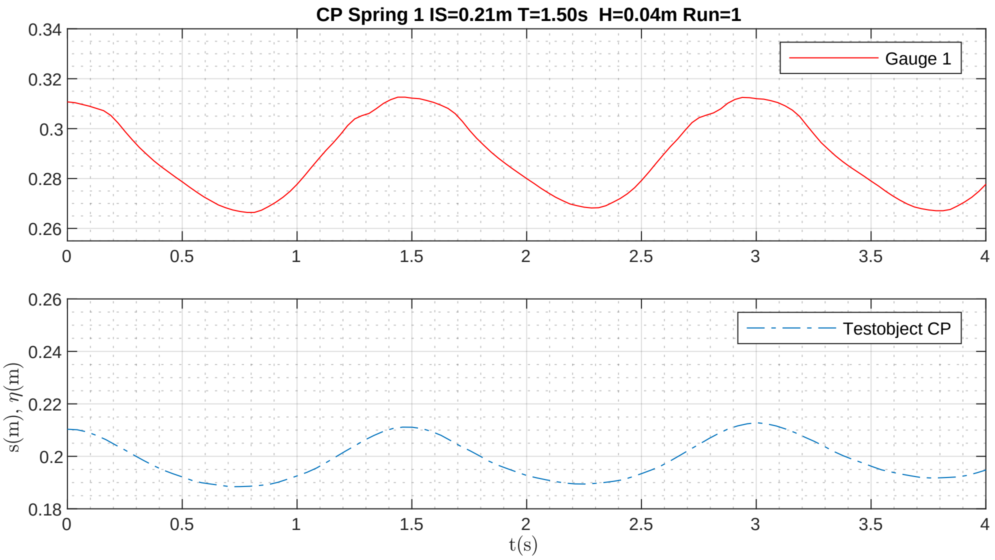

CP Spring 1 IS=0.21m T=1.25s H=0.04m Run=1
0.4
0.35
0.3
0.25
0.2
0.15
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

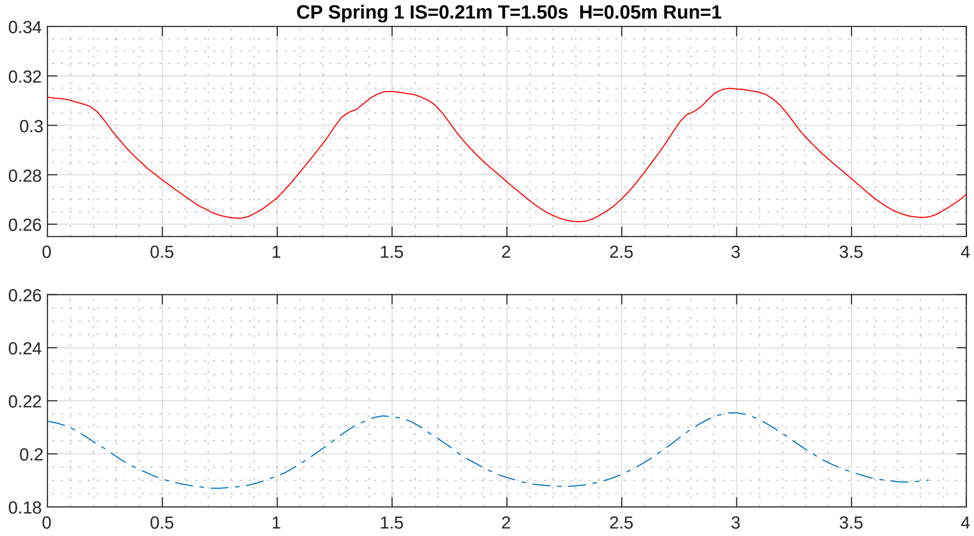

CP Spring 1 IS=0.21m T=1.25s H=0.05m Run=1
0.4
0.35
0.3
0.25
0.2
0.15
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

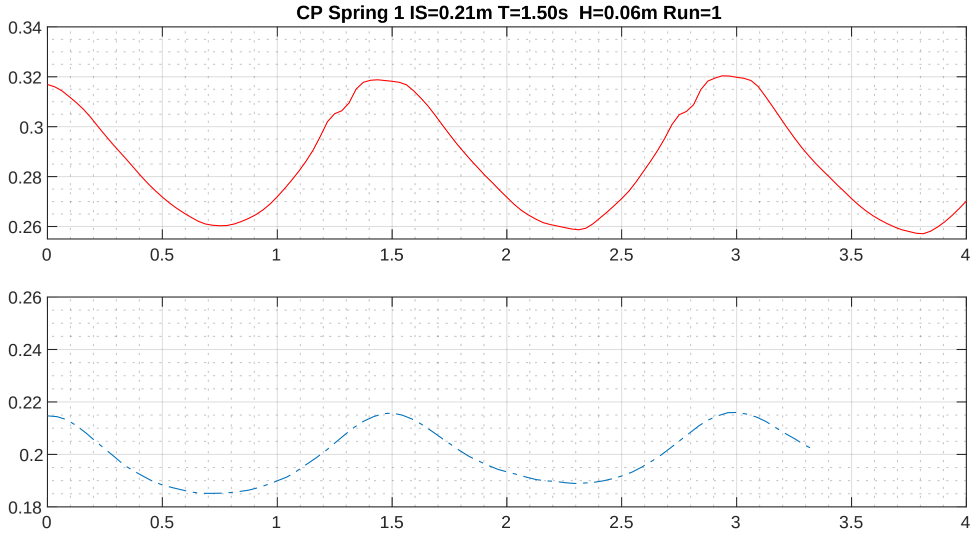

CP Spring 1 IS=0.21m T=1.25s H=0.06m Run=1
Gauge 1
Gauge 2
Gauge 3
Testobject CP
s(m), η(m)
t(s)
0.4
0.35
0.3
0.25
0.2
0.15
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

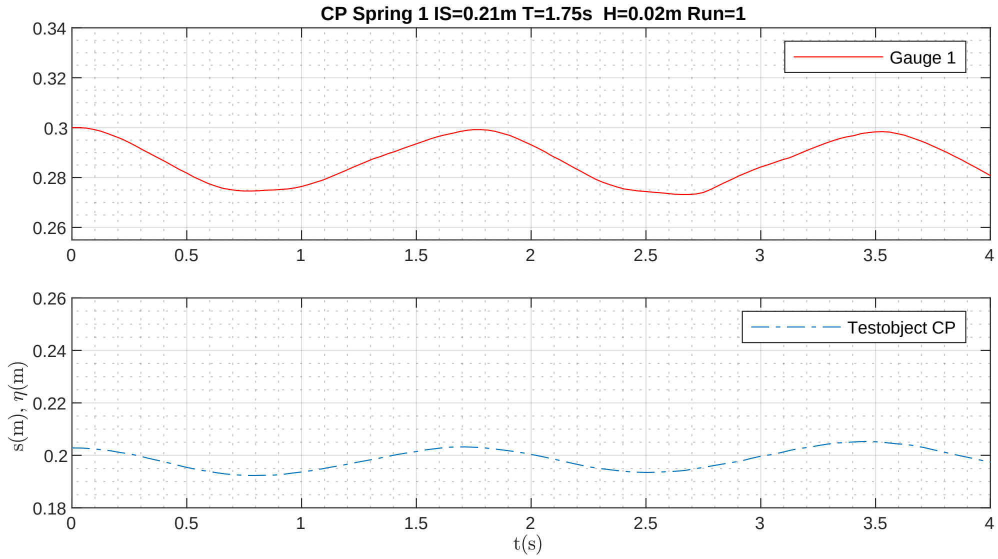

CP Spring 1 IS=0.21m T=1.50s H=0.02m Run=1

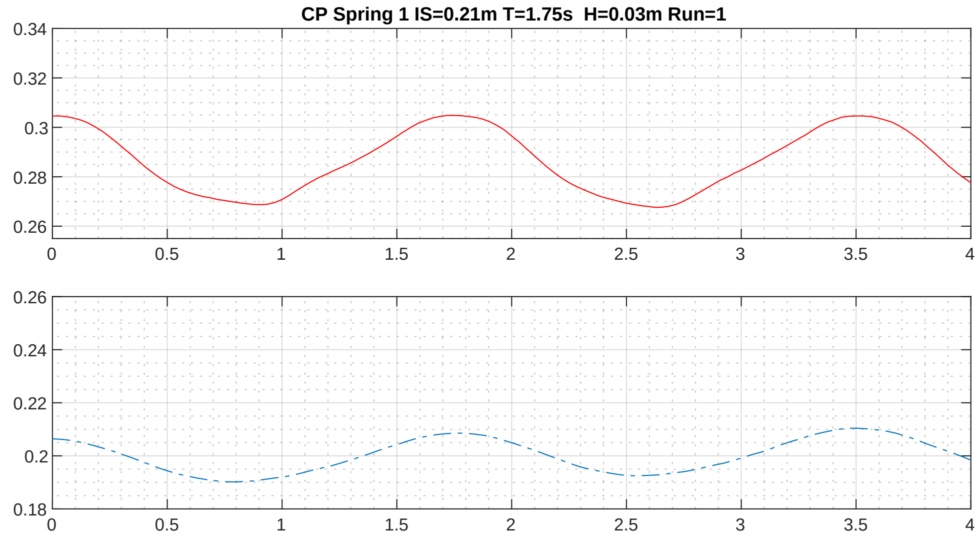

CP Spring 1 IS=0.21m T=1.50s H=0.03m Run=1

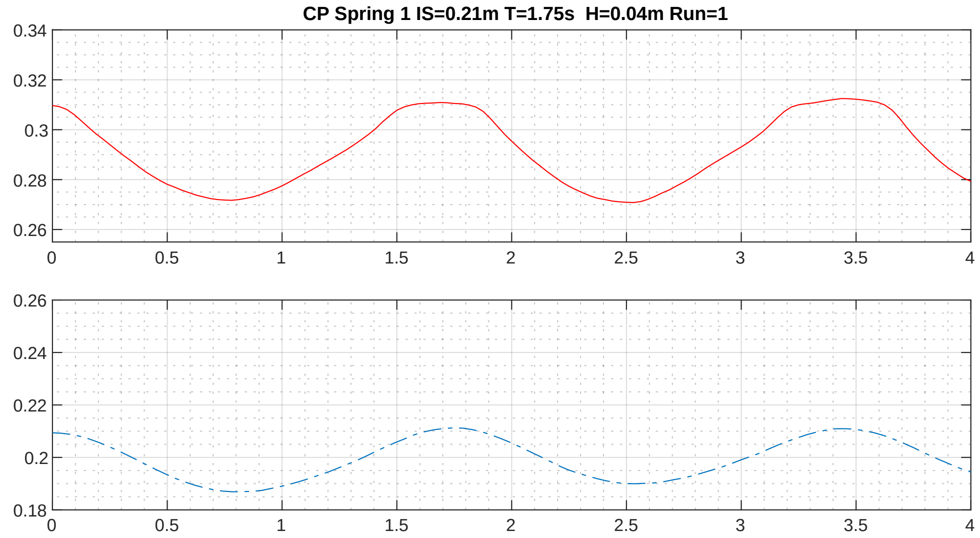

CP Spring 1 IS=0.21m T=1.50s H=0.04m Run=1

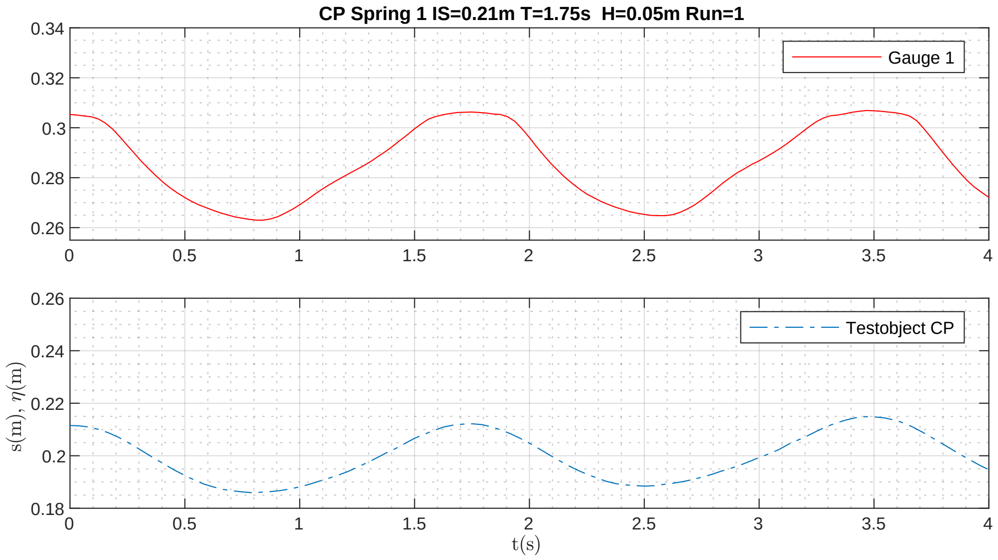

CP Spring 1 IS=0.21m T=1.50s H=0.05m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

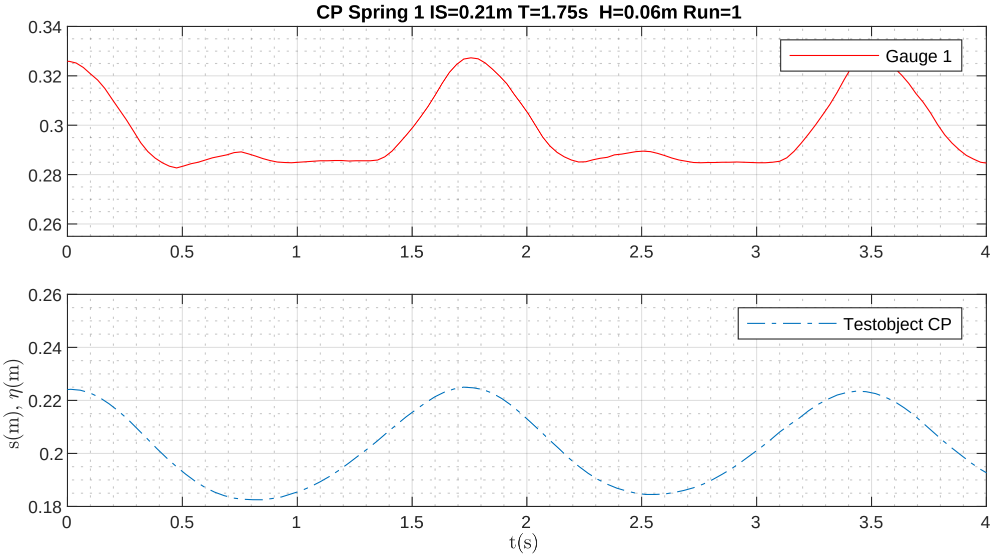

CP Spring 1 IS=0.21m T=1.50s H=0.06m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

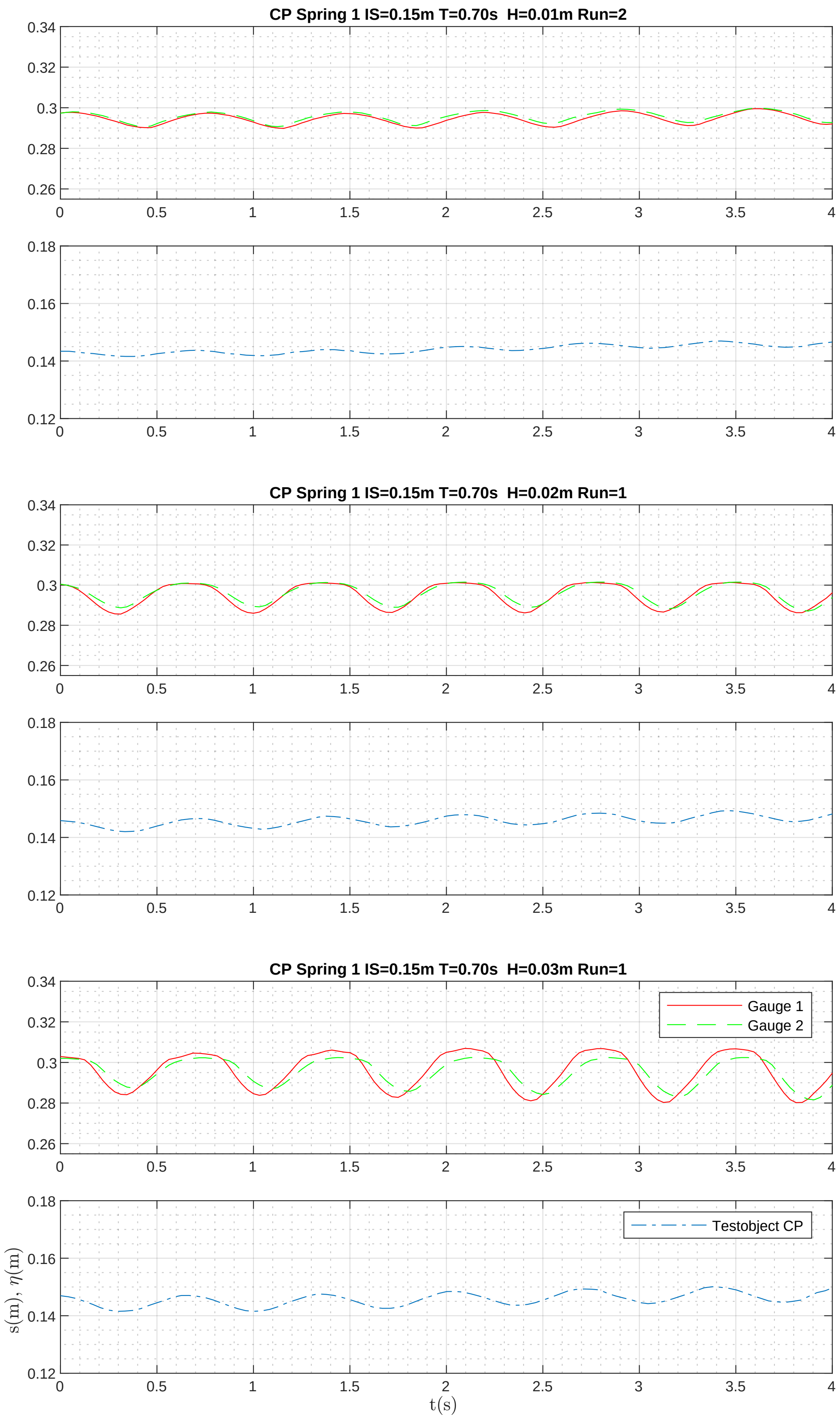

CP Spring 1 IS=0.21m T=1.75s H=0.02m Run=1
Gauge 1
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

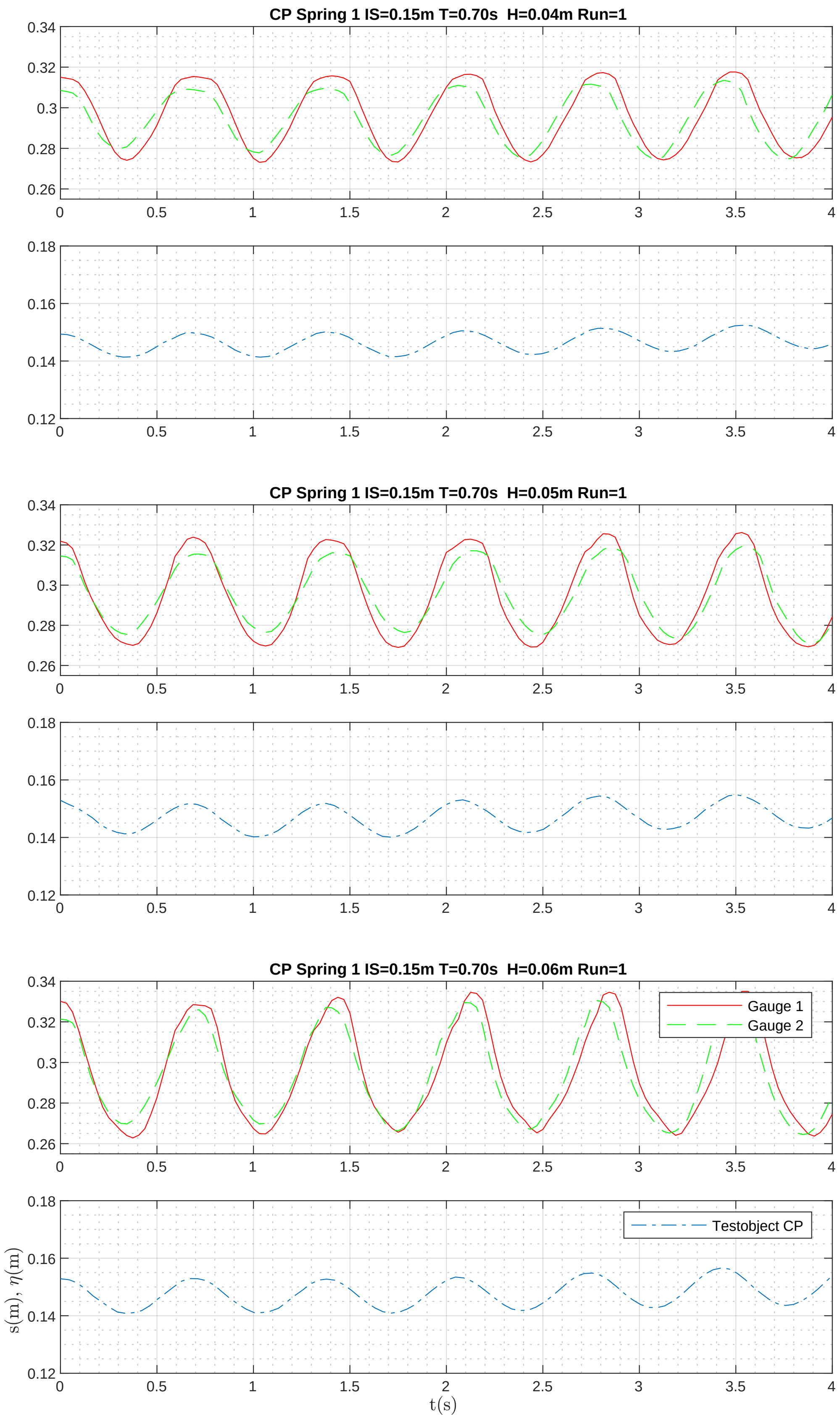

CP Spring 1 IS=0.21m T=1.75s H=0.03m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

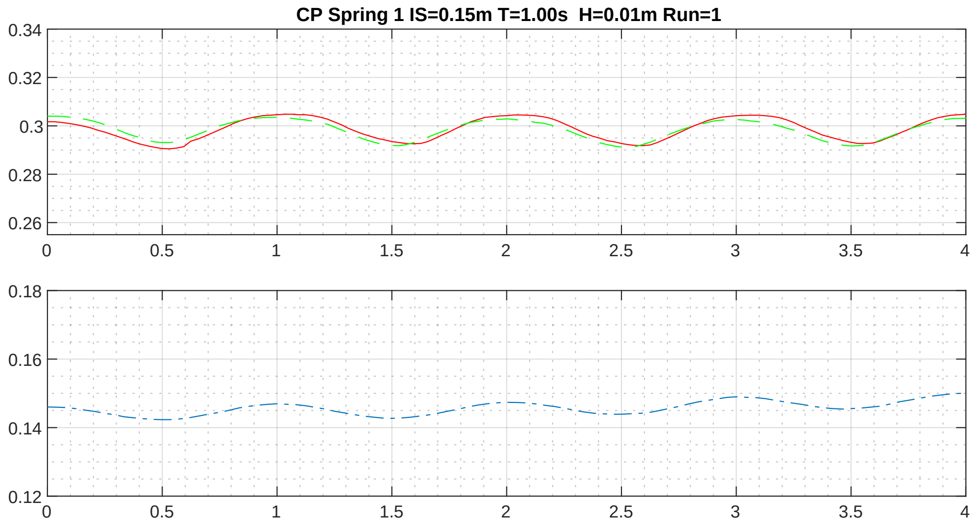

CP Spring 1 IS=0.21m T=1.75s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

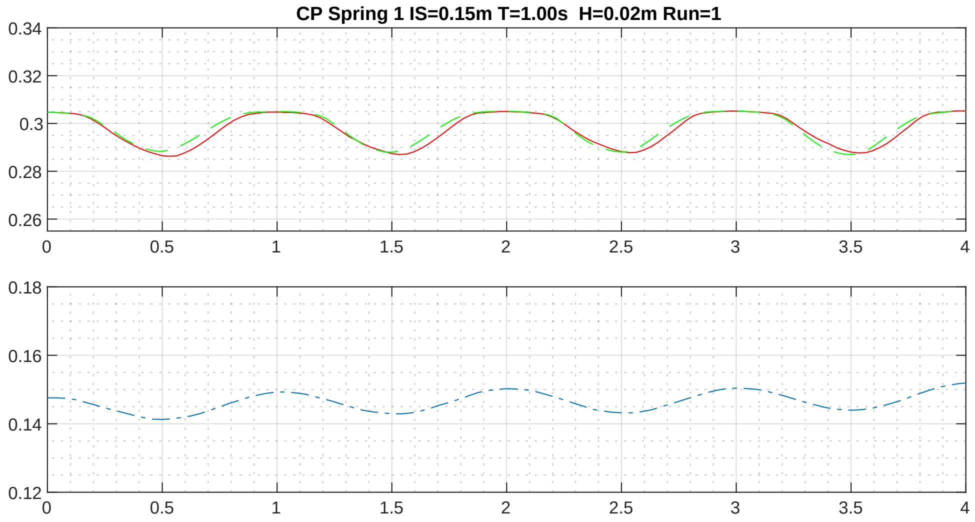

CP Spring 1 IS=0.21m T=1.75s H=0.05m Run=1
Gauge 1
Testobject CP
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4
s(m), η(m)
t(s)

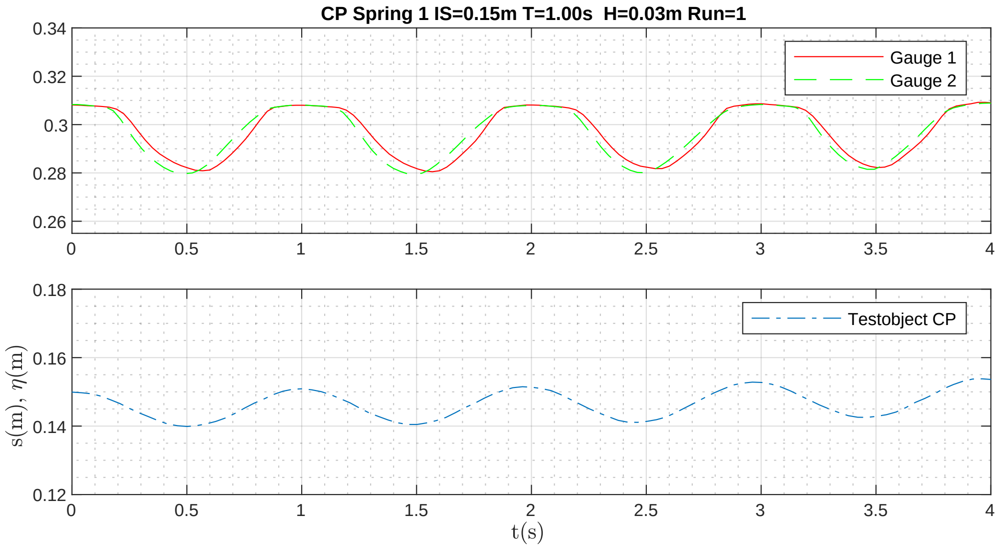

CP Spring 1 IS=0.21m T=1.75s H=0.06m Run=1
Gauge 1
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
Testobject CP
0.26
0.24
0.22
0.2
0.18
s(m), η(m)
t(s)

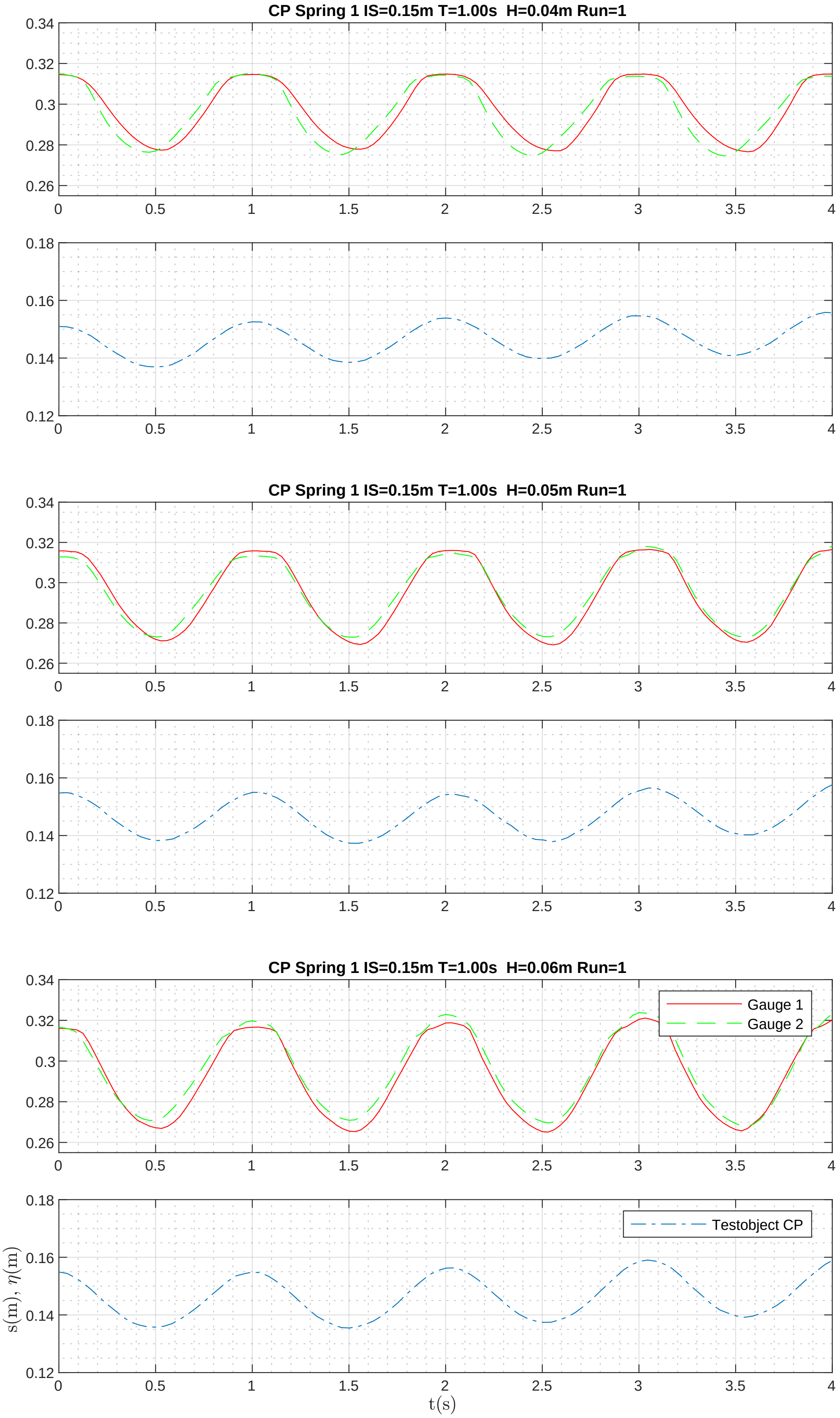

CP Spring 1 IS=0.15m T=0.70s H=0.01m Run=2
CP Spring 1 IS=0.15m T=0.70s H=0.02m Run=1
CP Spring 1 IS=0.15m T=0.70s H=0.03m Run=1
Gauge 1
Gauge 2
Testobject CP
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4
s(m), η(m)
t(s)

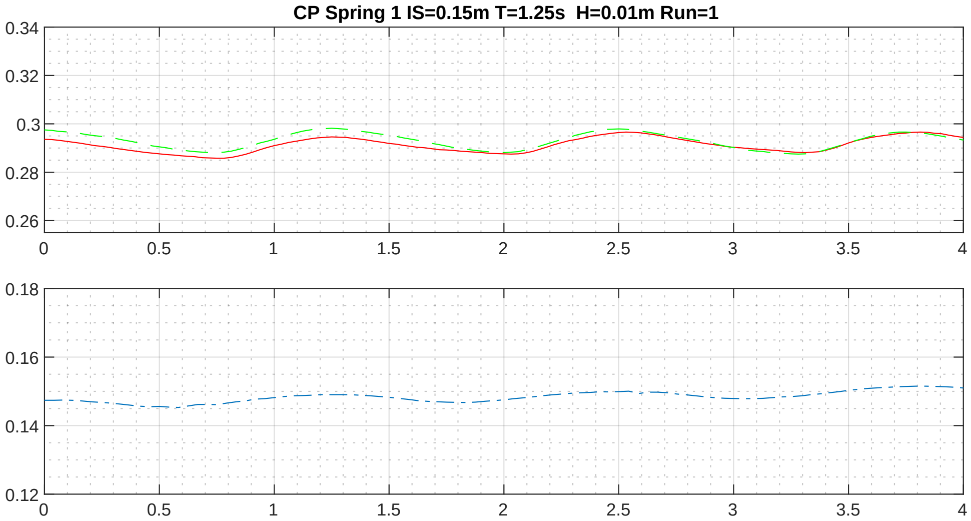

CP Spring 1 IS=0.15m T=0.70s H=0.04m Run=1
CP Spring 1 IS=0.15m T=0.70s H=0.05m Run=1
CP Spring 1 IS=0.15m T=0.70s H=0.06m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

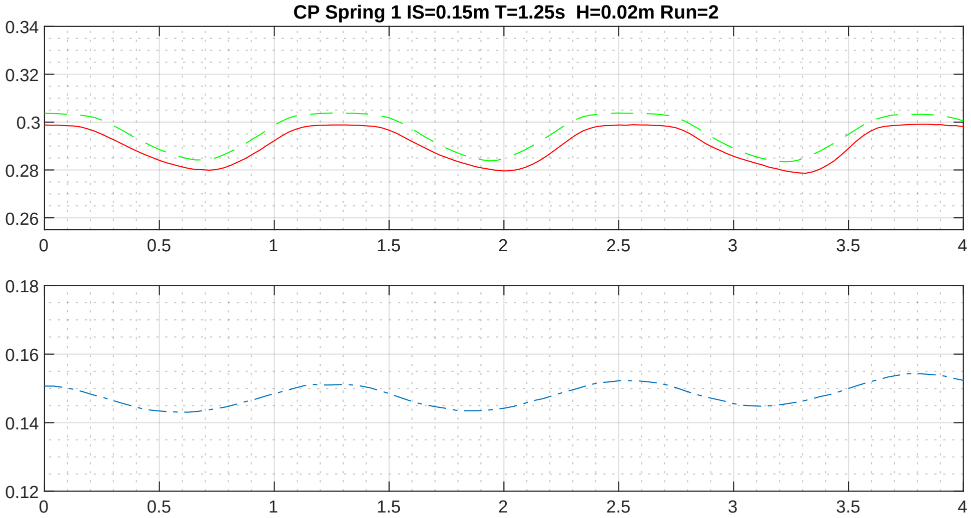

CP Spring 1 IS=0.15m T=1.00s H=0.01m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

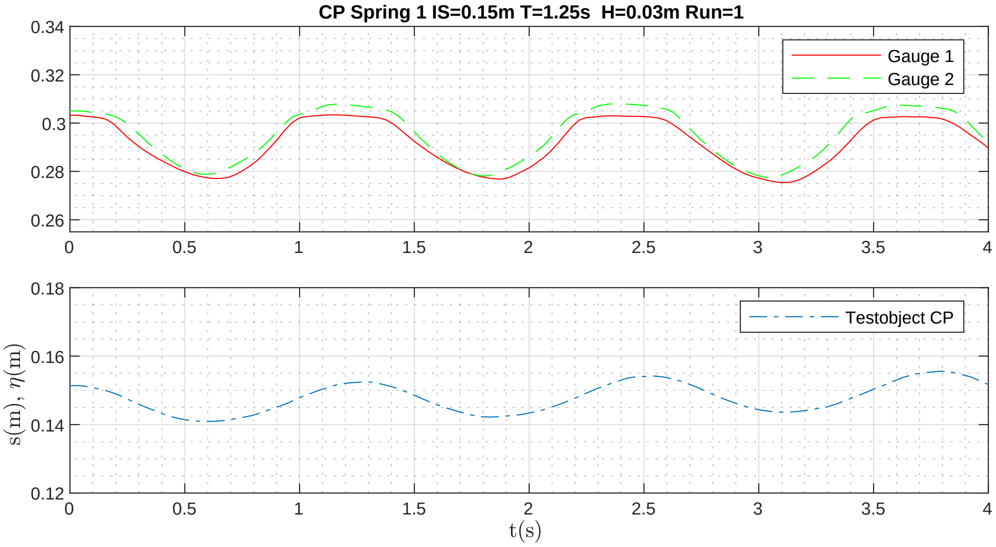

CP Spring 1 IS=0.15m T=1.00s H=0.02m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

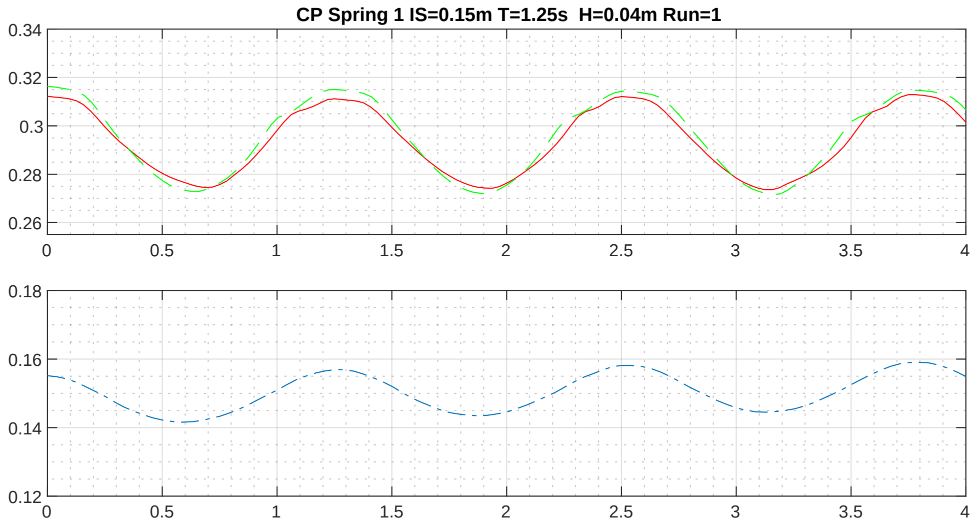

CP Spring 1 IS=0.15m T=1.00s H=0.03m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

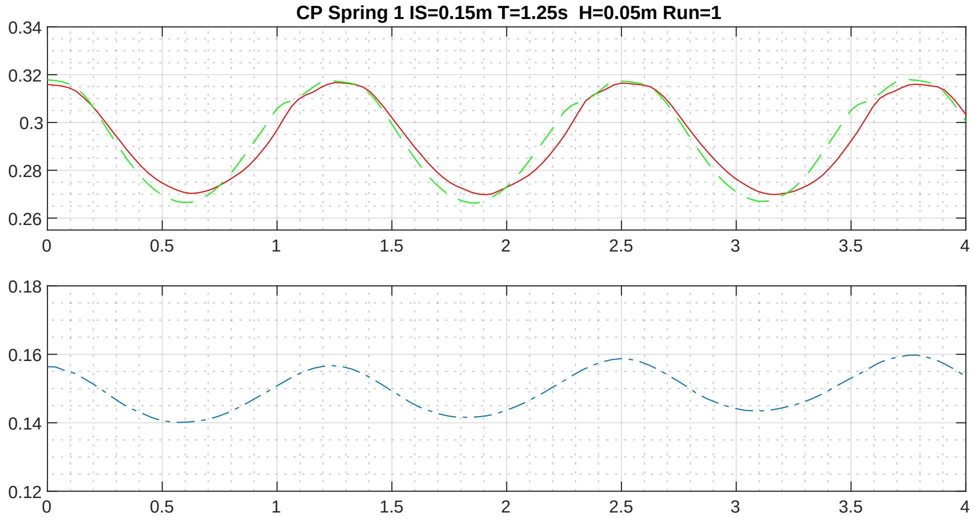

CP Spring 1 IS=0.15m T=1.00s H=0.04m Run=1
CP Spring 1 IS=0.15m T=1.00s H=0.05m Run=1
CP Spring 1 IS=0.15m T=1.00s H=0.06m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

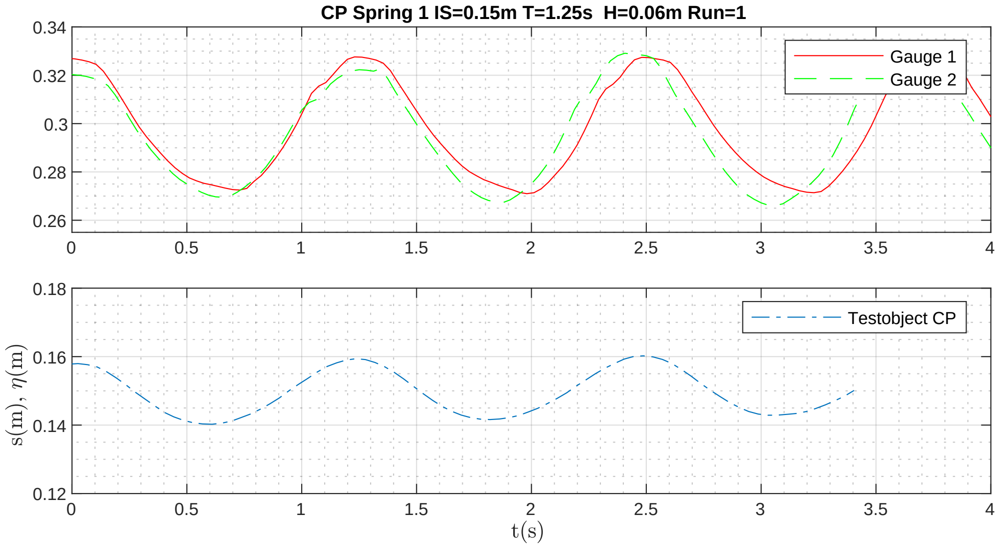

CP Spring 1 IS=0.15m T=1.25s H=0.01m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

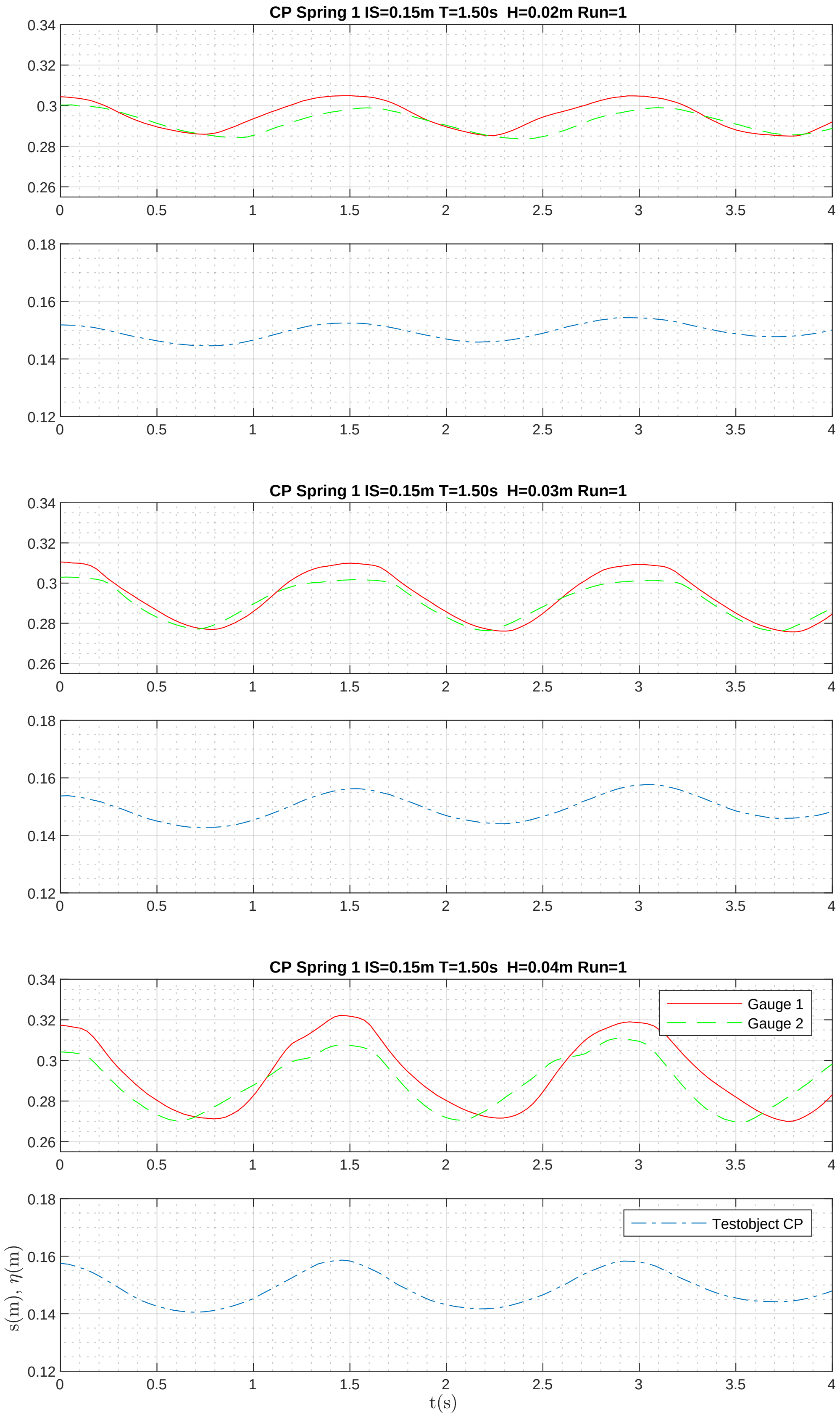

CP Spring 1 IS=0.15m T=1.25s H=0.02m Run=2
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

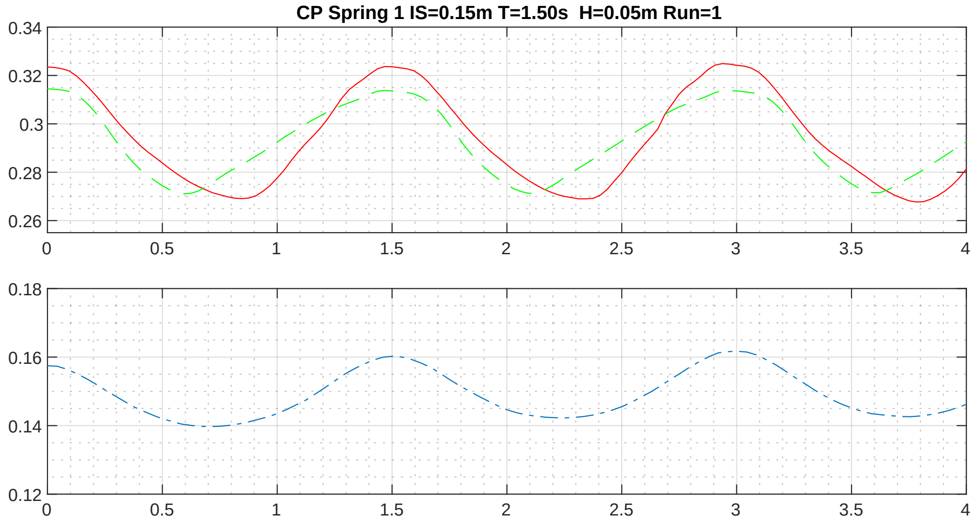

CP Spring 1 IS=0.15m T=1.25s H=0.03m Run=1
Gauge 1
Gauge 2
Testobject CP
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
s(m), η(m)
0
0.5
1
1.5
2
2.5
3
3.5
4
t(s)

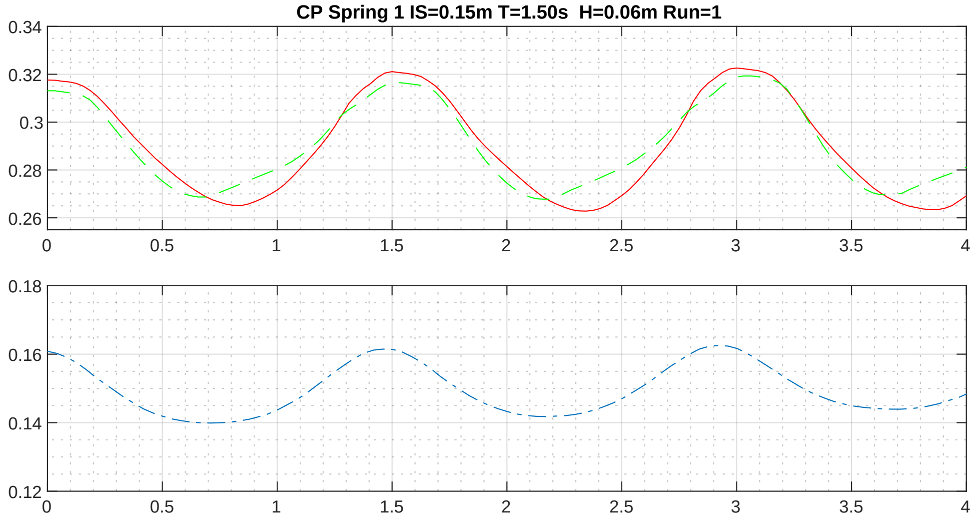

CP Spring 1 IS=0.15m T=1.25s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

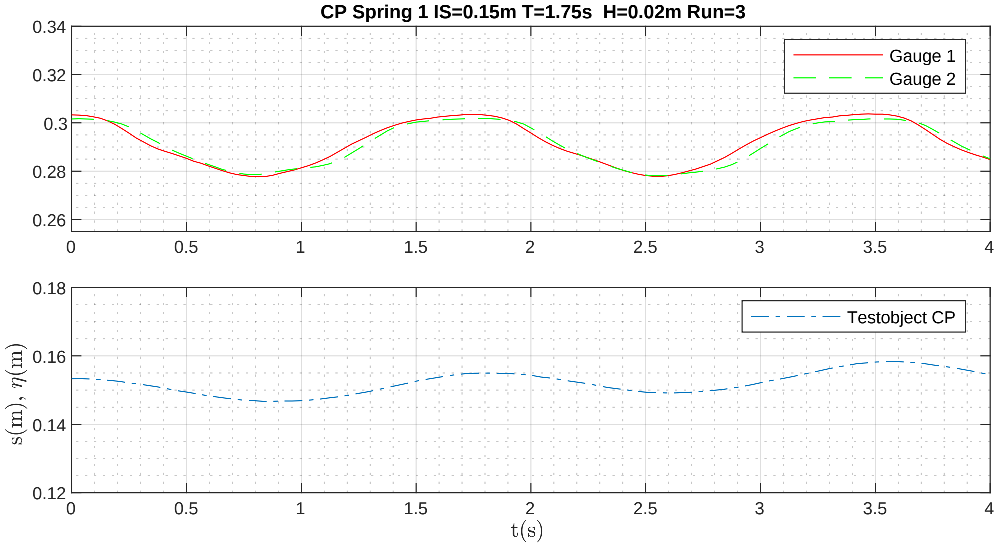

CP Spring 1 IS=0.15m T=1.25s H=0.05m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

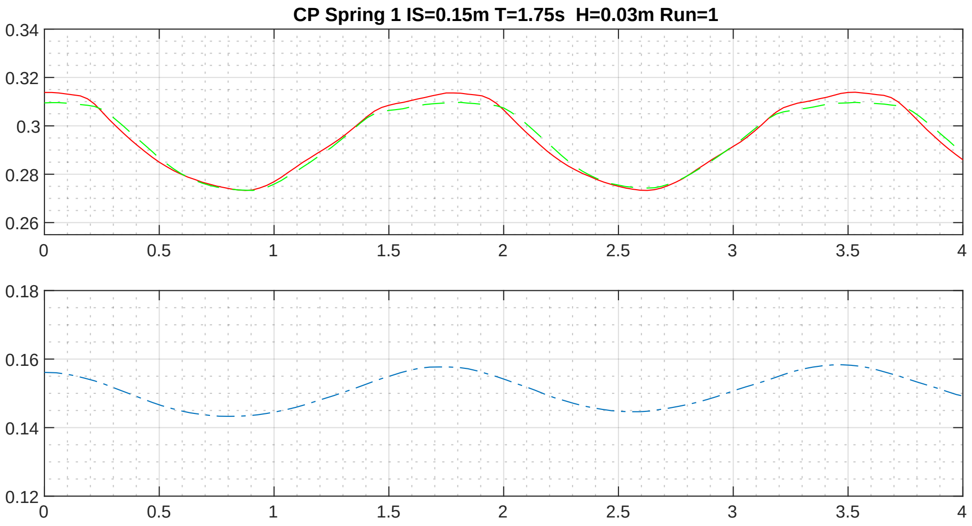

CP Spring 1 IS=0.15m T=1.25s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject CP
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
s(m), η(m)
0
0.5
1
1.5
2
2.5
3
3.5
4
t(s)

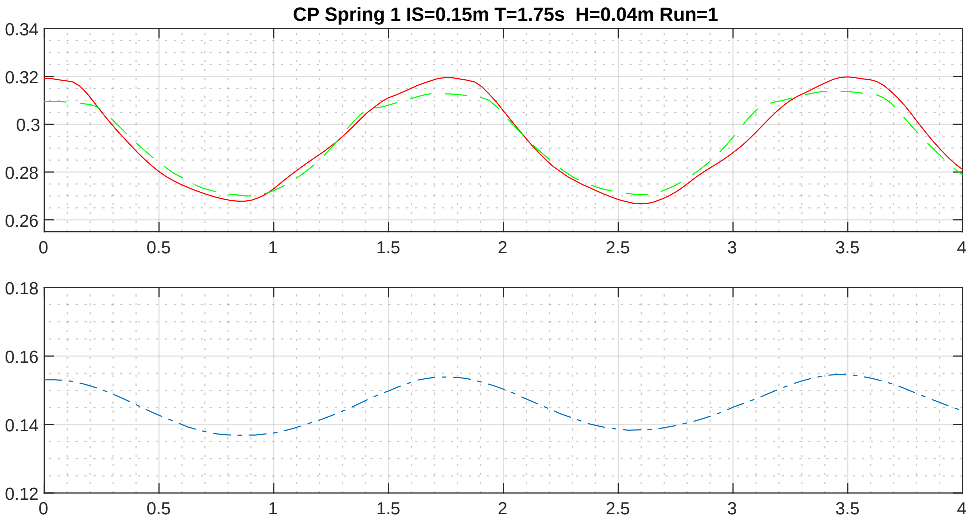

CP Spring 1 IS=0.15m T=1.50s H=0.02m Run=1
CP Spring 1 IS=0.15m T=1.50s H=0.03m Run=1
CP Spring 1 IS=0.15m T=1.50s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

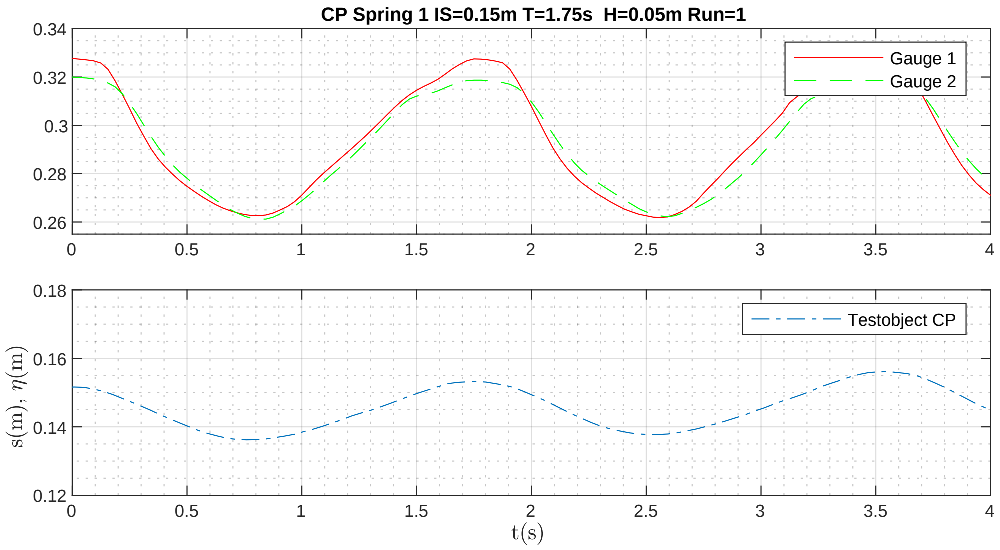

CP Spring 1 IS=0.15m T=1.50s H=0.05m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

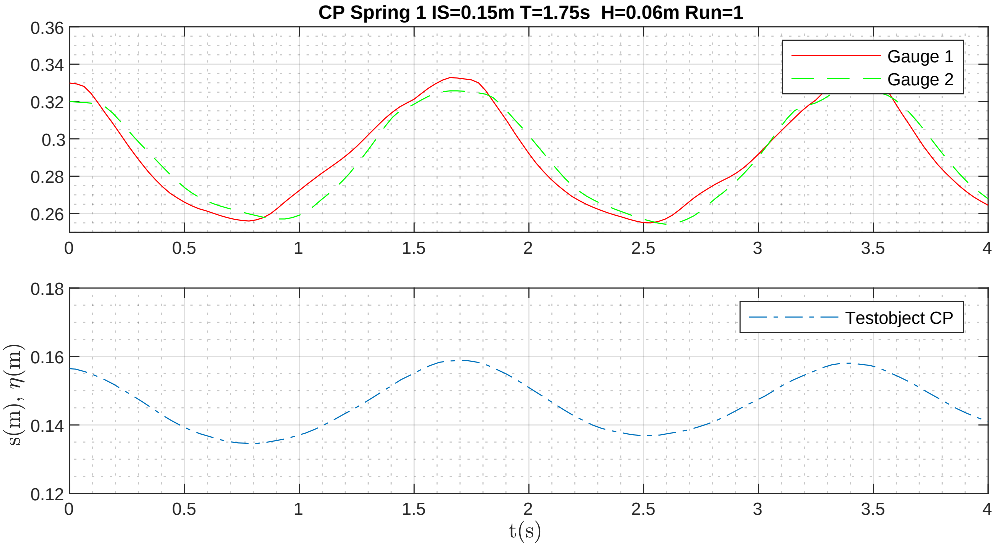

CP Spring 1 IS=0.15m T=1.50s H=0.06m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

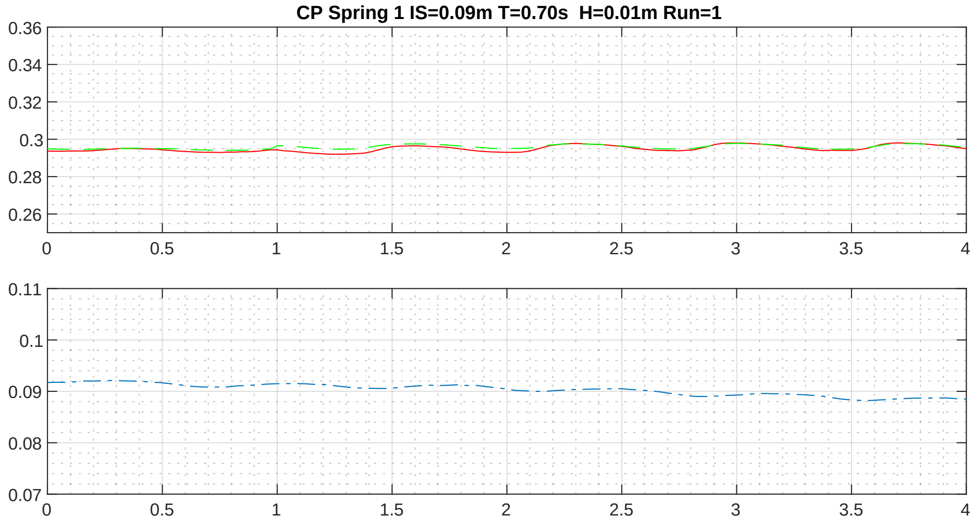

CP Spring 1 IS=0.15m T=1.75s H=0.02m Run=3
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

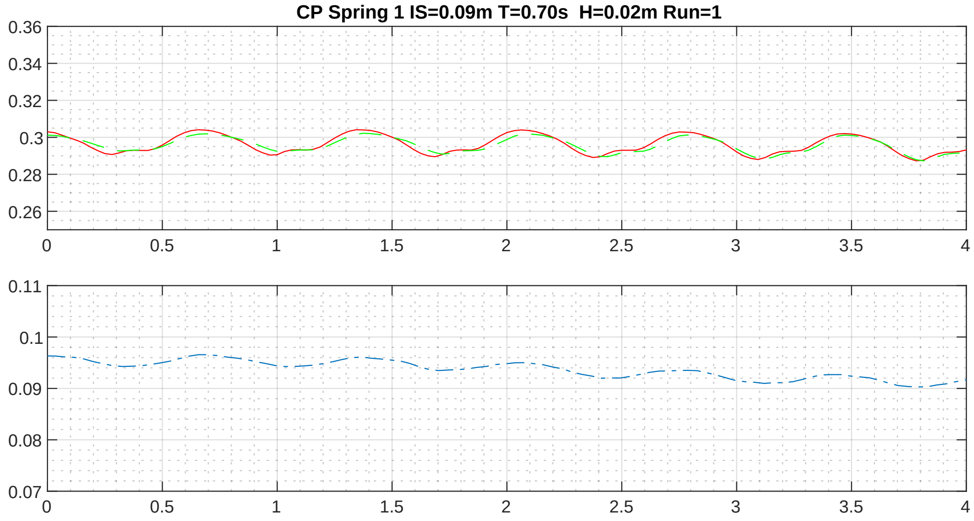

CP Spring 1 IS=0.15m T=1.75s H=0.03m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

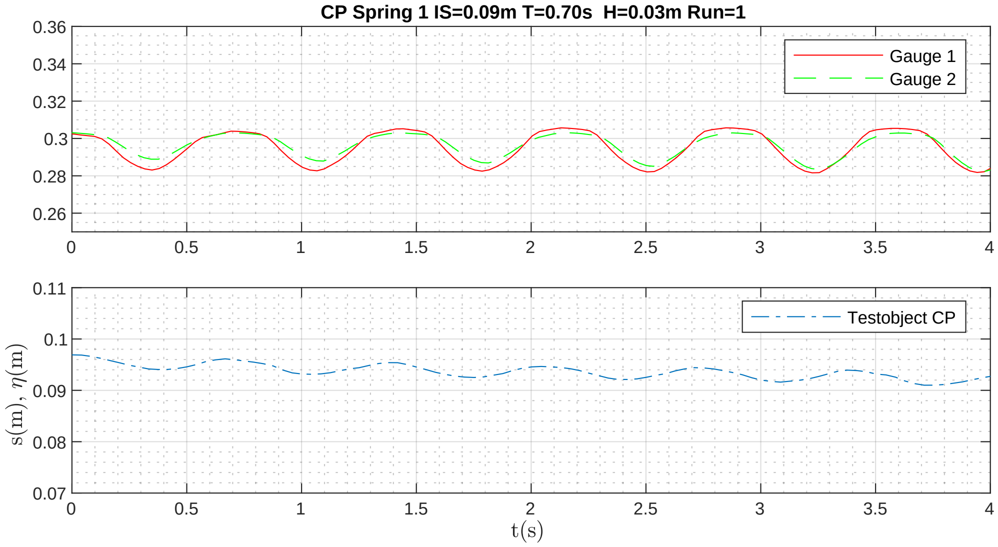

CP Spring 1 IS=0.15m T=1.75s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

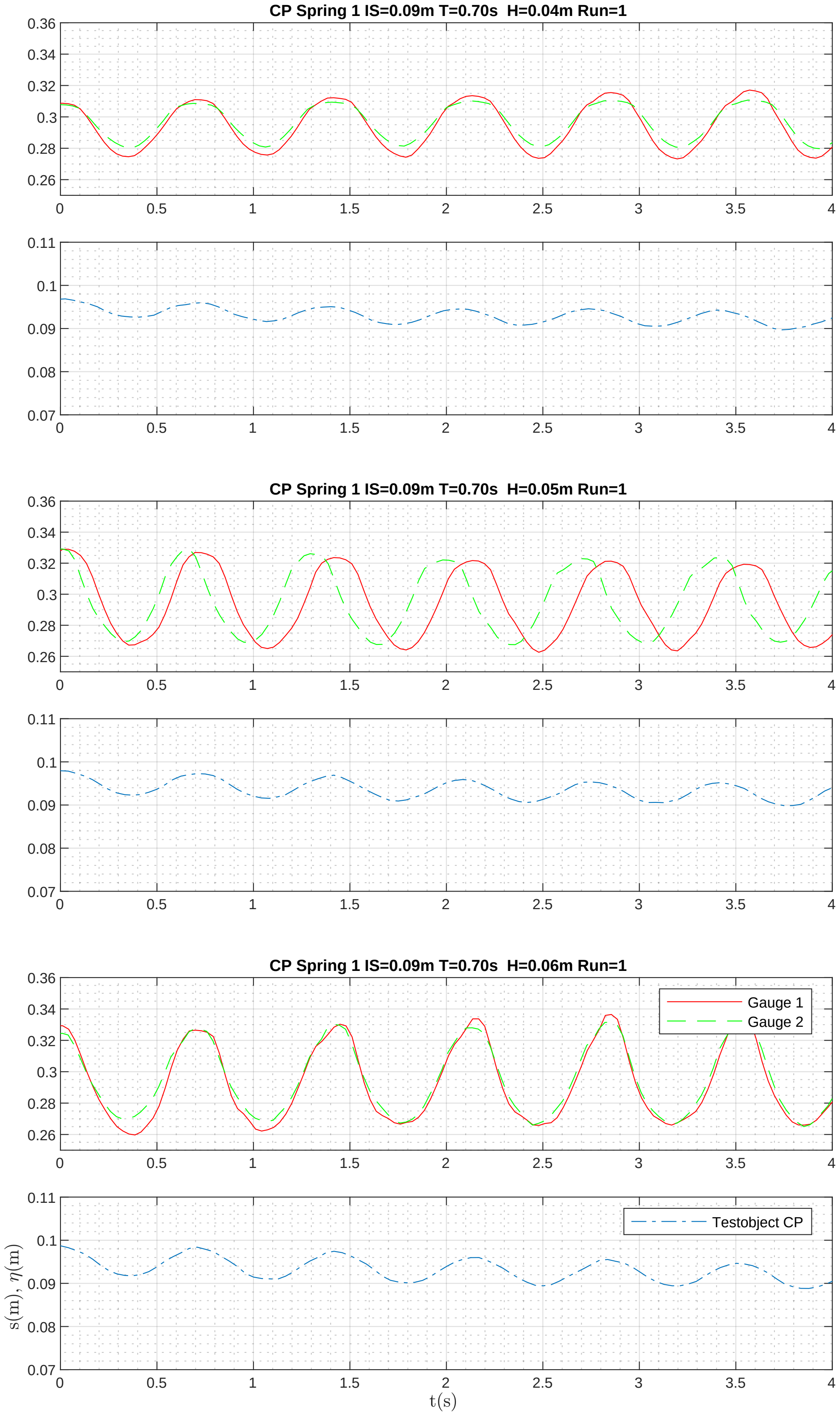

CP Spring 1 IS=0.15m T=1.75s H=0.05m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

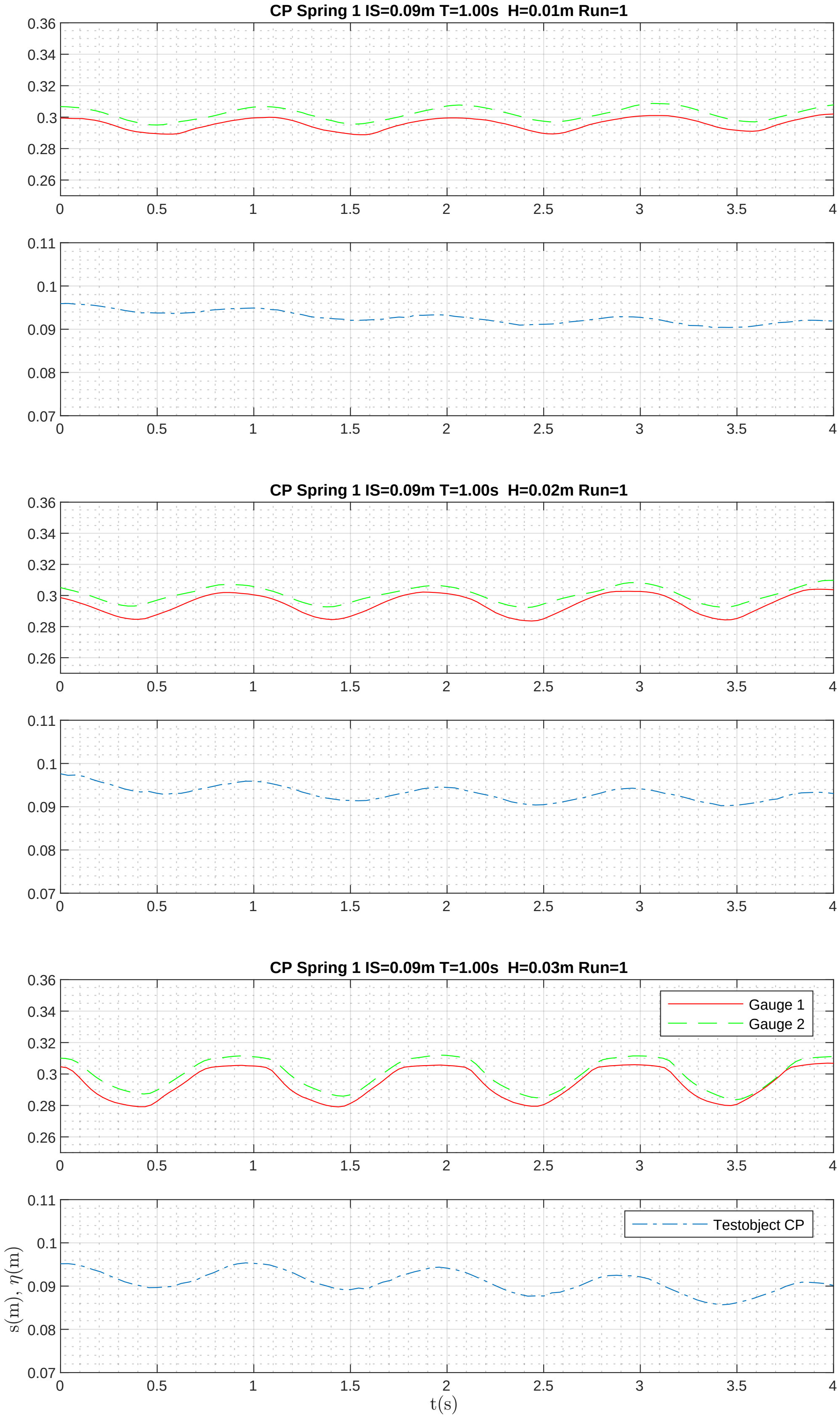

CP Spring 1 IS=0.15m T=1.75s H=0.06m Run=1
Gauge 1
Gauge 2
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
Testobject CP
0.18
0.16
0.14
0.12
s(m), η(m)
t(s)

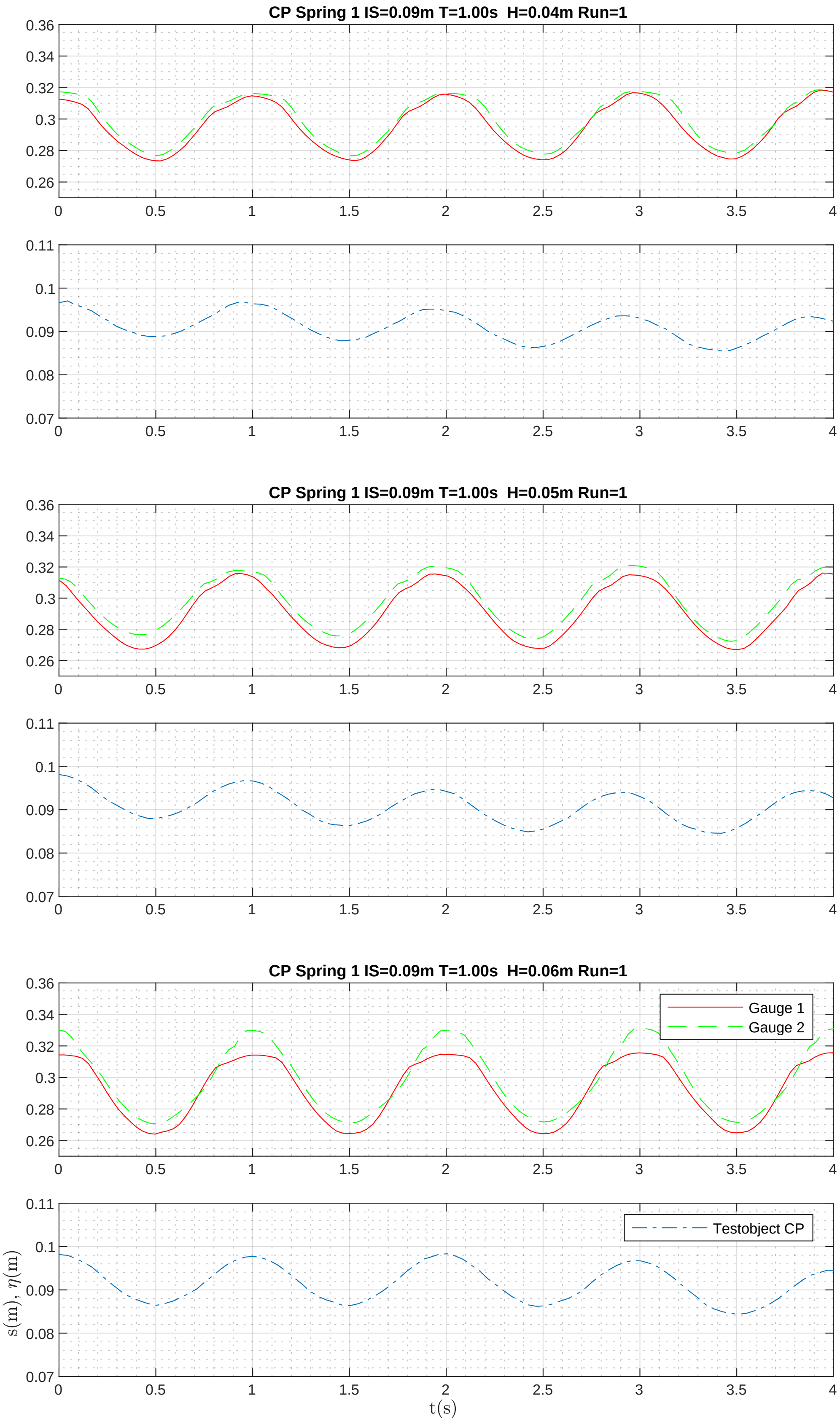

CP Spring 1 IS=0.09m T=0.70s H=0.01m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.11
0.1
0.09
0.08
0.07

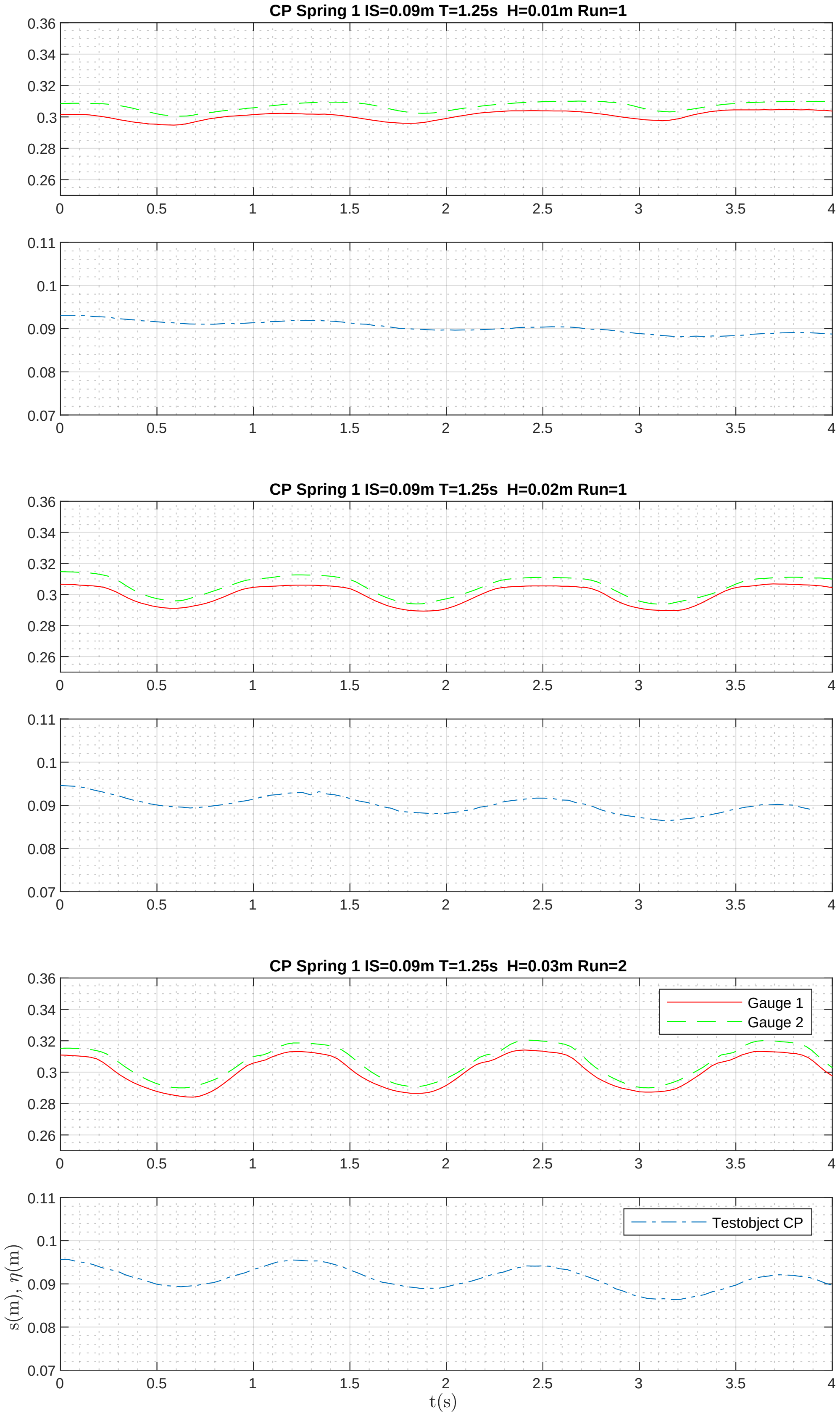

CP Spring 1 IS=0.09m T=0.70s H=0.02m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.11
0.1
0.09
0.08
0.07

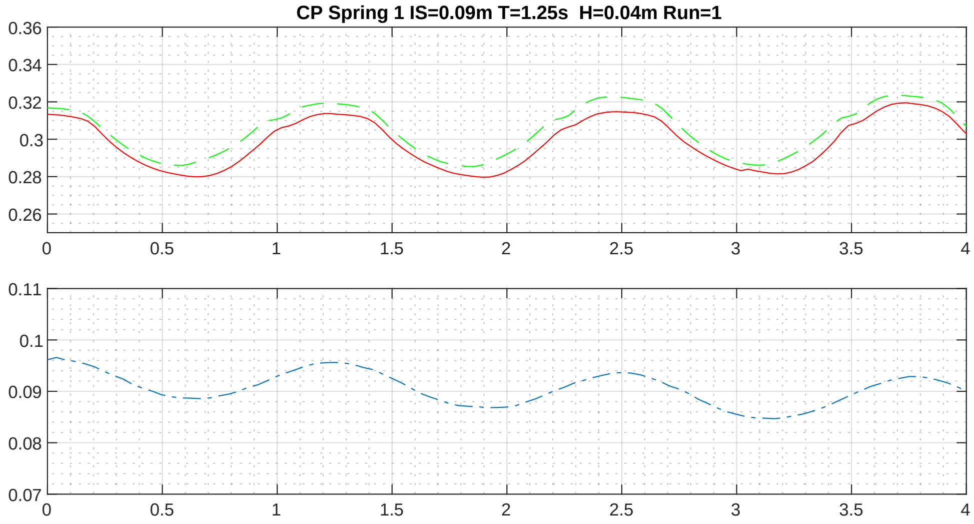

CP Spring 1 IS=0.09m T=0.70s H=0.03m Run=1
Gauge 1
Gauge 2
Testobject CP
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.11
0.1
0.09
0.08
0.07
s(m), η(m)
t(s)

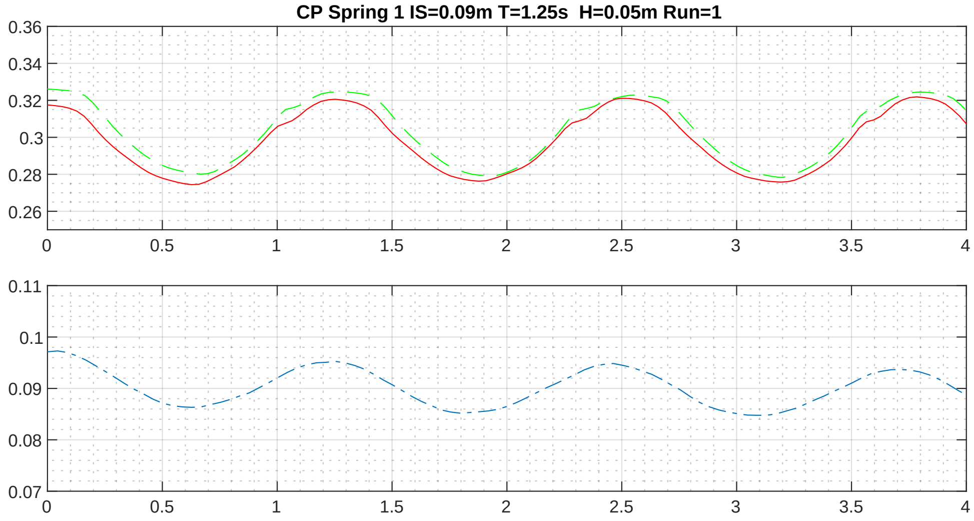

CP Spring 1 IS=0.09m T=0.70s H=0.04m Run=1
CP Spring 1 IS=0.09m T=0.70s H=0.05m Run=1
CP Spring 1 IS=0.09m T=0.70s H=0.06m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.11
0.1
0.09
0.08
0.07
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

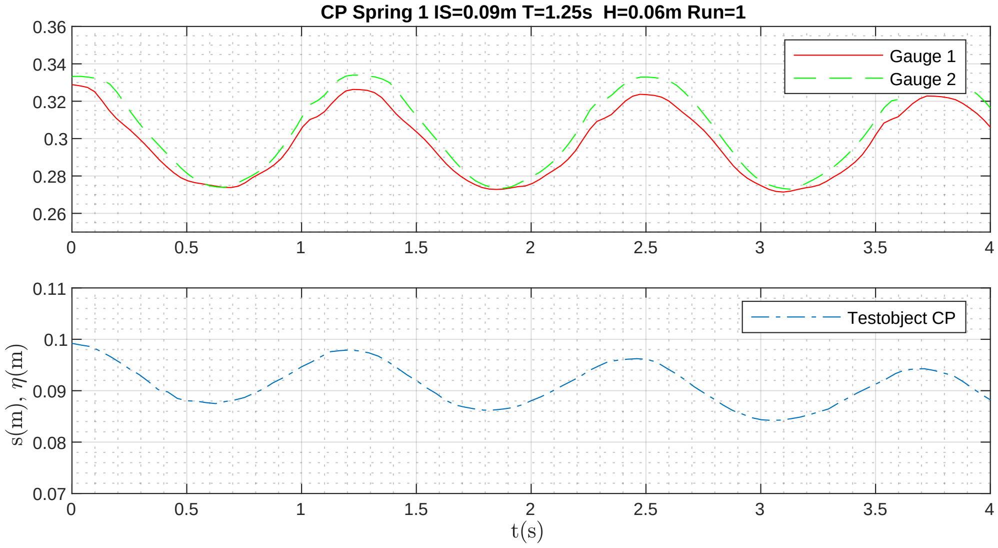

CP Spring 1 IS=0.09m T=1.00s H=0.01m Run=1
CP Spring 1 IS=0.09m T=1.00s H=0.02m Run=1
CP Spring 1 IS=0.09m T=1.00s H=0.03m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.11
0.1
0.09
0.08
0.07
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

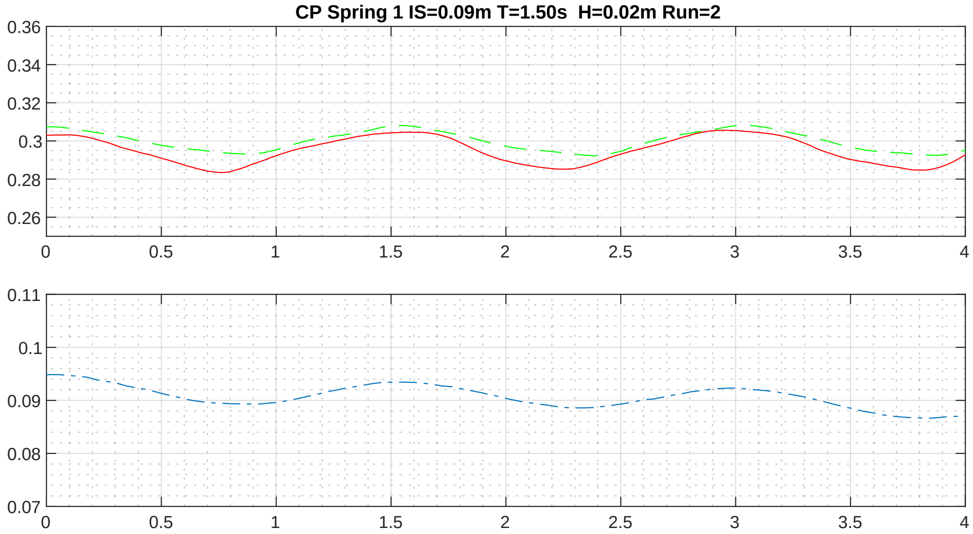

CP Spring 1 IS=0.09m T=1.00s H=0.04m Run=1
CP Spring 1 IS=0.09m T=1.00s H=0.05m Run=1
CP Spring 1 IS=0.09m T=1.00s H=0.06m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.11
0.1
0.09
0.08
0.07
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

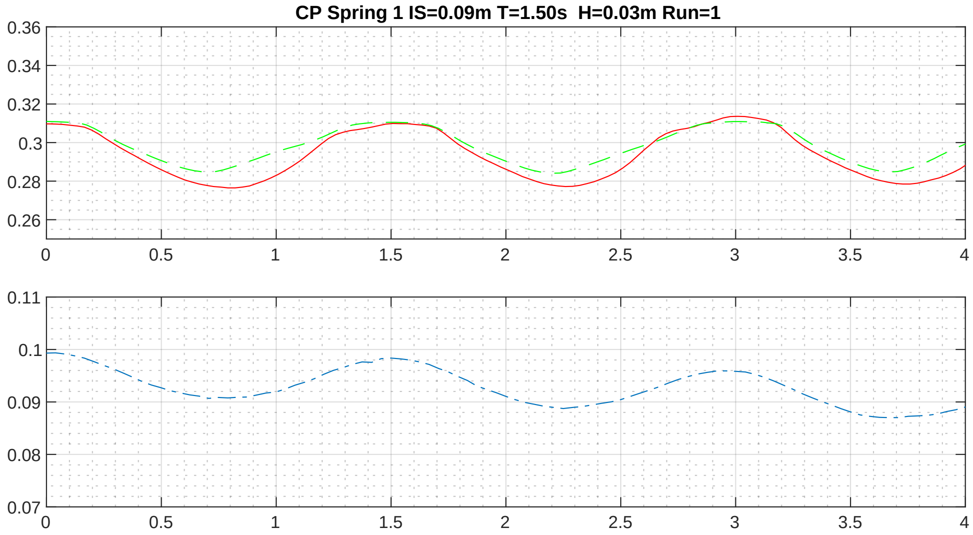

CP Spring 1 IS=0.09m T=1.25s H=0.01m Run=1
CP Spring 1 IS=0.09m T=1.25s H=0.02m Run=1
CP Spring 1 IS=0.09m T=1.25s H=0.03m Run=2
Gauge 1
Gauge 2
Testobject CP
0.36
0.34
0.32
0.3
0.28
0.26
0.11
0.1
0.09
0.08
0.07
0
0.5
1
1.5
2
2.5
3
3.5
4
s(m), η(m)
t(s)

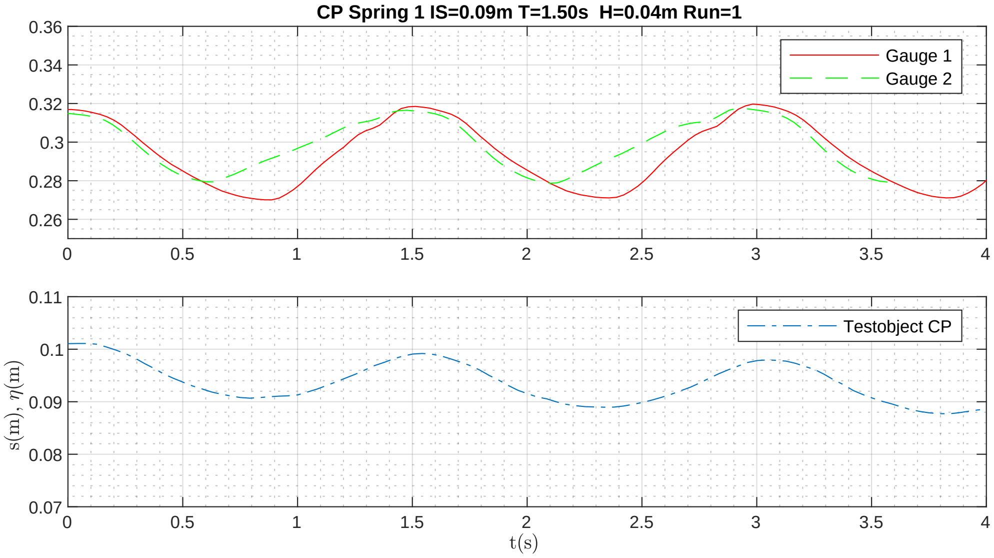

CP Spring 1 IS=0.09m T=1.25s H=0.04m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.11
0.1
0.09
0.08
0.07
0
0.5
1
1.5
2
2.5
3
3.5
4

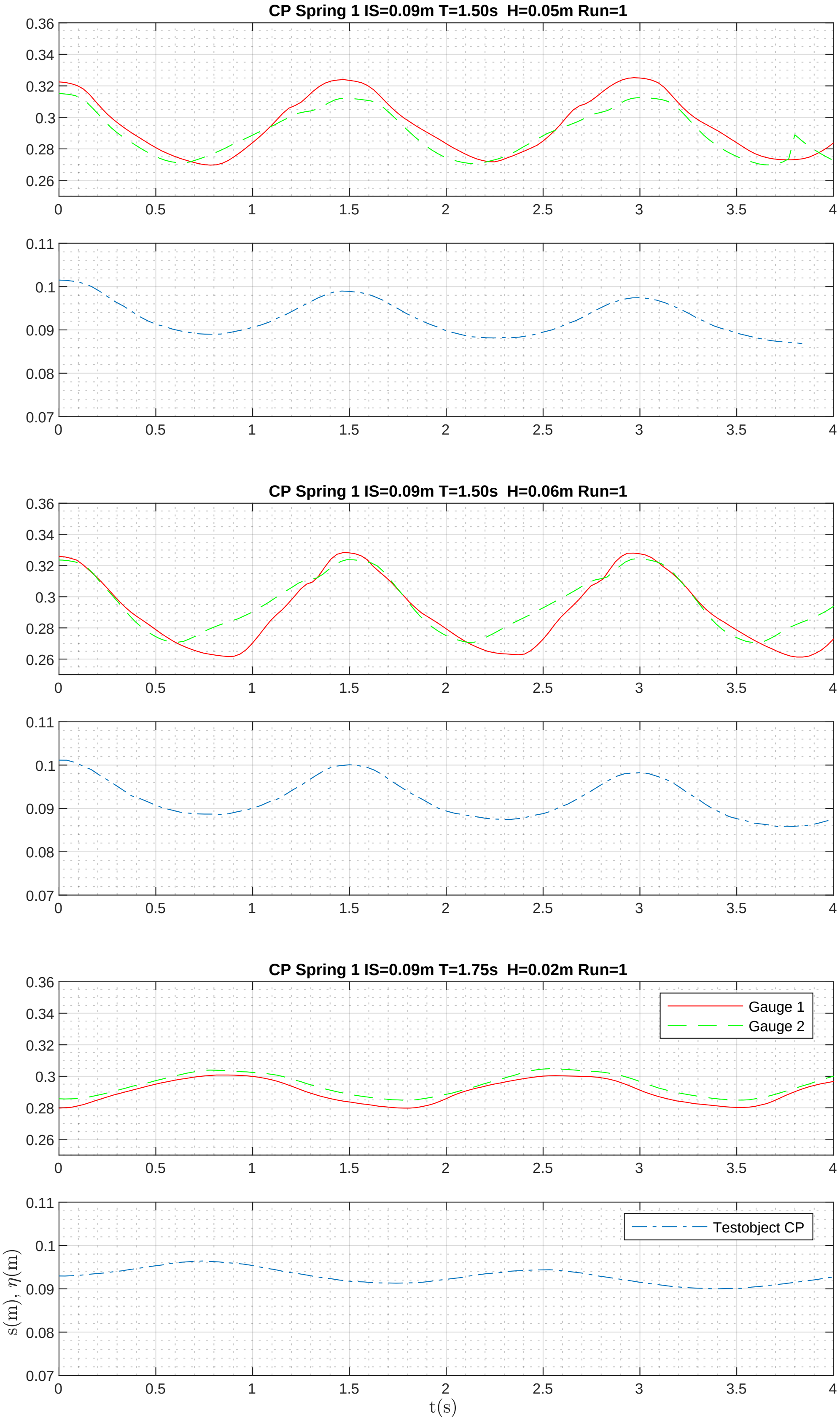

CP Spring 1 IS=0.09m T=1.25s H=0.05m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.11
0.1
0.09
0.08
0.07
0
0.5
1
1.5
2
2.5
3
3.5
4

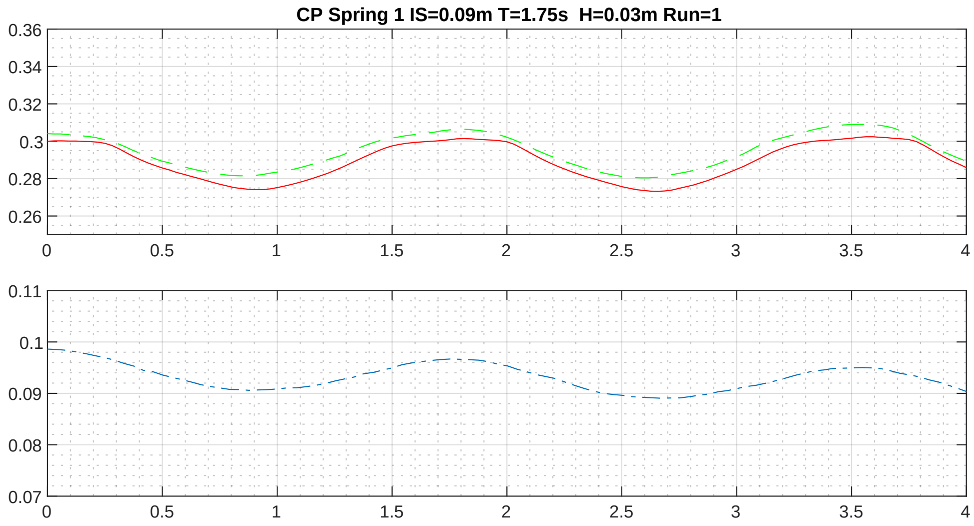

CP Spring 1 IS=0.09m T=1.25s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.36
0.34
0.32
0.3
0.28
0.26
0.11
0.1
0.09
0.08
0.07
0
0.5
1
1.5
2
2.5
3
3.5
4


CP Spring 1 IS=0.09m T=1.50s H=0.02m Run=2
0.36
0.34
0.32
0.3
0.28
0.26
0.11
0.1
0.09
0.08
0.07
0
0.5
1
1.5
2
2.5
3
3.5
4

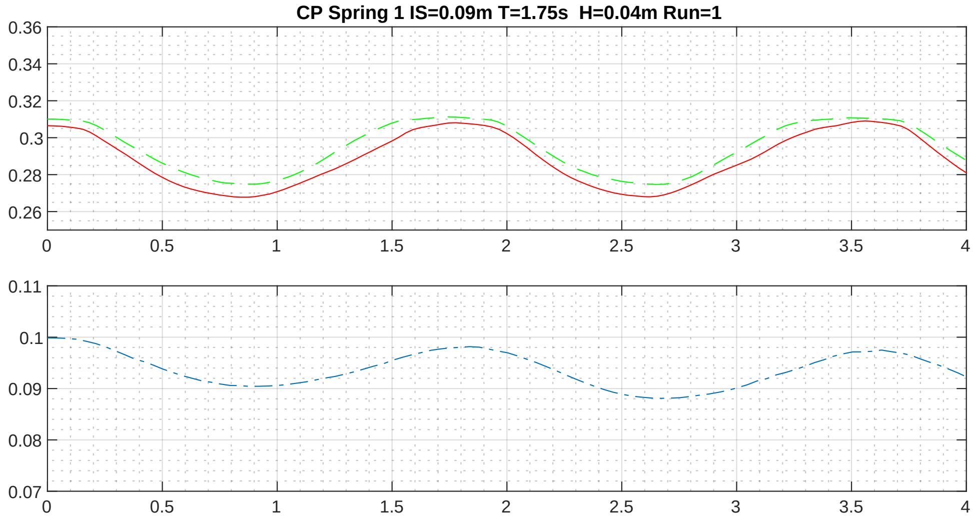

CP Spring 1 IS=0.09m T=1.50s H=0.03m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.11
0.1
0.09
0.08
0.07
0
0.5
1
1.5
2
2.5
3
3.5
4


CP Spring 1 IS=0.09m T=1.50s H=0.04m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.36
0.34
0.32
0.3
0.28
0.26
0.11
0.1
0.09
0.08
0.07
0
0.5
1
1.5
2
2.5
3
3.5
4

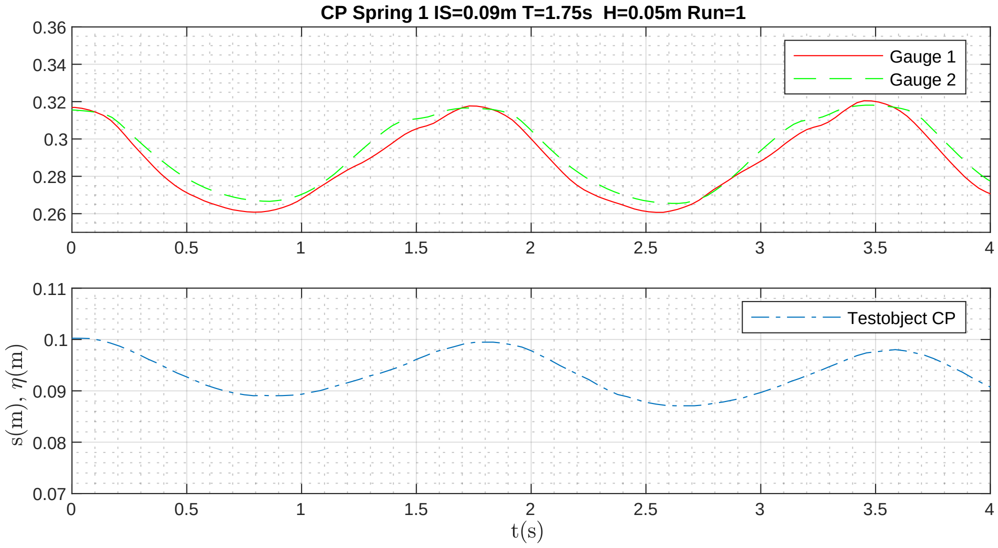

CP Spring 1 IS=0.09m T=1.50s H=0.05m Run=1
CP Spring 1 IS=0.09m T=1.50s H=0.06m Run=1
CP Spring 1 IS=0.09m T=1.75s H=0.02m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.11
0.1
0.09
0.08
0.07
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)



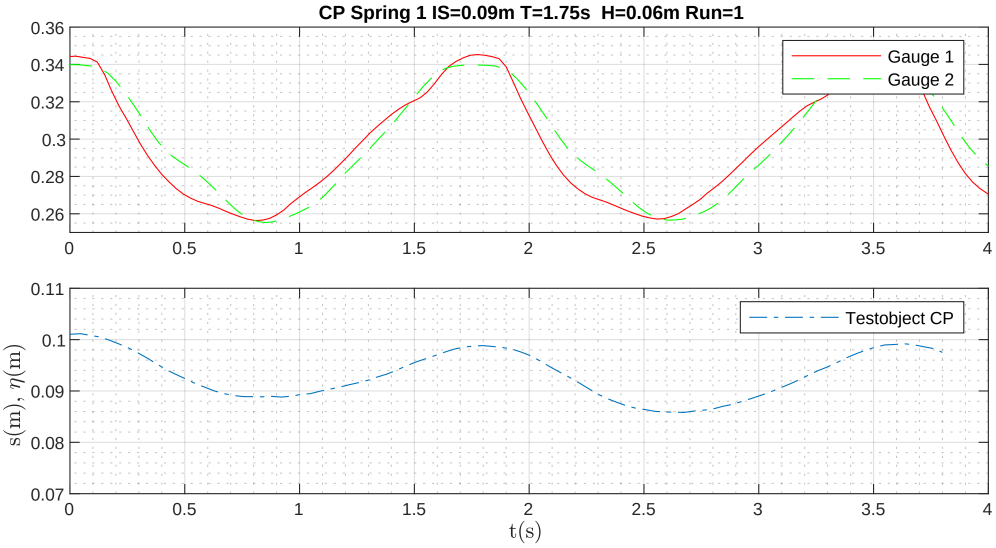
CP Spring 1 IS=0.09m T=1.75s H=0.03m Run=1

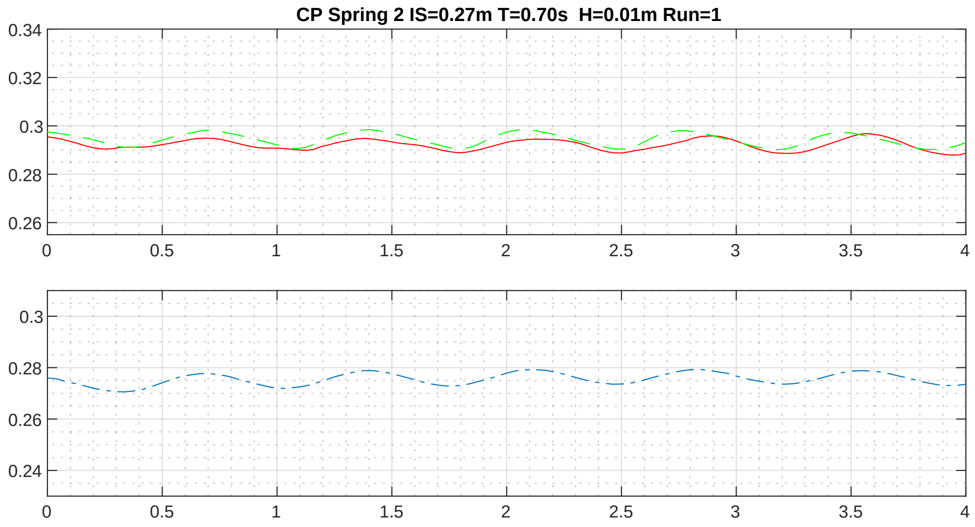


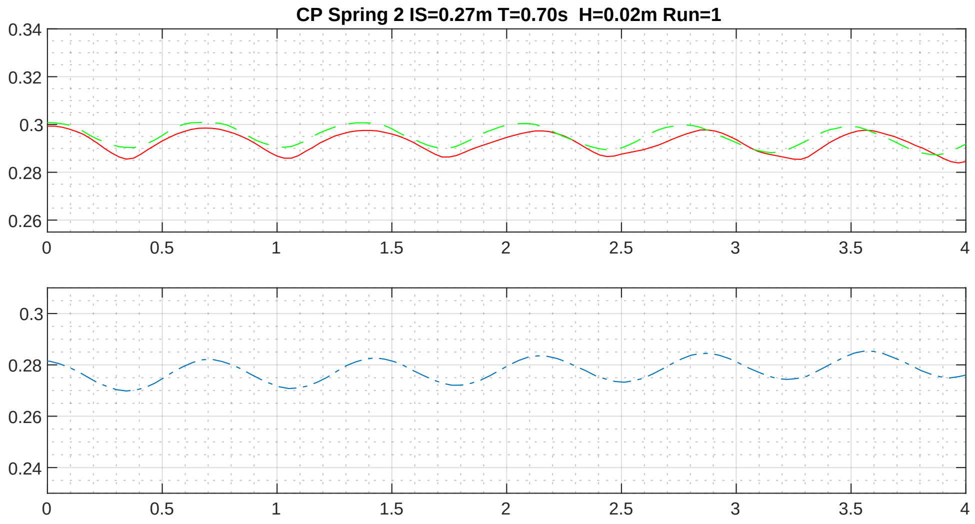
CP Spring 1 IS=0.09m T=1.75s H=0.04m Run=1

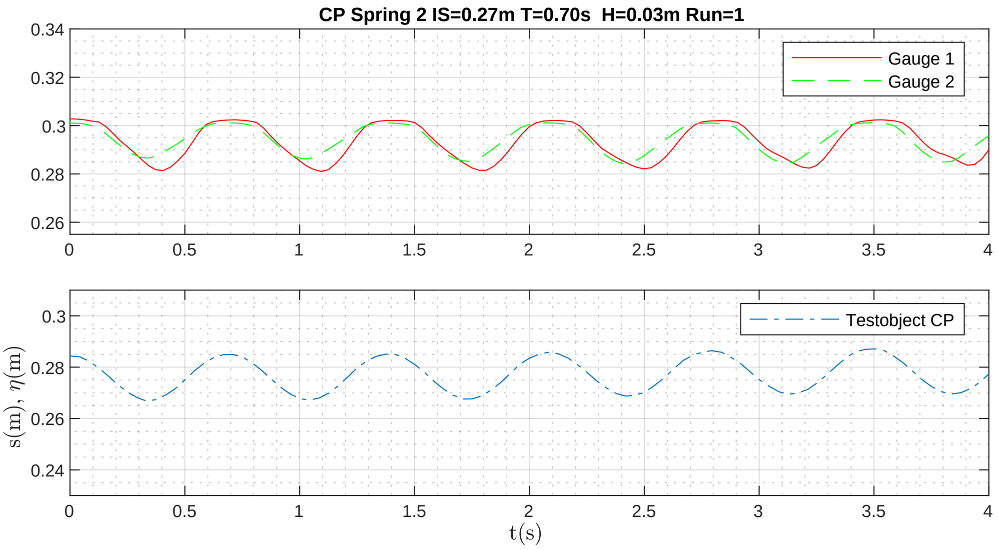


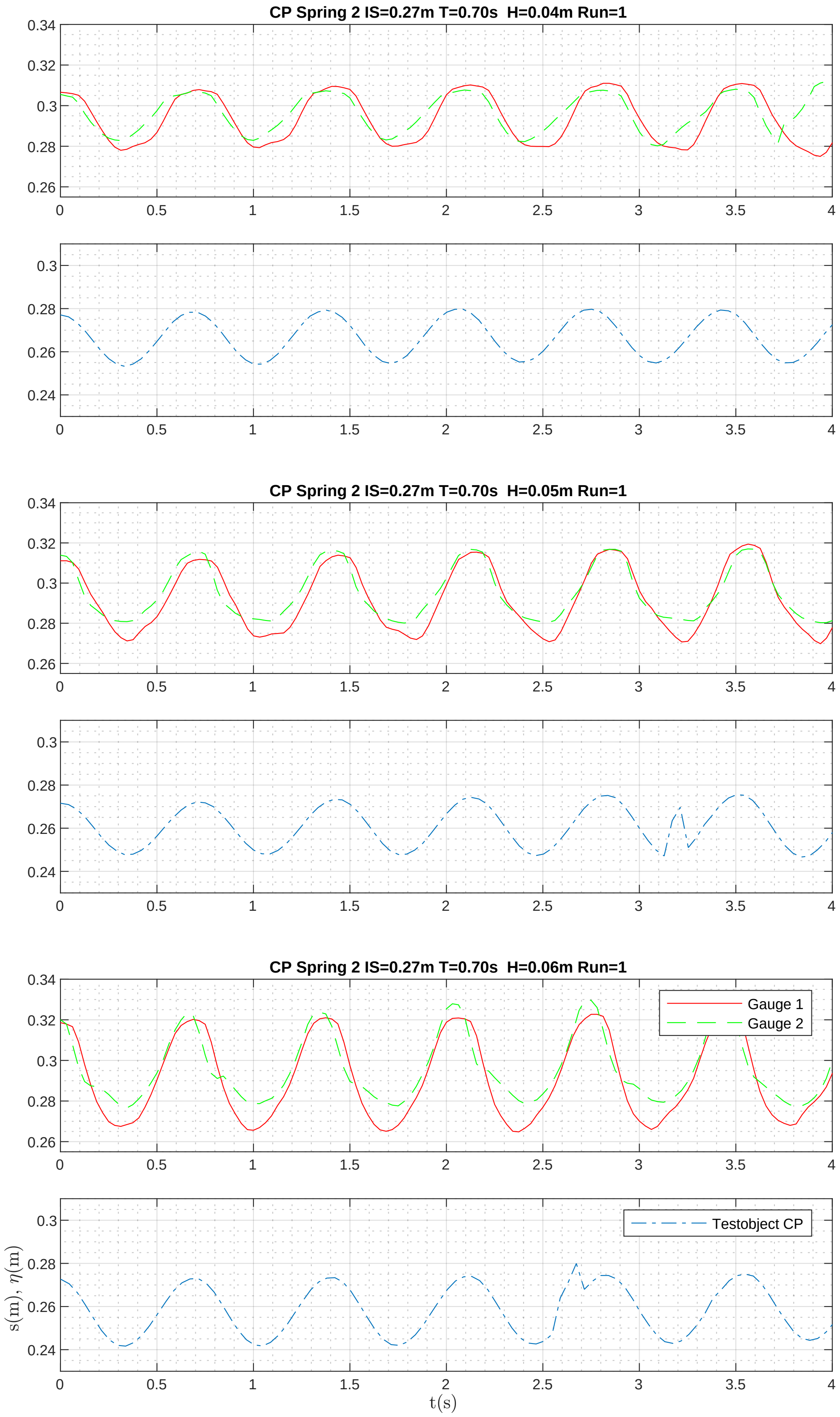
CP Spring 1 IS=0.09m T=1.75s H=0.05m Run=1


CP Spring 1 IS=0.09m T=1.75s H=0.06m Run=1
Gauge 1
Gauge 2
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
Testobject CP
0.11
0.1
0.09
0.08
0.07
s(m), η(m)
t(s)


CP Spring 2 IS=0.27m T=0.70s H=0.01m Run=1
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4


CP Spring 2 IS=0.27m T=0.70s H=0.02m Run=1
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4

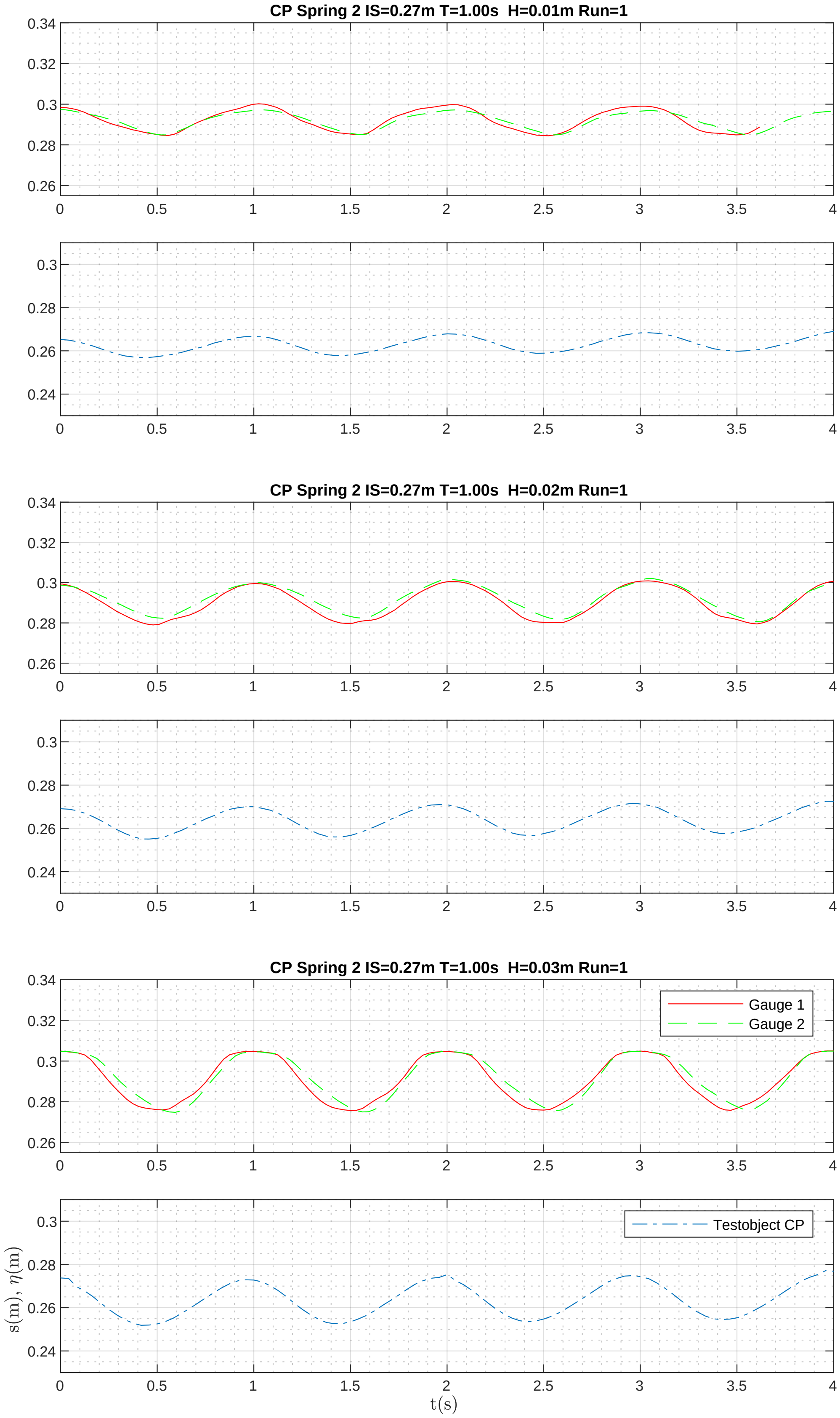

CP Spring 2 IS=0.27m T=0.70s H=0.03m Run=1
Gauge 1
Gauge 2
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
Testobject CP
s(m), η(m)
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4
t(s)

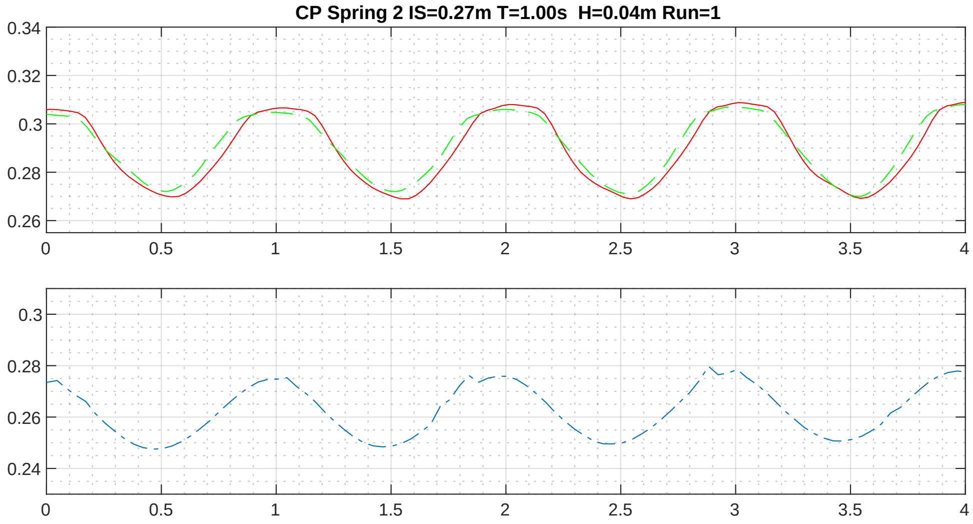


CP Spring 2 IS=0.27m T=0.70s H=0.04m Run=1

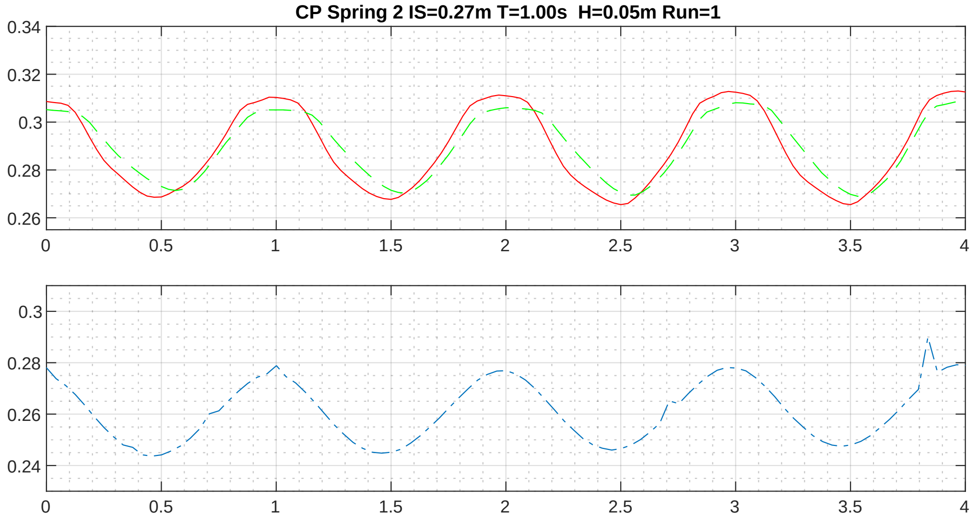
CP Spring 2 IS=0.27m T=0.70s H=0.05m Run=1

CP Spring 2 IS=0.27m T=0.70s H=0.06m Run=1

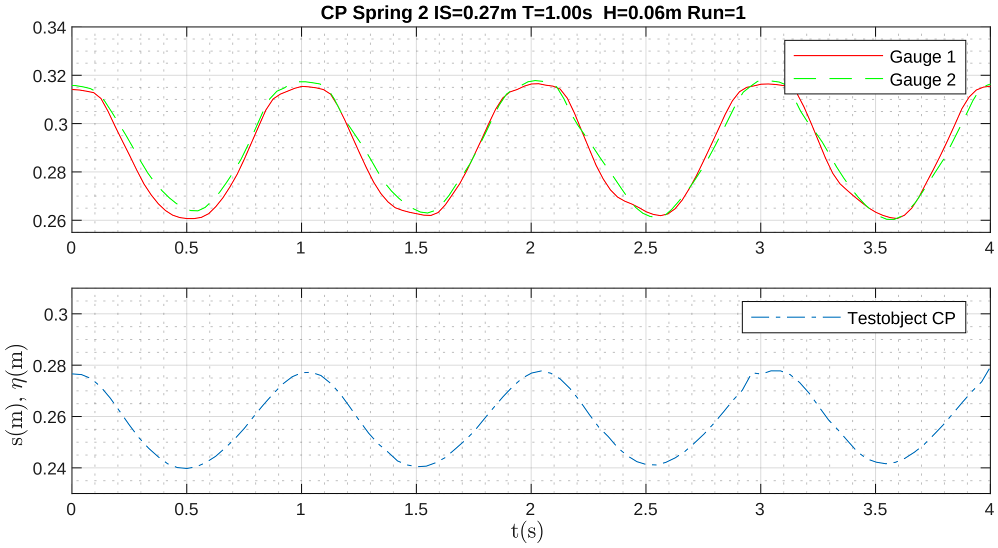

CP Spring 2 IS=0.27m T=1.00s H=0.01m Run=1
CP Spring 2 IS=0.27m T=1.00s H=0.02m Run=1
CP Spring 2 IS=0.27m T=1.00s H=0.03m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)



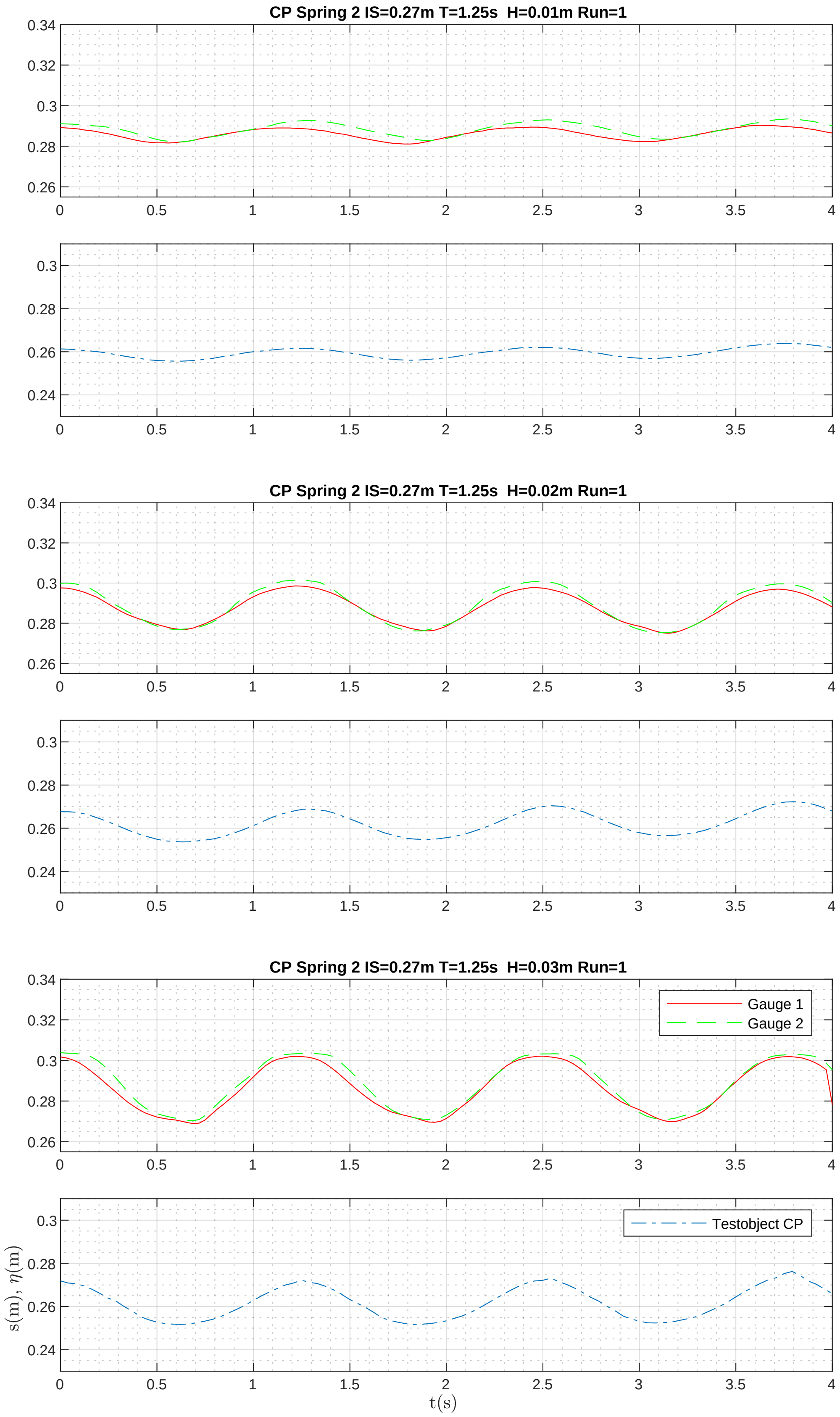
CP Spring 2 IS=0.27m T=1.00s H=0.04m Run=1

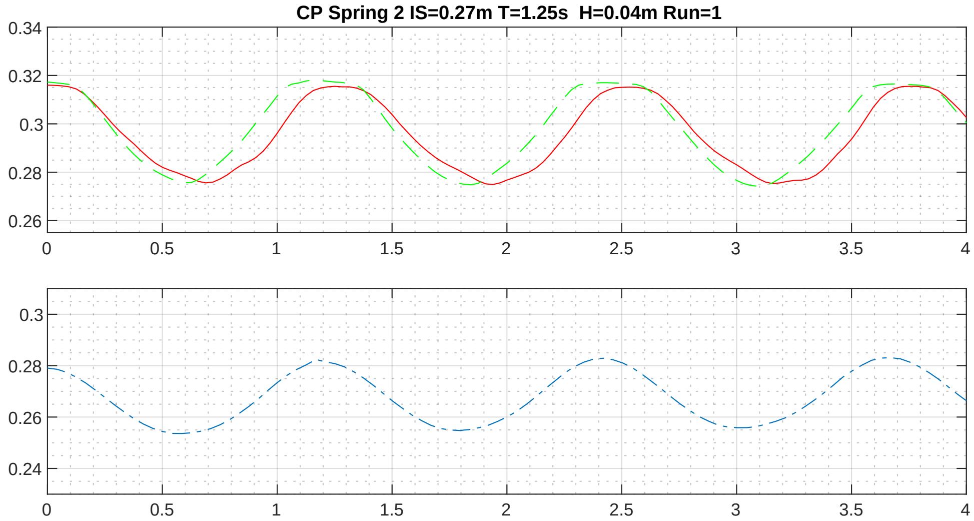


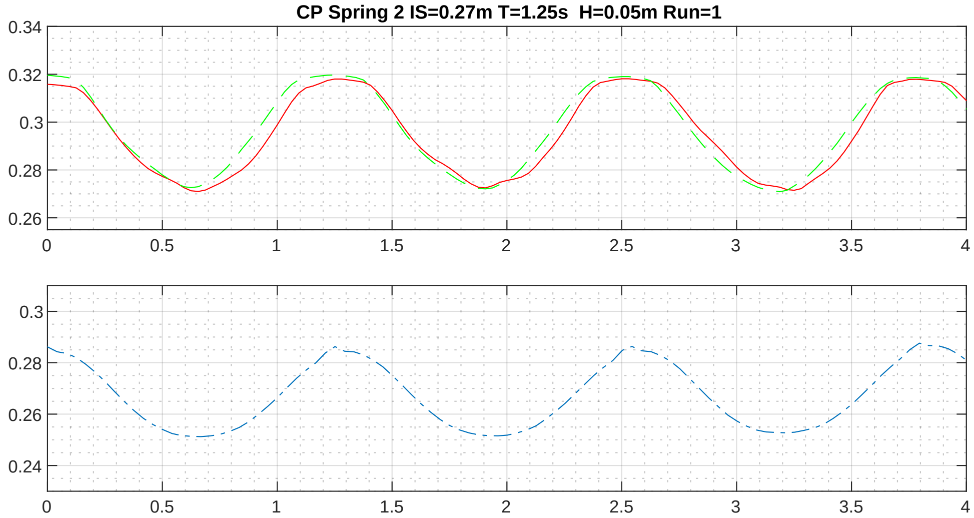
CP Spring 2 IS=0.27m T=1.00s H=0.05m Run=1

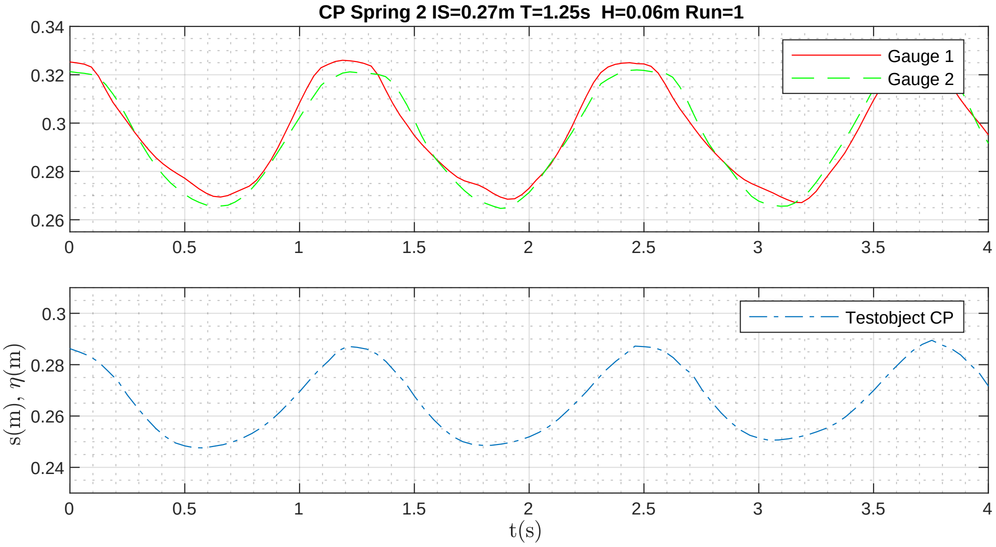


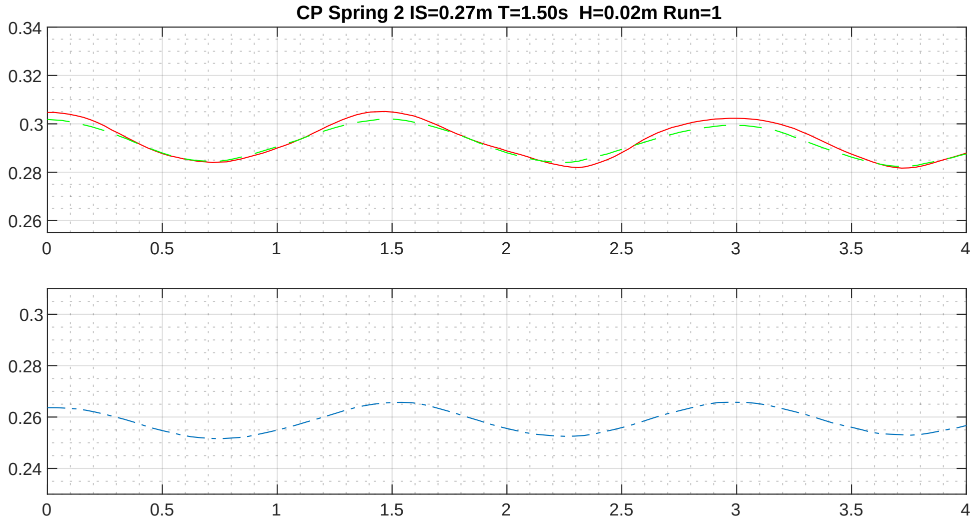
CP Spring 2 IS=0.27m T=1.00s H=0.06m Run=1

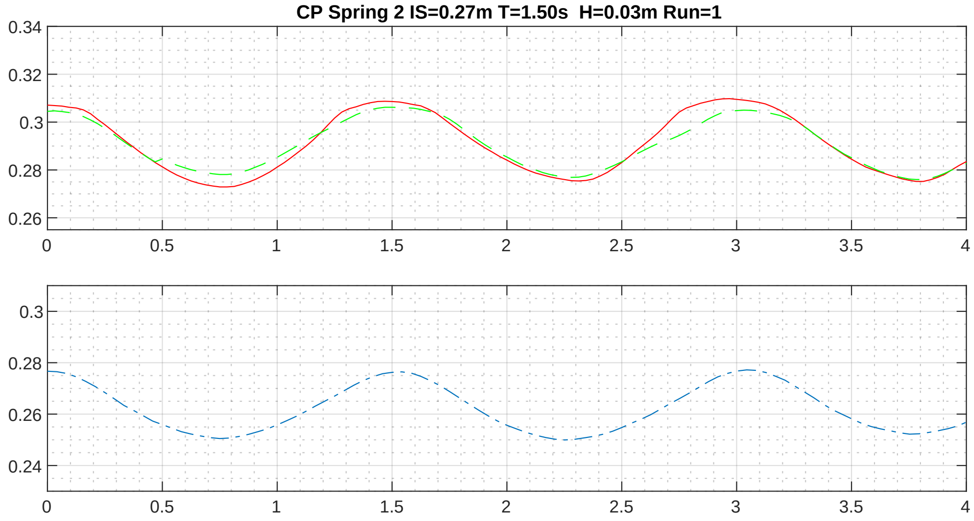

CP Spring 2 IS=0.27m T=1.25s H=0.01m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4
CP Spring 2 IS=0.27m T=1.25s H=0.02m Run=1
CP Spring 2 IS=0.27m T=1.25s H=0.03m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

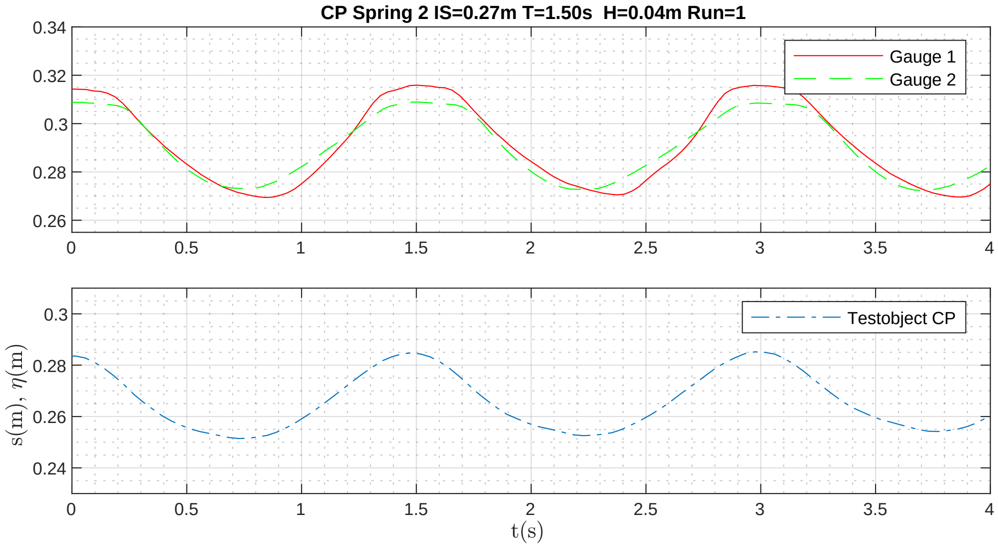

CP Spring 2 IS=0.27m T=1.25s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.3
0.28
0.26
0.24

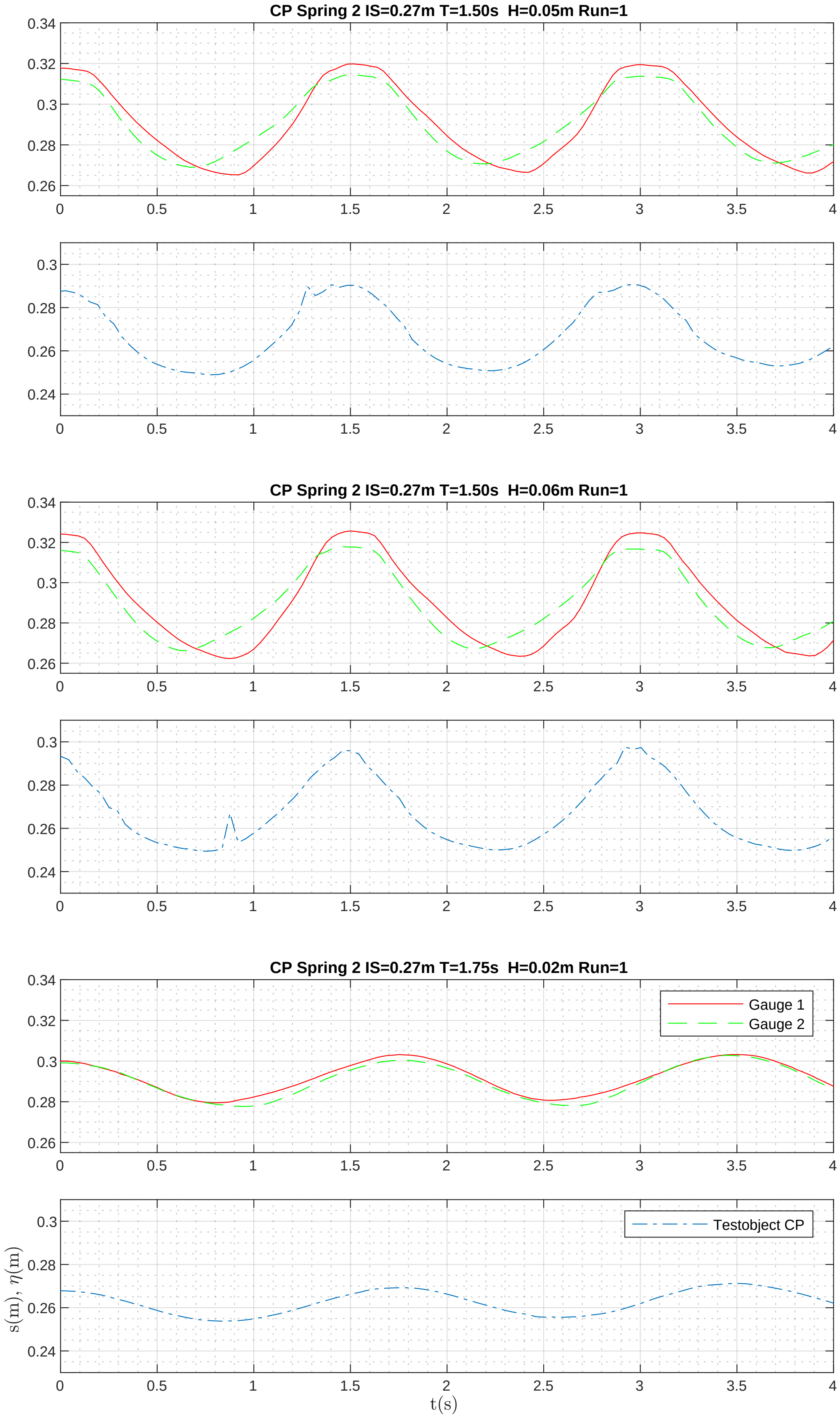

CP Spring 2 IS=0.27m T=1.25s H=0.05m Run=1
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.3
0.28
0.26
0.24

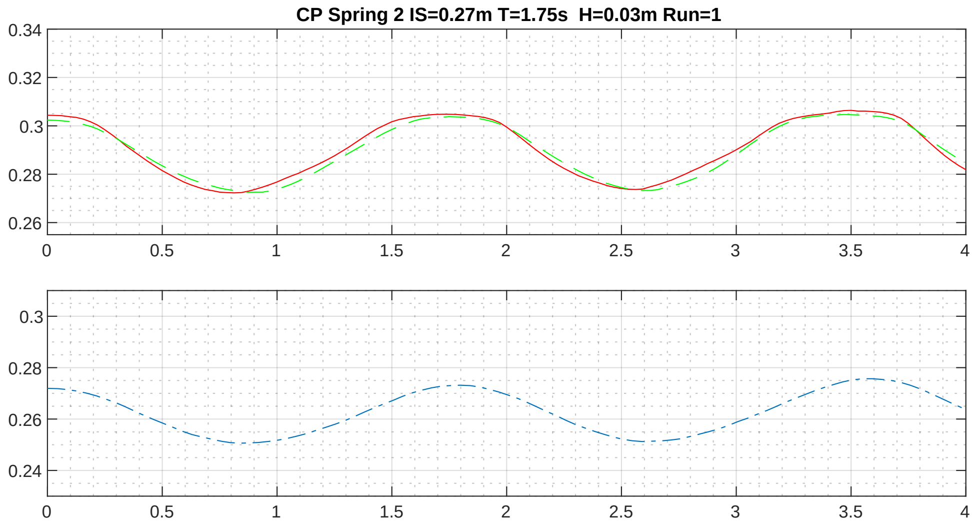

CP Spring 2 IS=0.27m T=1.25s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject CP
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4
s(m), η(m)
t(s)


CP Spring 2 IS=0.27m T=1.50s H=0.02m Run=1

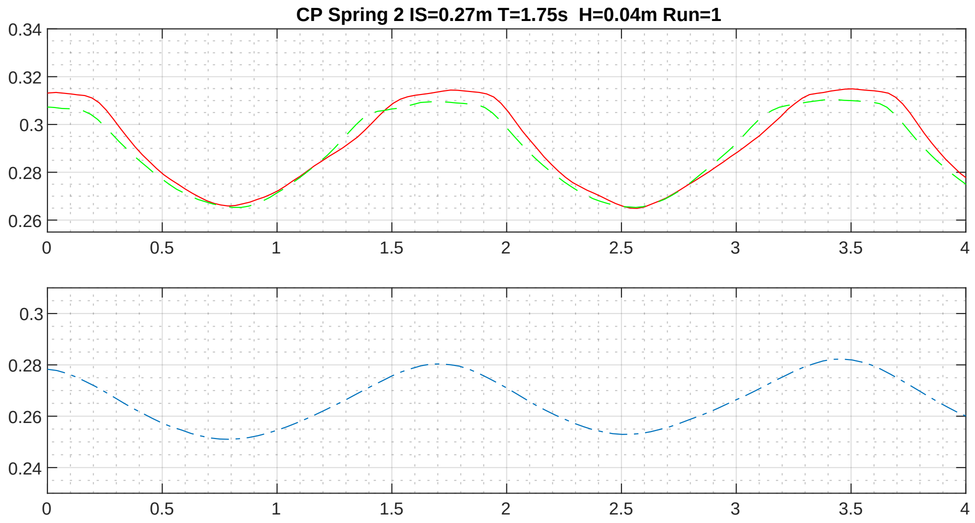

CP Spring 2 IS=0.27m T=1.50s H=0.03m Run=1


CP Spring 2 IS=0.27m T=1.50s H=0.04m Run=1

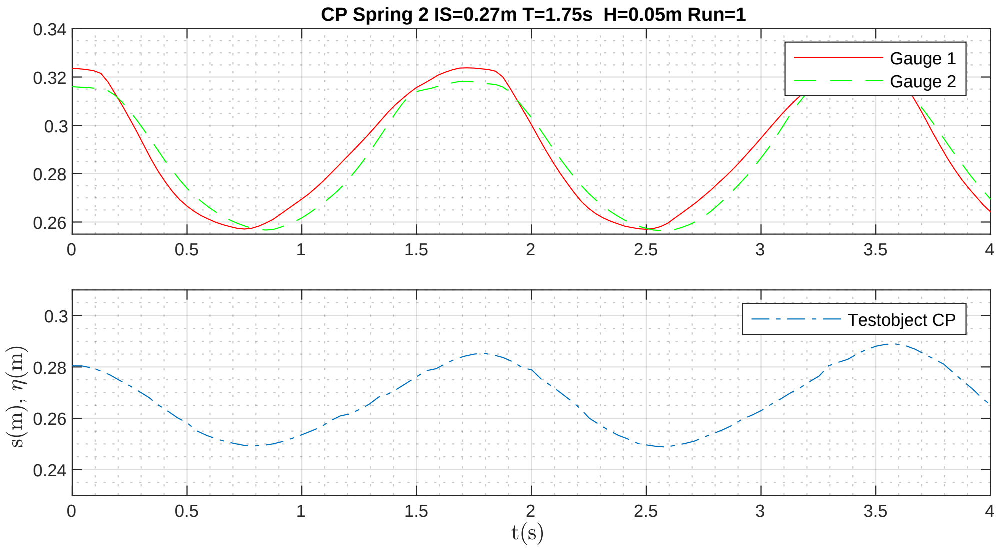

CP Spring 2 IS=0.27m T=1.50s H=0.05m Run=1
CP Spring 2 IS=0.27m T=1.50s H=0.06m Run=1
CP Spring 2 IS=0.27m T=1.75s H=0.02m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)



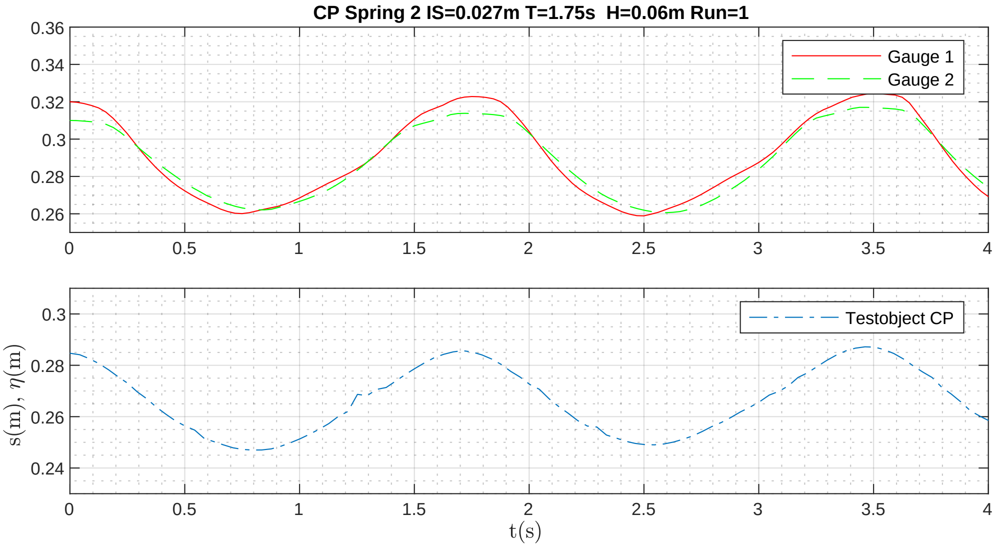
CP Spring 2 IS=0.27m T=1.75s H=0.03m Run=1

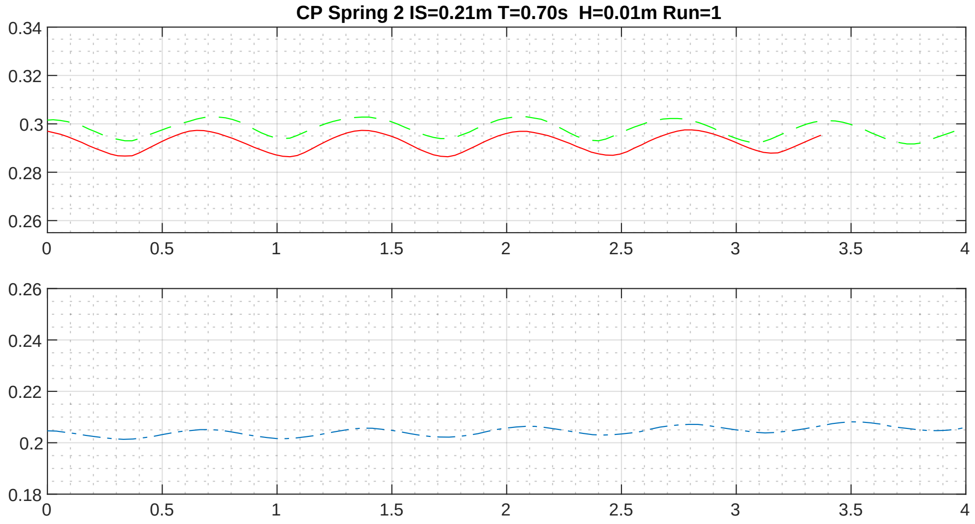


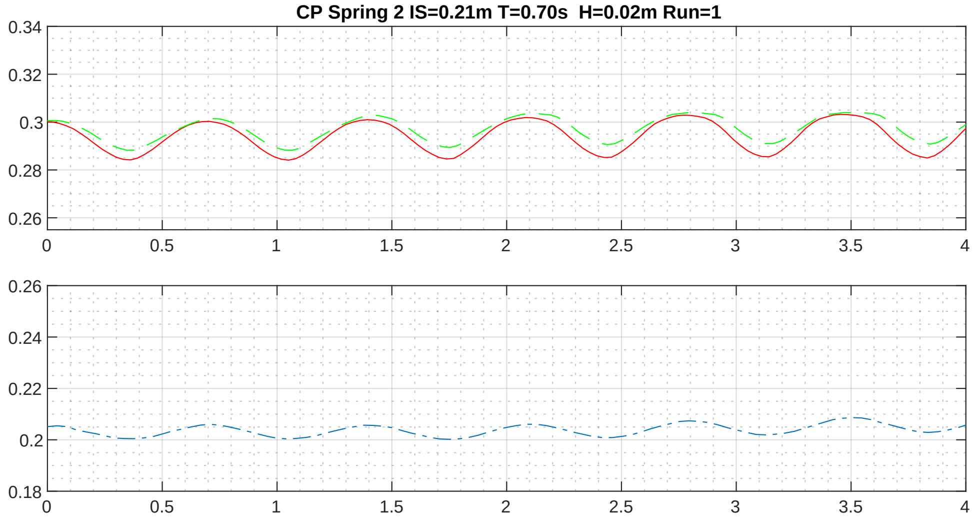
CP Spring 2 IS=0.27m T=1.75s H=0.04m Run=1

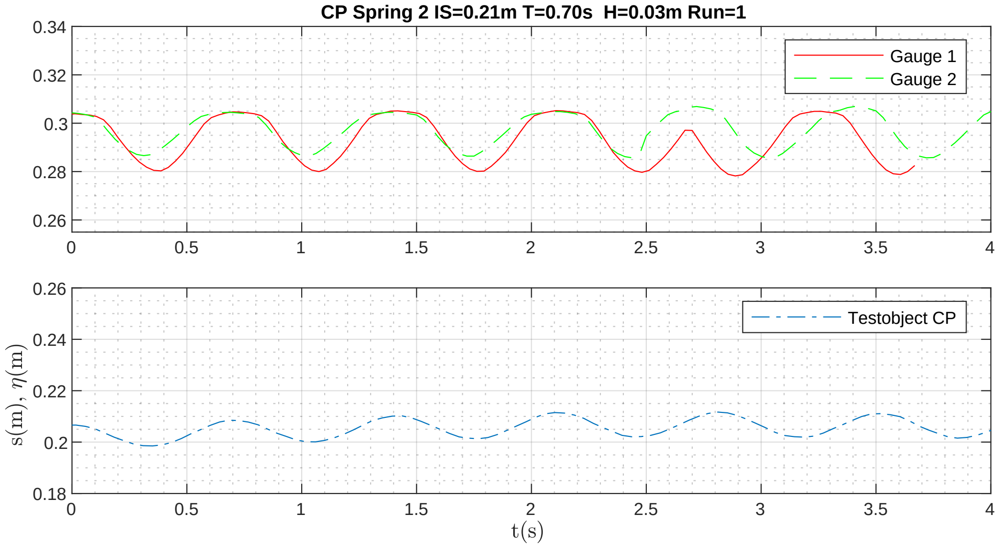


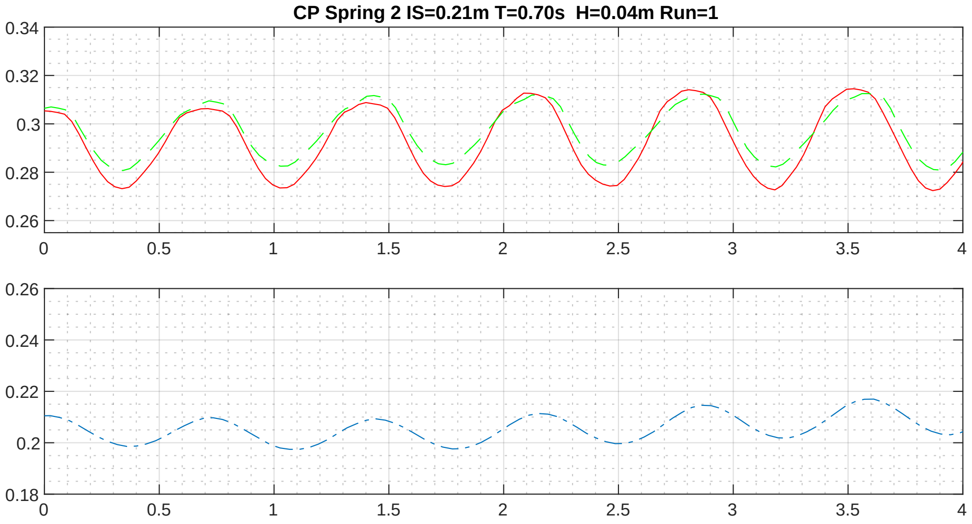
CP Spring 2 IS=0.27m T=1.75s H=0.05m Run=1

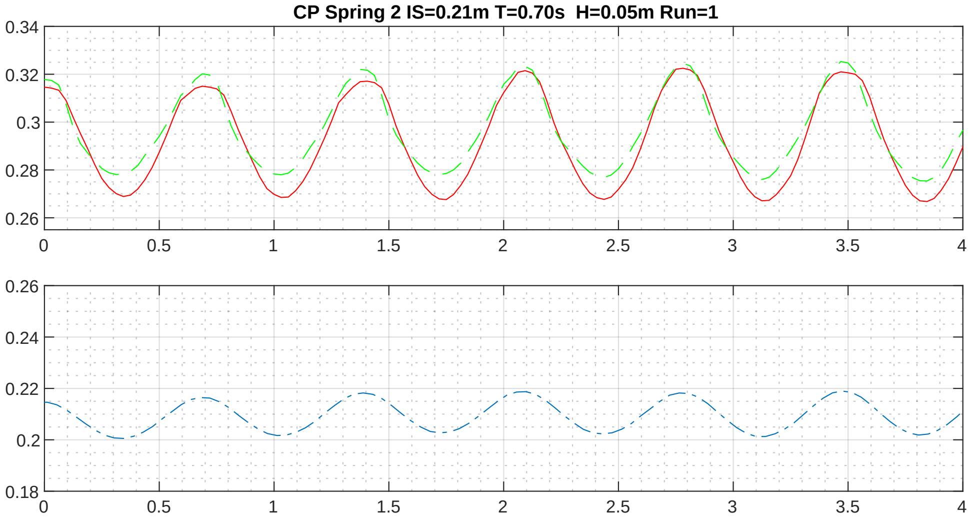

CP Spring 2 IS=0.027m T=1.75s H=0.06m Run=1
Gauge 1
Gauge 2
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
Testobject CP
0.3
0.28
0.26
0.24
s(m), η(m)
t(s)

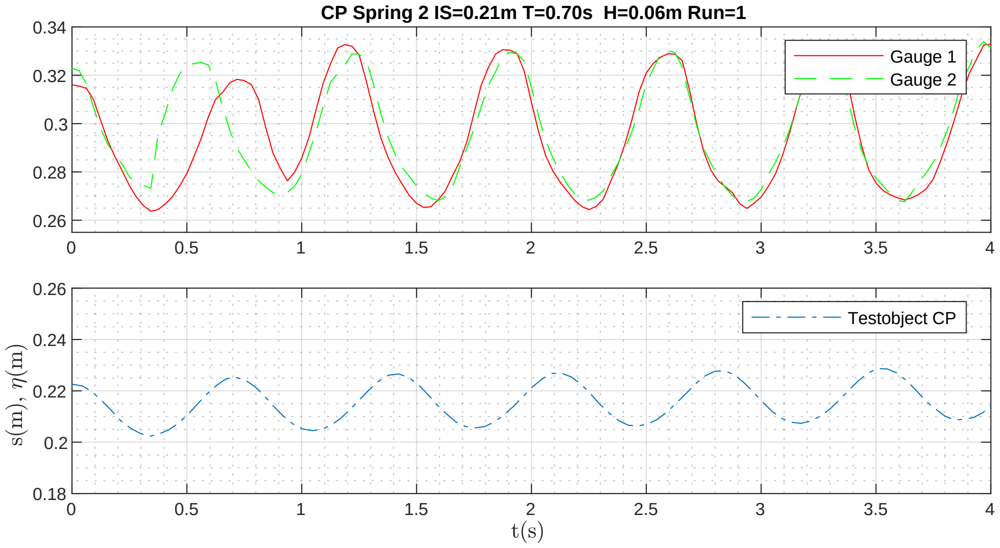

CP Spring 2 IS=0.21m T=0.70s H=0.01m Run=1
0.34
0.32
0.3
0.28
0.26
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

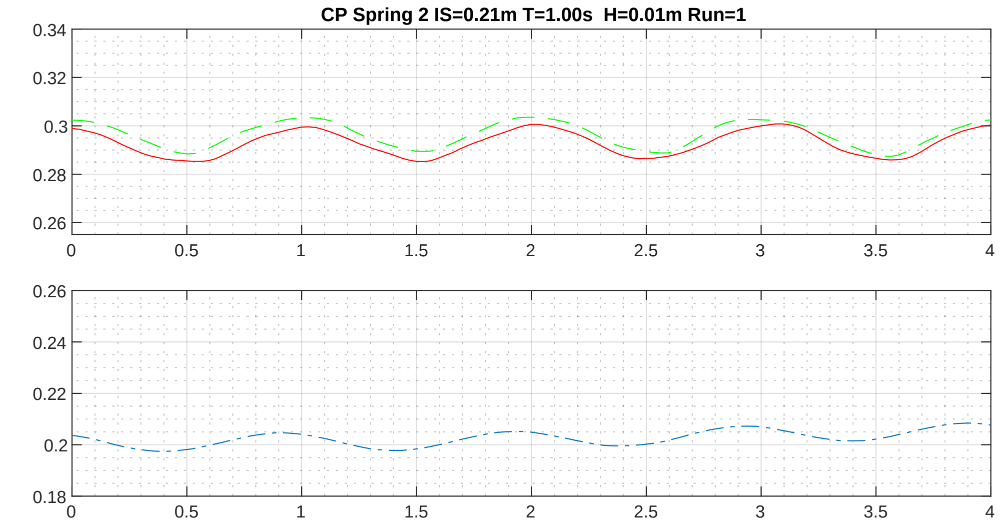

CP Spring 2 IS=0.21m T=0.70s H=0.02m Run=1
0.34
0.32
0.3
0.28
0.26
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

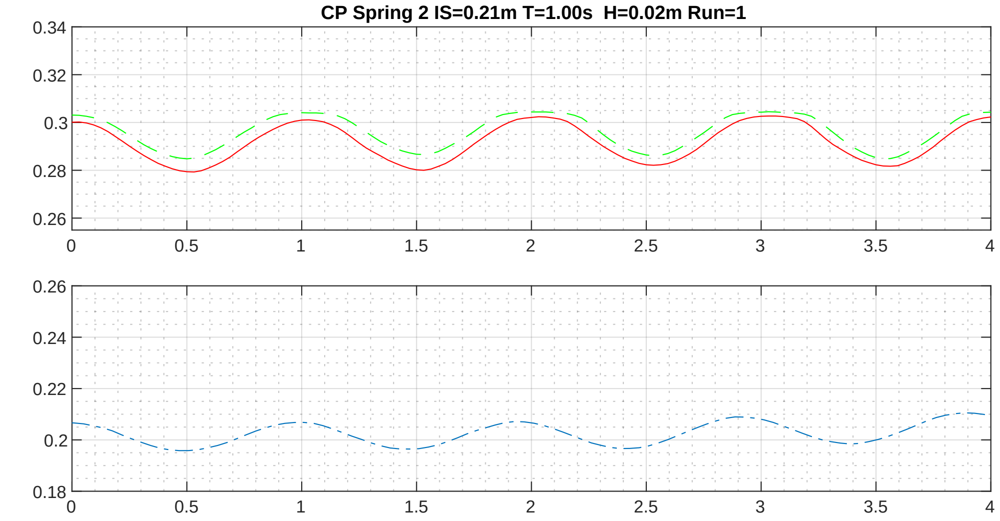

CP Spring 2 IS=0.21m T=0.70s H=0.03m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

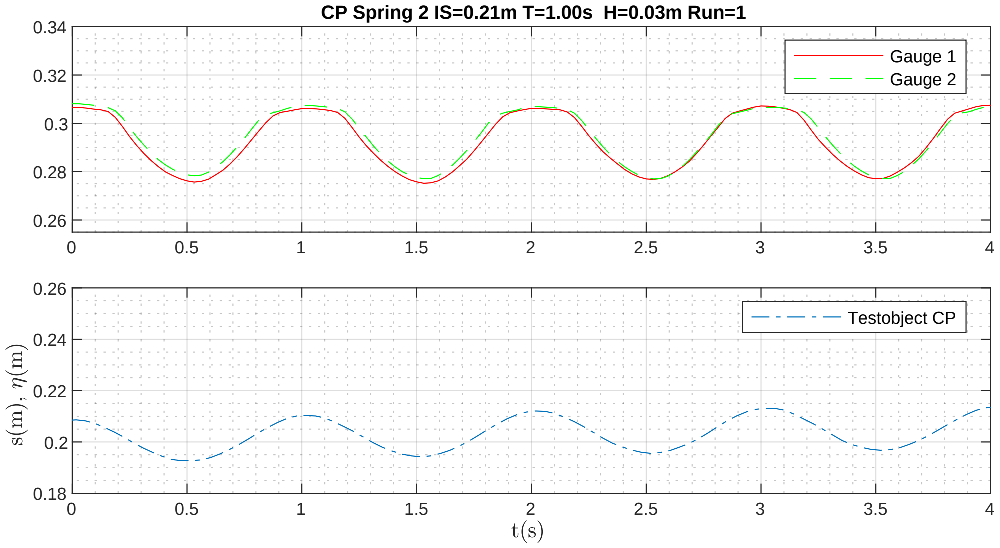

CP Spring 2 IS=0.21m T=0.70s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

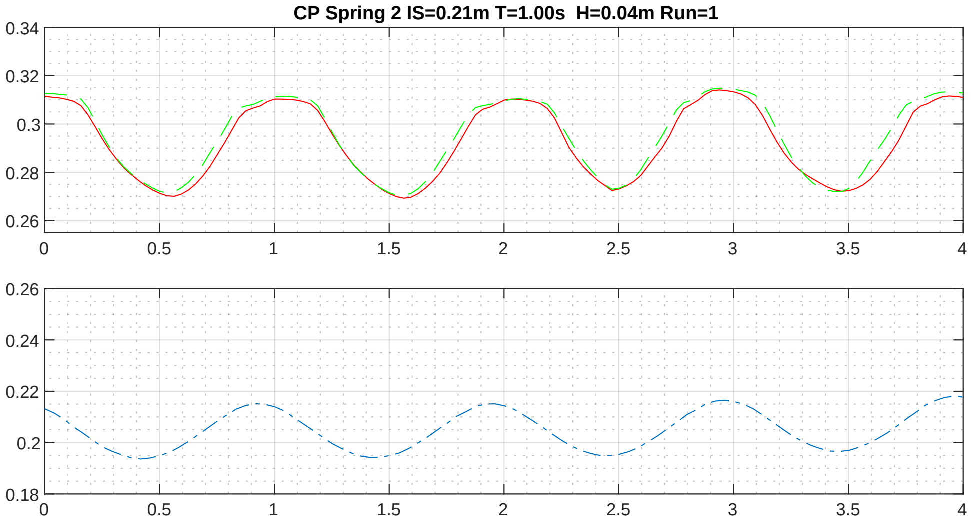

CP Spring 2 IS=0.21m T=0.70s H=0.05m Run=1
0.34
0.32
0.3
0.28
0.26
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

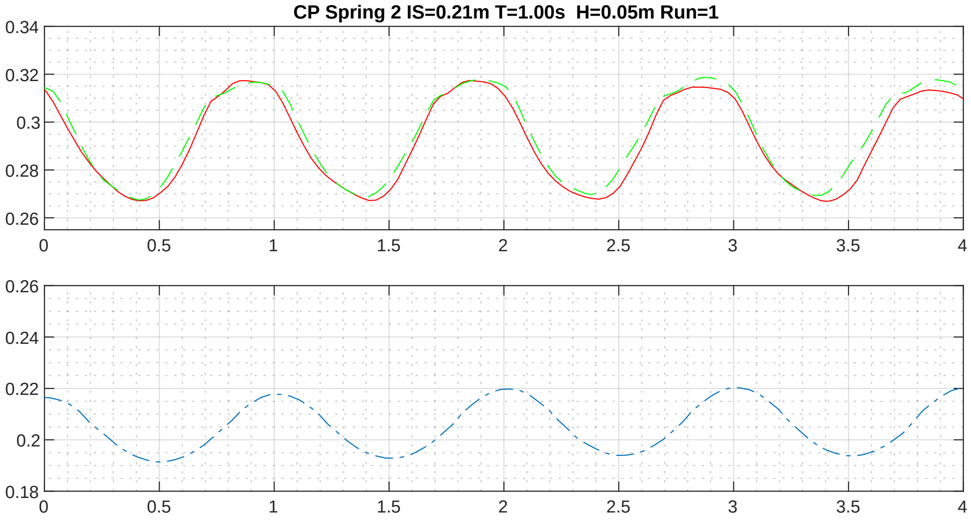

CP Spring 2 IS=0.21m T=0.70s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

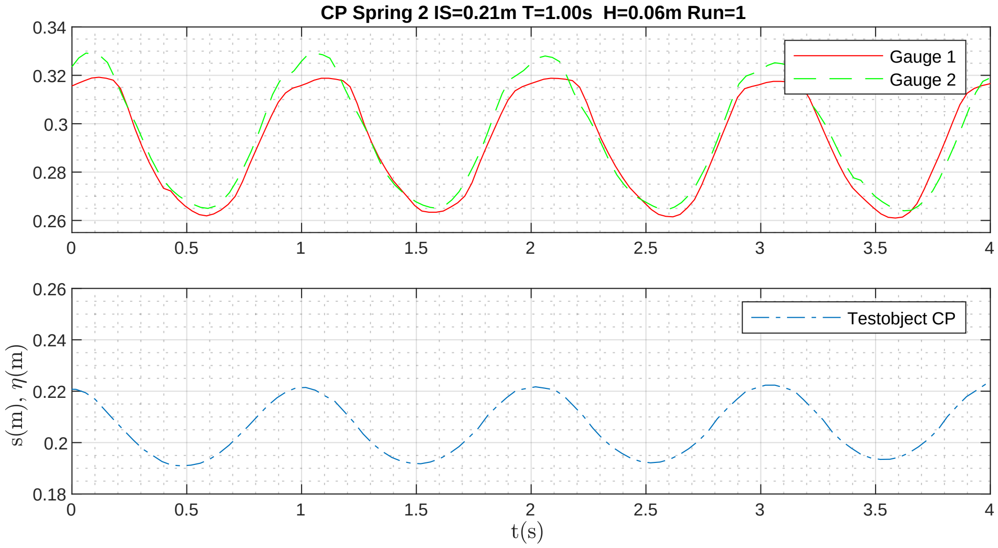

CP Spring 2 IS=0.21m T=1.00s H=0.01m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

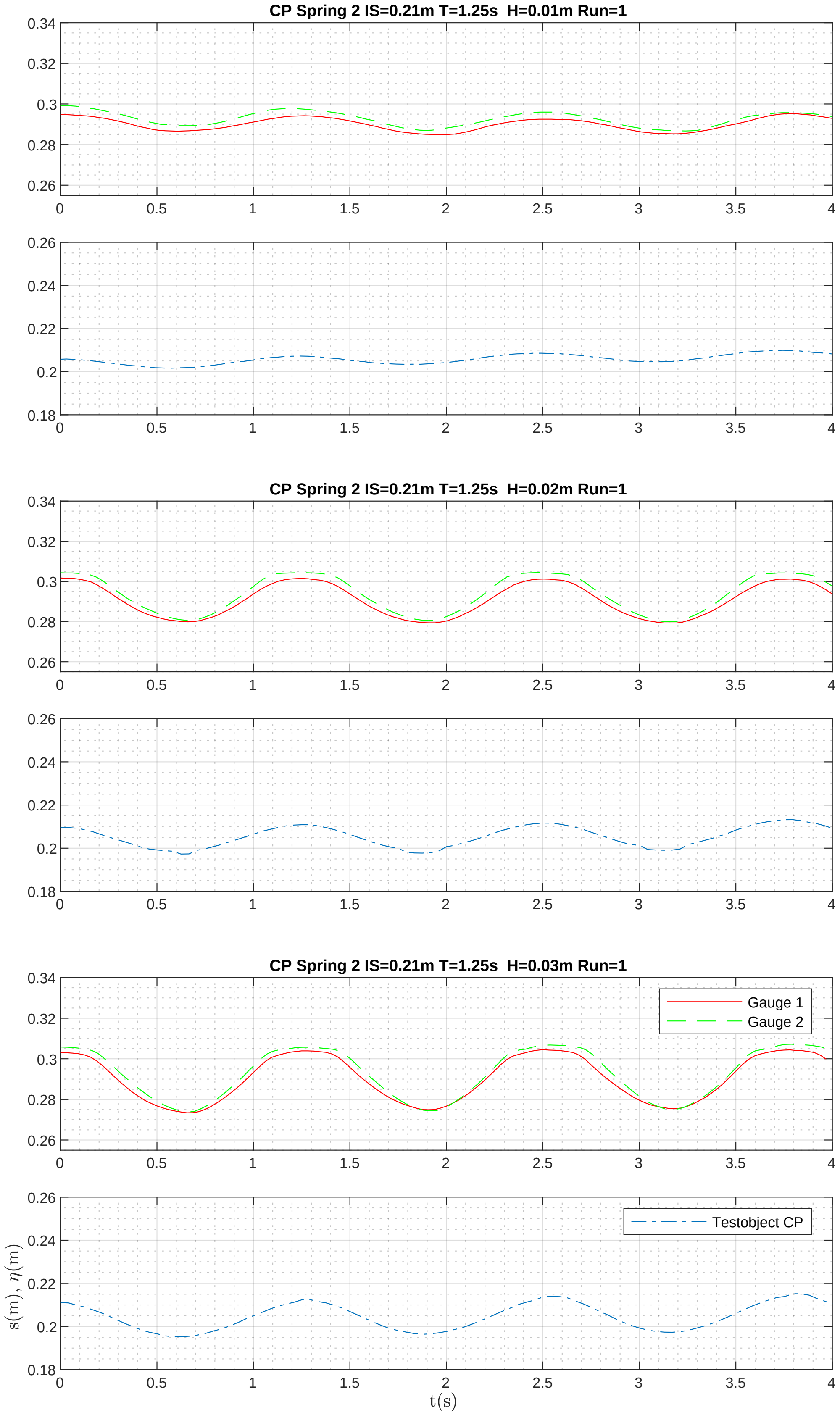

CP Spring 2 IS=0.21m T=1.00s H=0.02m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

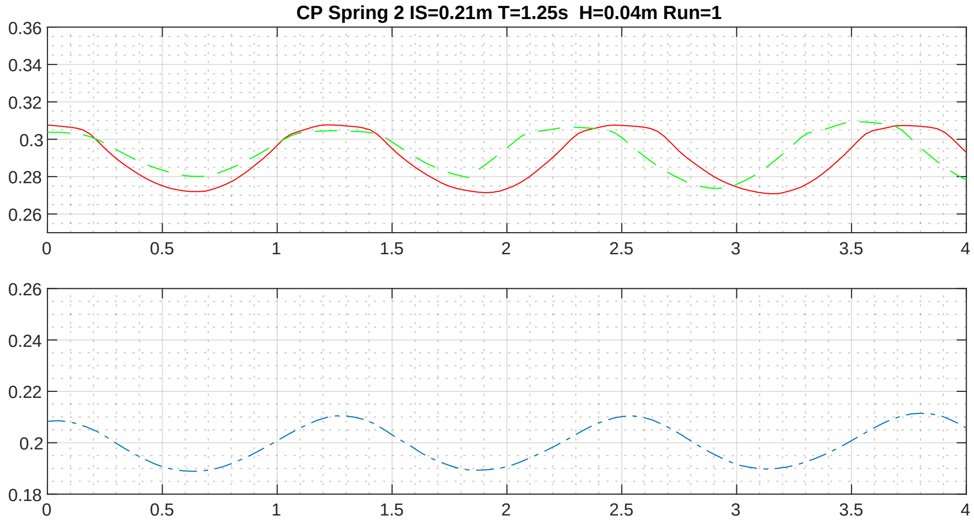

CP Spring 2 IS=0.21m T=1.00s H=0.03m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

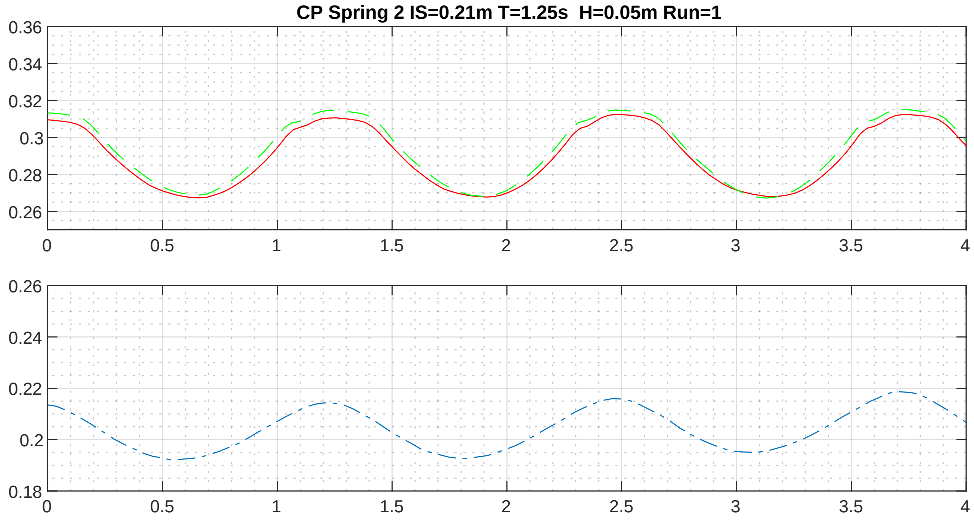

CP Spring 2 IS=0.21m T=1.00s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.22
0.24
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

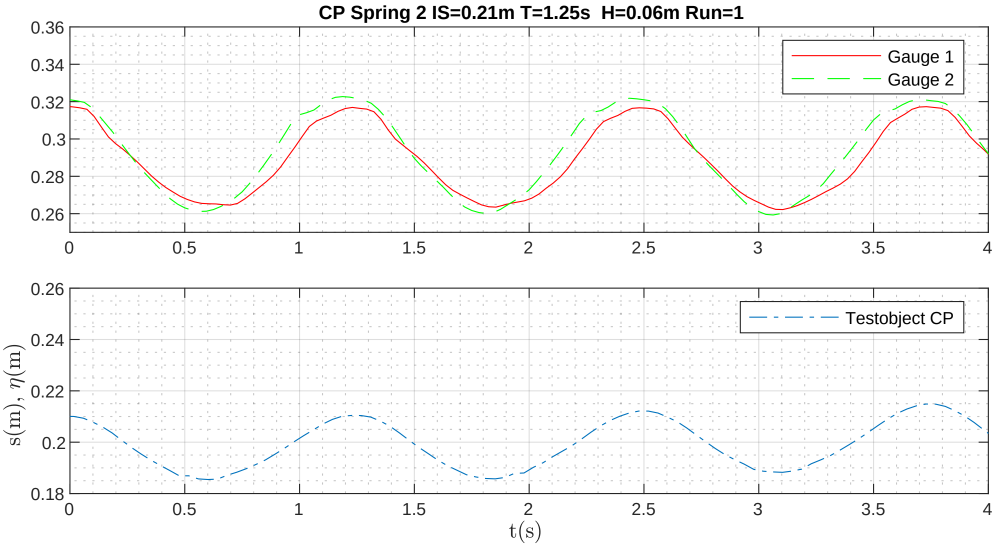

CP Spring 2 IS=0.21m T=1.00s H=0.05m Run=1
0.34
0.32
0.3
0.28
0.26
0.22
0.24
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

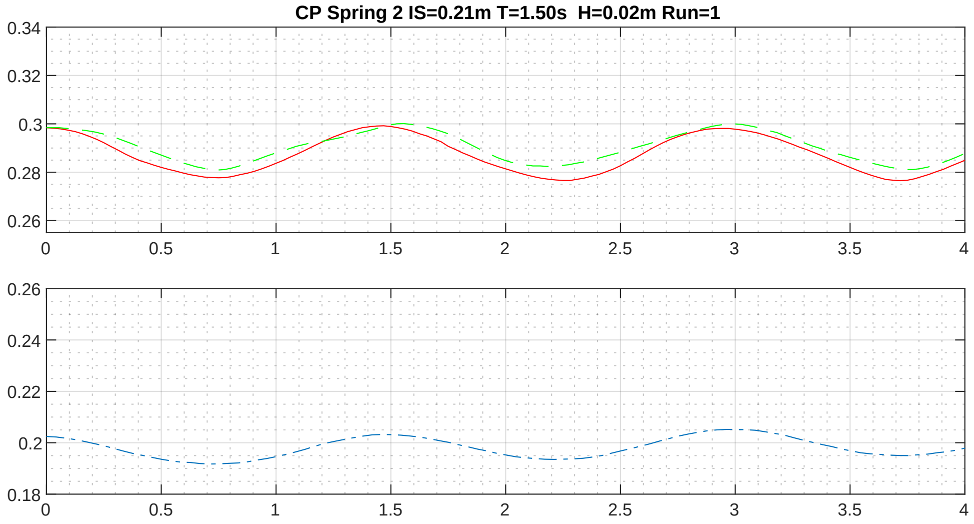

CP Spring 2 IS=0.21m T=1.00s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.22
0.24
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

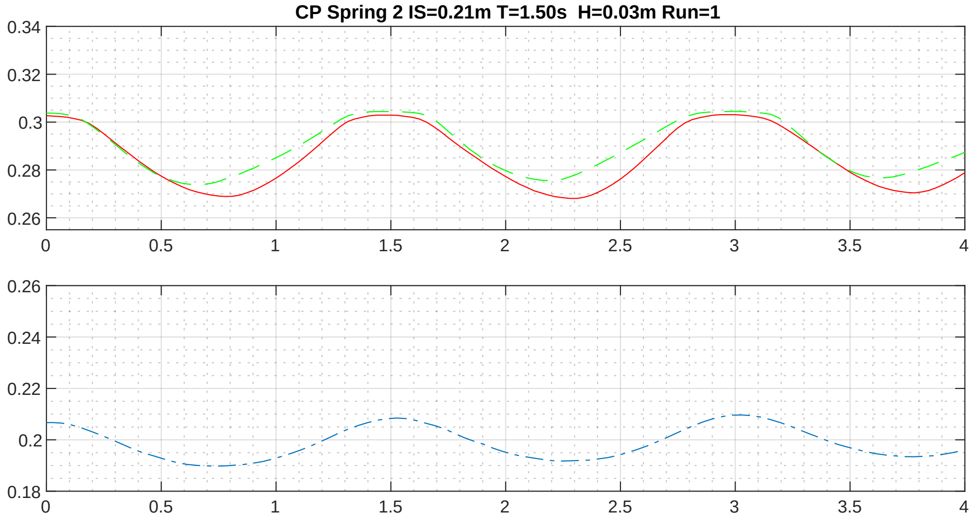

CP Spring 2 IS=0.21m T=1.25s H=0.01m Run=1
CP Spring 2 IS=0.21m T=1.25s H=0.02m Run=1
CP Spring 2 IS=0.21m T=1.25s H=0.03m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

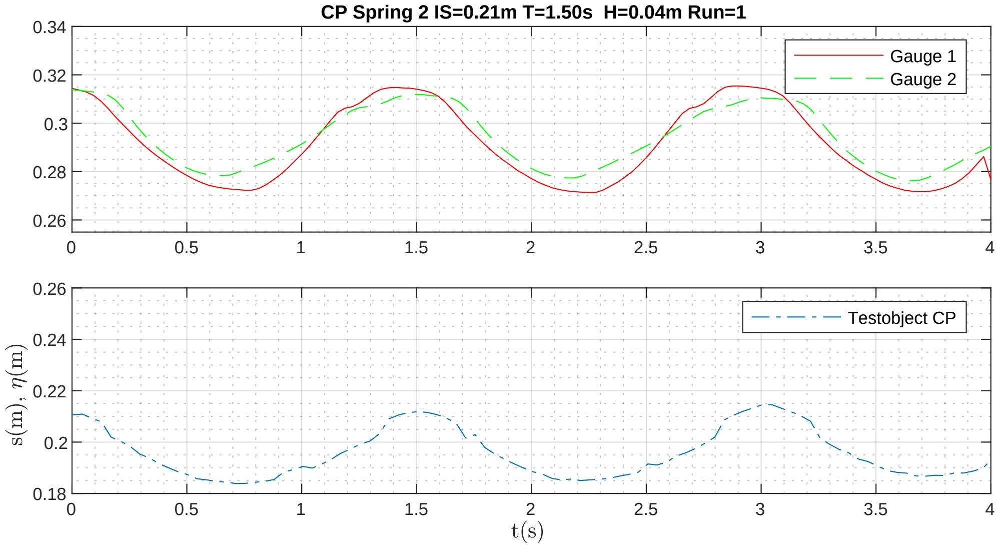

CP Spring 2 IS=0.21m T=1.25s H=0.04m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

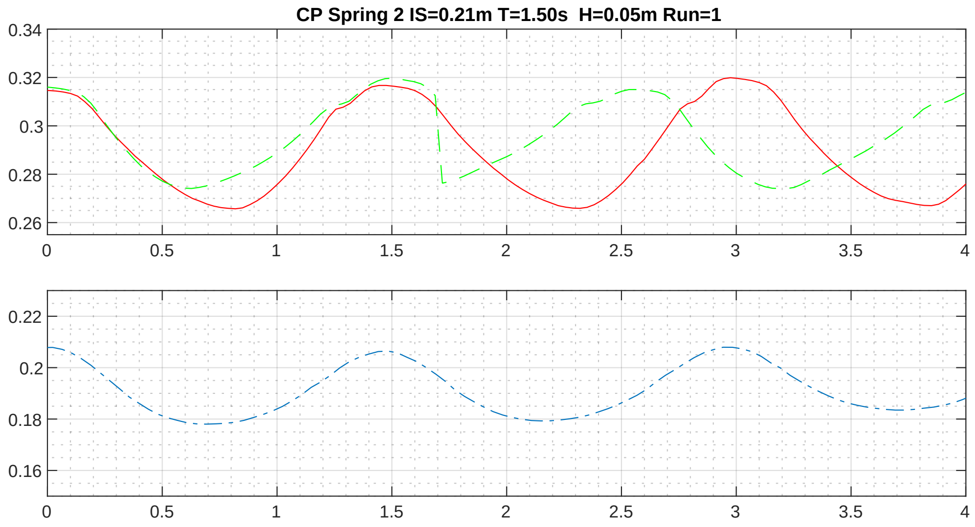

CP Spring 2 IS=0.21m T=1.25s H=0.05m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

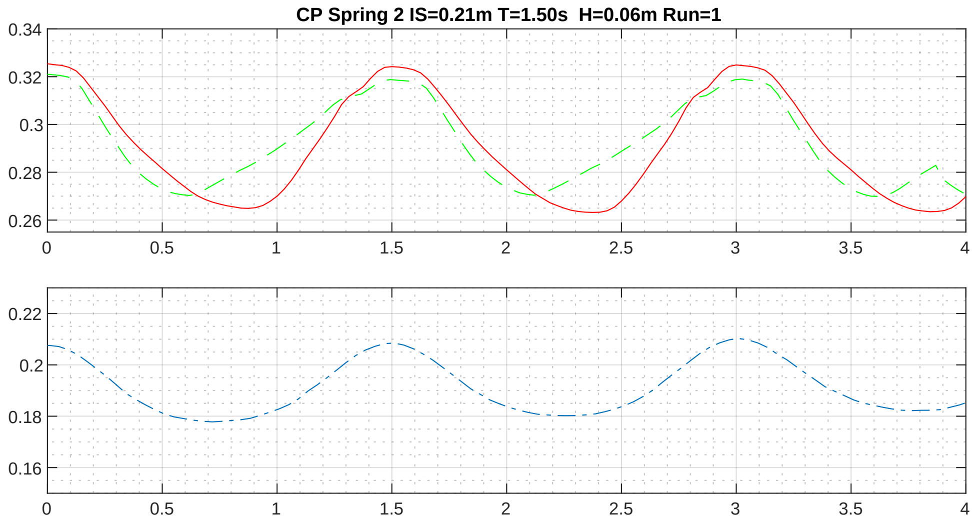

CP Spring 2 IS=0.21m T=1.25s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.36
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

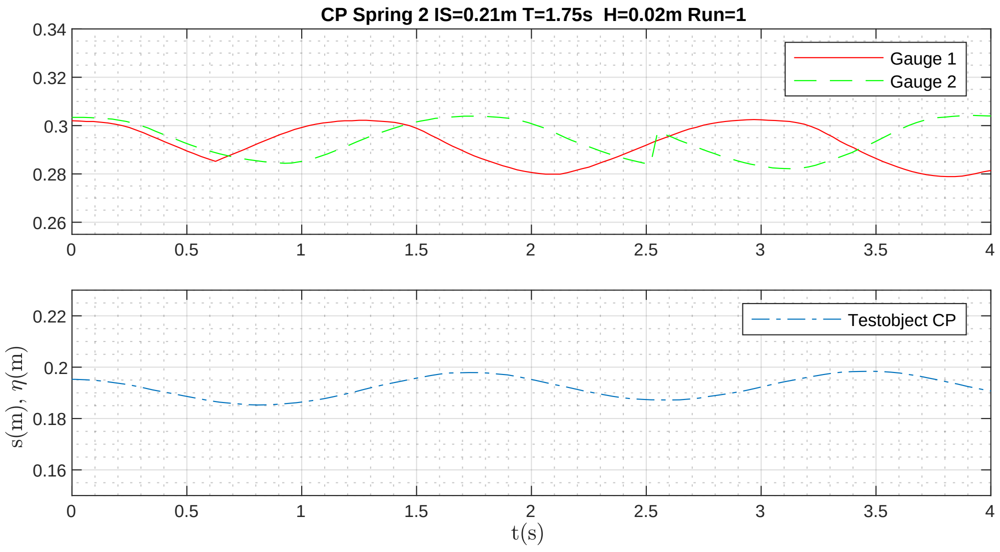

CP Spring 2 IS=0.21m T=1.50s H=0.02m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

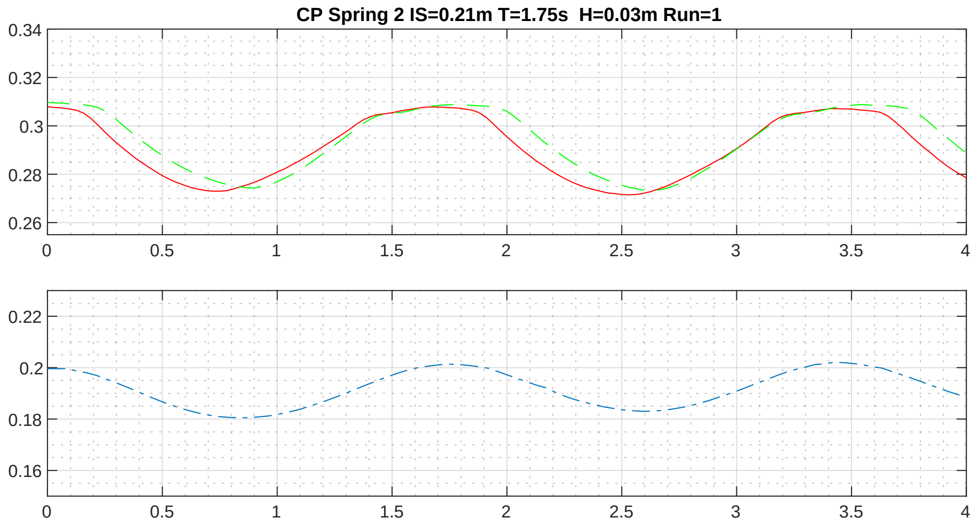

CP Spring 2 IS=0.21m T=1.50s H=0.03m Run=1
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4

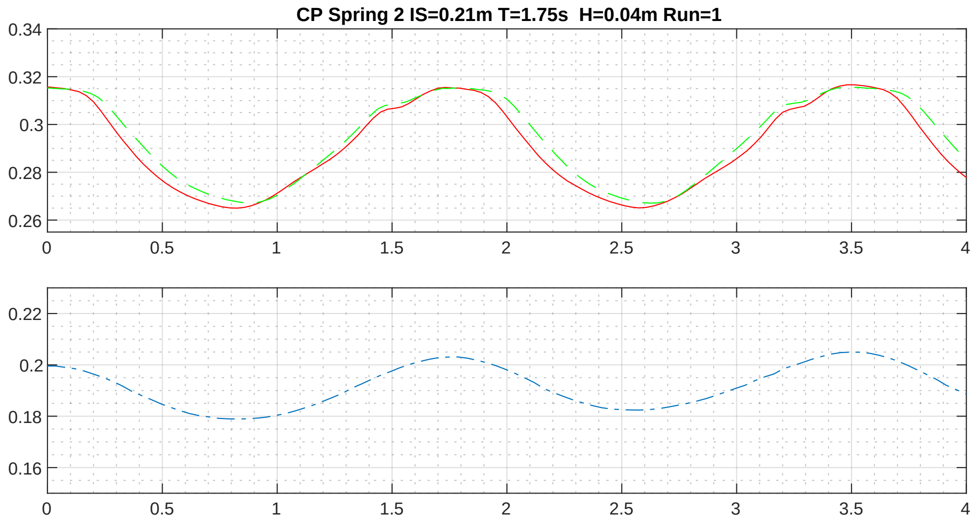

CP Spring 2 IS=0.21m T=1.50s H=0.04m Run=1
Gauge 1
Gauge 2
Testobject CP
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0
0.5
1
1.5
2
2.5
3
3.5
4
s(m), η(m)
t(s)

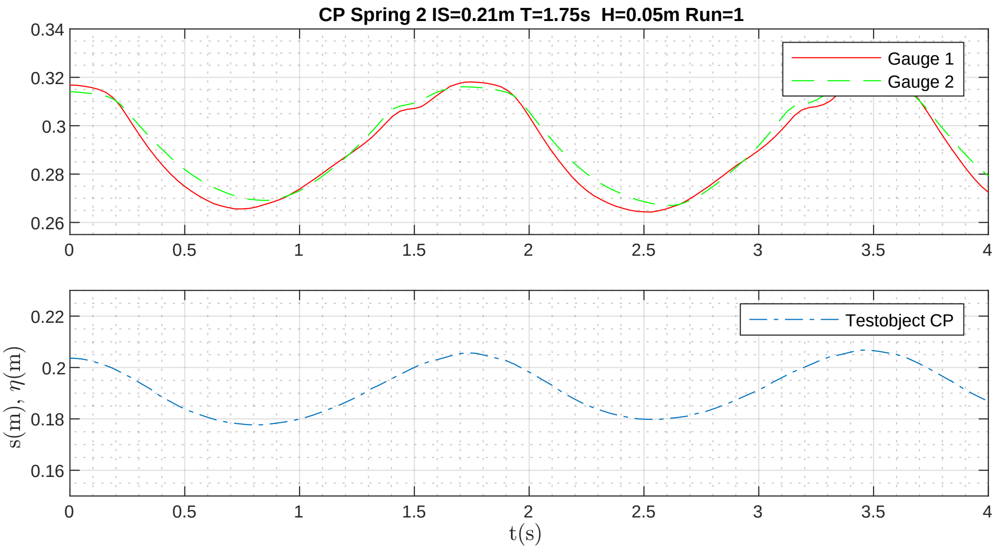

CP Spring 2 IS=0.21m T=1.50s H=0.05m Run=1
0.34
0.32
0.3
0.28
0.26
0.22
0.2
0.18
0.16
0
0.5
1
1.5
2
2.5
3
3.5
4

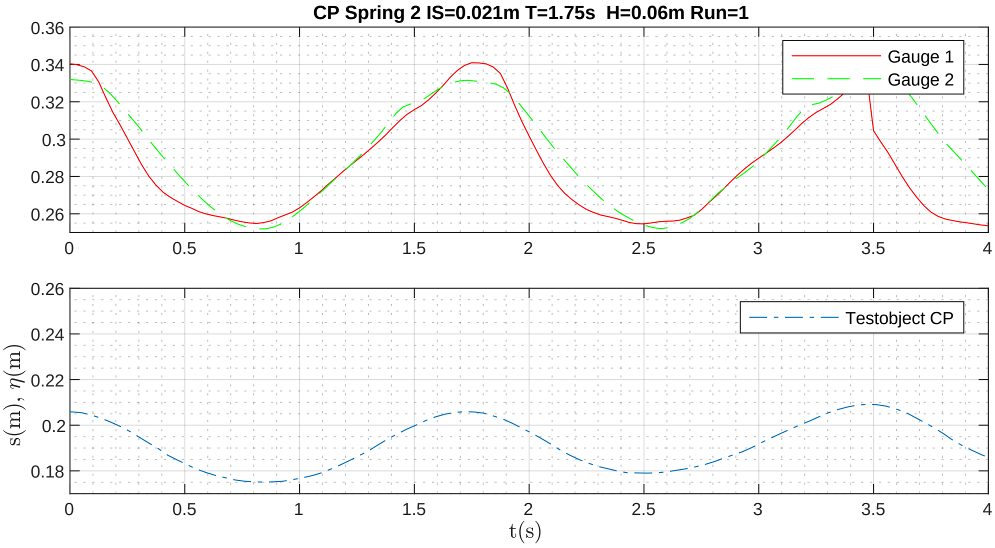

CP Spring 2 IS=0.21m T=1.50s H=0.06m Run=1
0.34
0.32
0.3
0.28
0.26
0.22
0.2
0.18
0.16
0
0.5
1
1.5
2
2.5
3
3.5
4

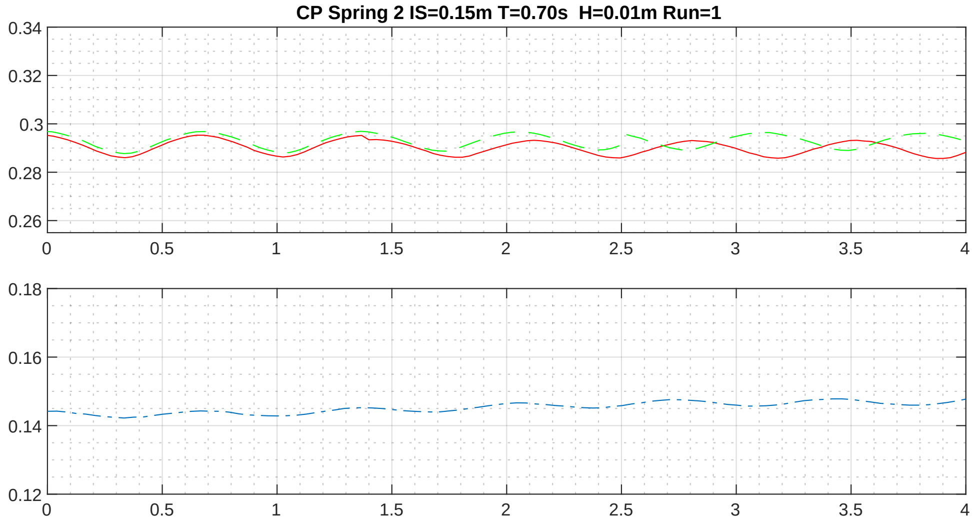

CP Spring 2 IS=0.21m T=1.75s H=0.02m Run=1
Gauge 1
Gauge 2
Testobject CP
0.34
0.32
0.3
0.28
0.26
0.22
0.2
0.18
0.16
s(m), η(m)
0
0.5
1
1.5
2
2.5
3
3.5
4
t(s)

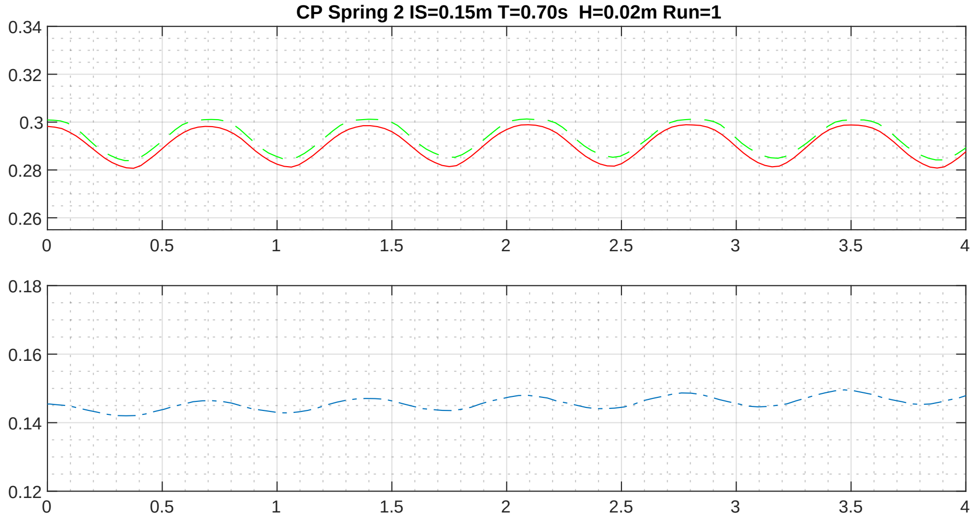

CP Spring 2 IS=0.21m T=1.75s H=0.03m Run=1
0.34
0.32
0.3
0.28
0.26
0.22
0.2
0.18
0.16
0
0.5
1
1.5
2
2.5
3
3.5
4

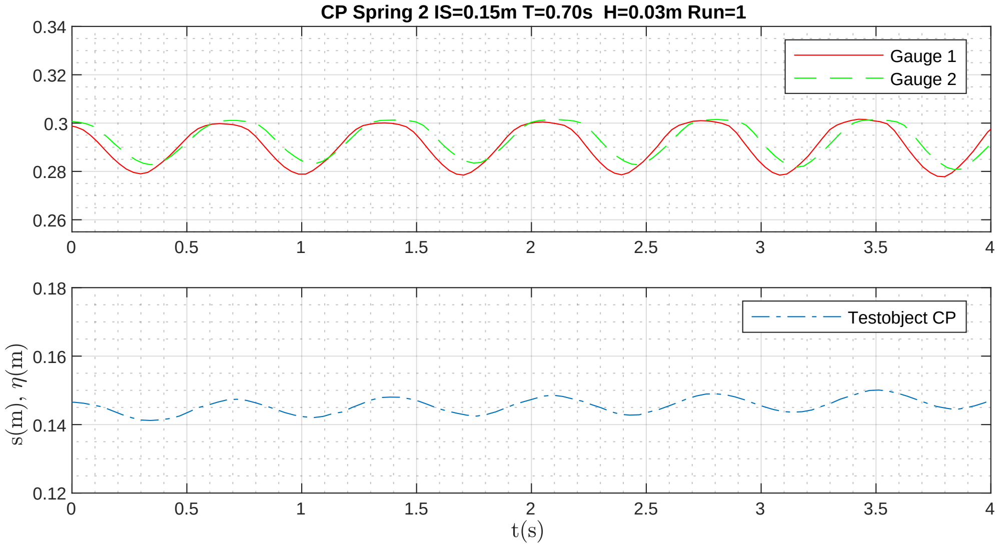

CP Spring 2 IS=0.21m T=1.75s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.22
0.2
0.18
0.16
0
0.5
1
1.5
2
2.5
3
3.5
4

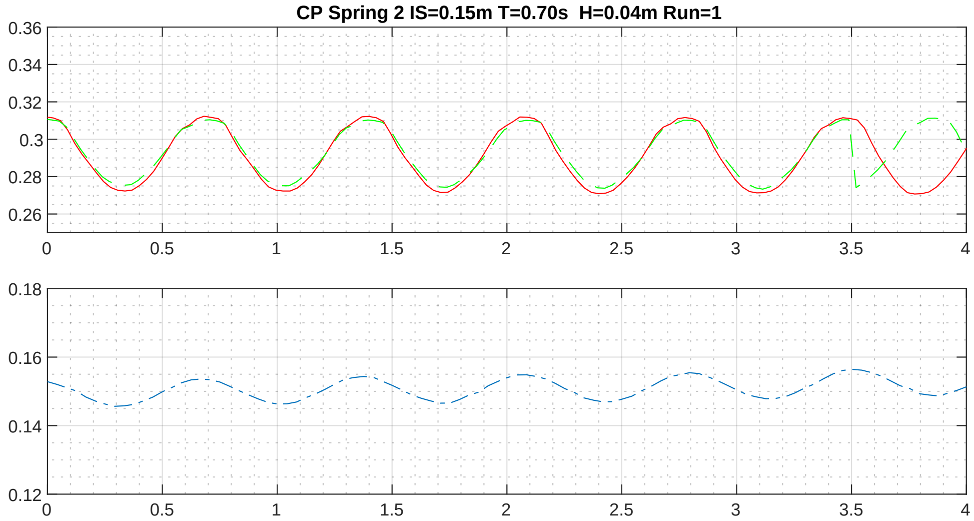

CP Spring 2 IS=0.21m T=1.75s H=0.05m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.22
0.2
0.18
0.16
0
0.5
1
1.5
2
2.5
3
3.5
4

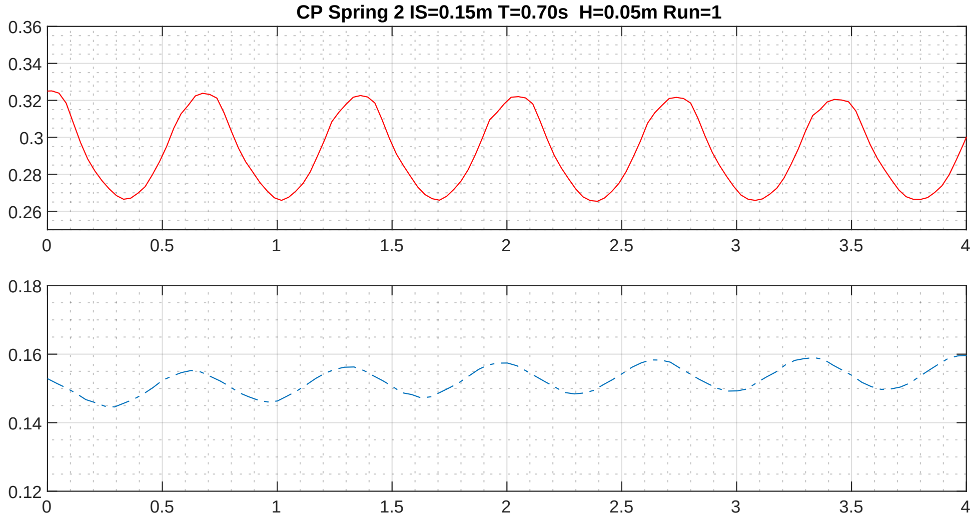

CP Spring 2 IS=0.021m T=1.75s H=0.06m Run=1
Gauge 1
Gauge 2
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
Testobject CP
0.26
0.24
0.22
0.2
0.18
s(m), η(m)
t(s)

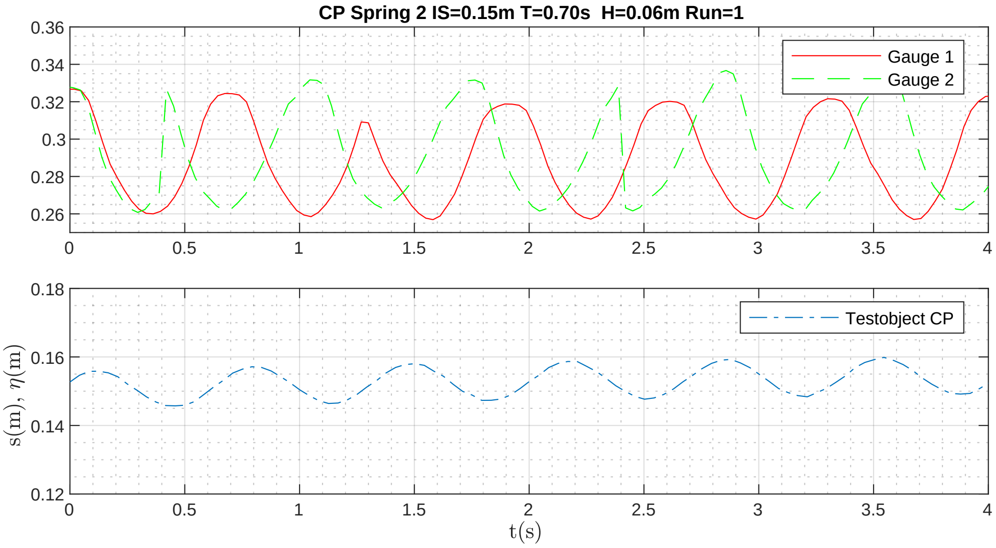

CP Spring 2 IS=0.15m T=0.70s H=0.01m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

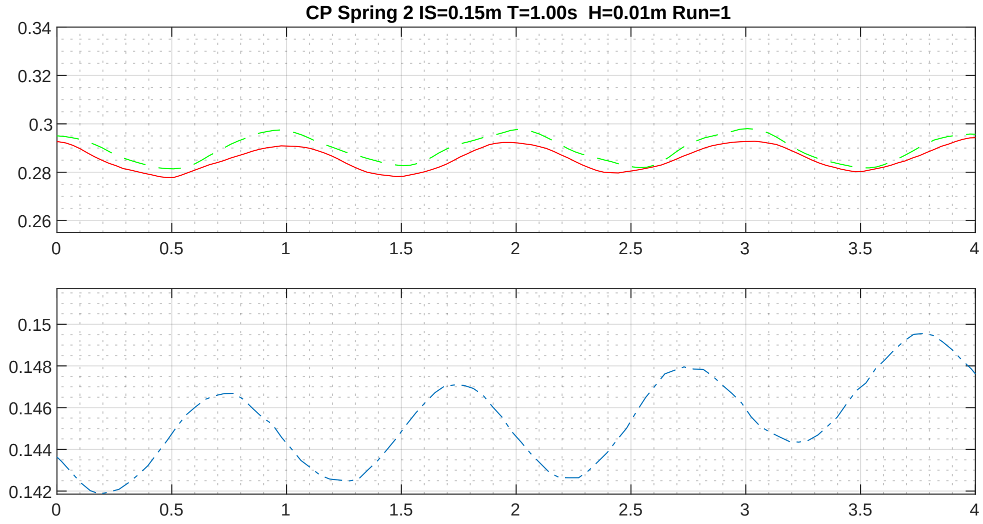

CP Spring 2 IS=0.15m T=0.70s H=0.02m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

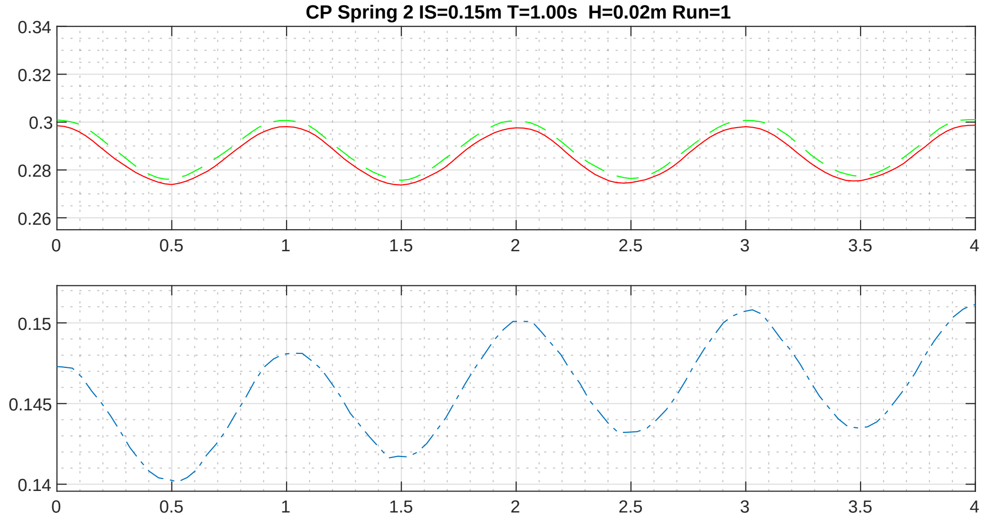

CP Spring 2 IS=0.15m T=0.70s H=0.03m Run=1
Gauge 1
Gauge 2
Testobject CP
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4
s(m), η(m)
t(s)

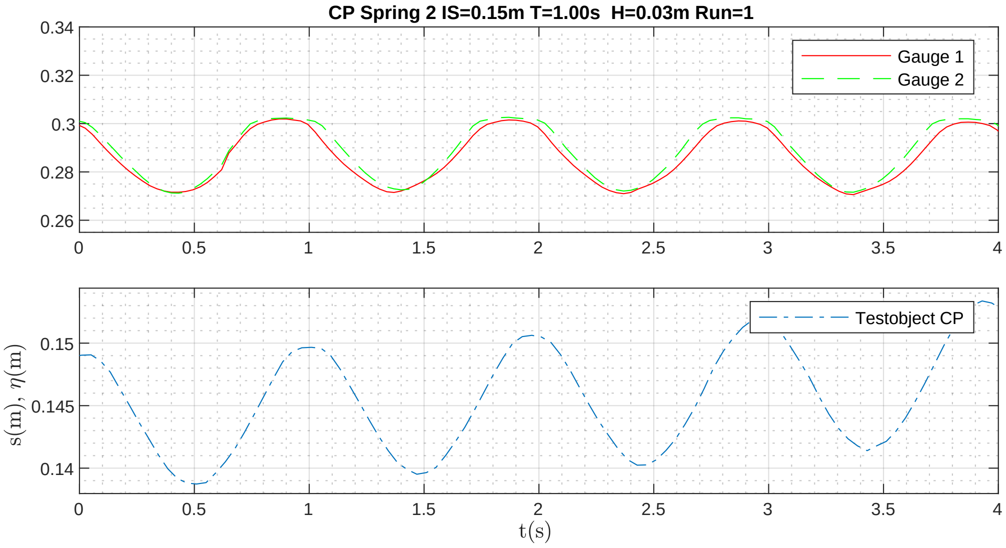

CP Spring 2 IS=0.15m T=0.70s H=0.04m Run=1

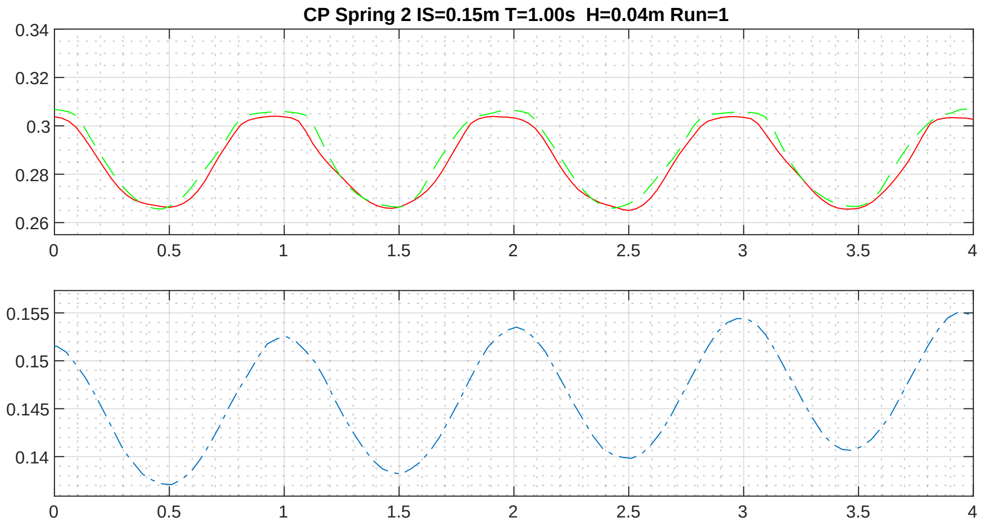

CP Spring 2 IS=0.15m T=0.70s H=0.05m Run=1

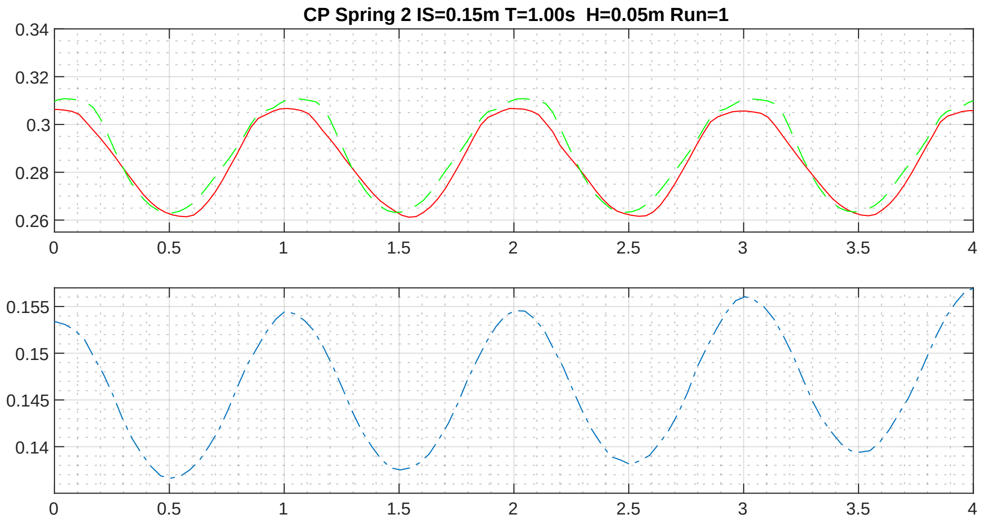

CP Spring 2 IS=0.15m T=0.70s H=0.06m Run=1

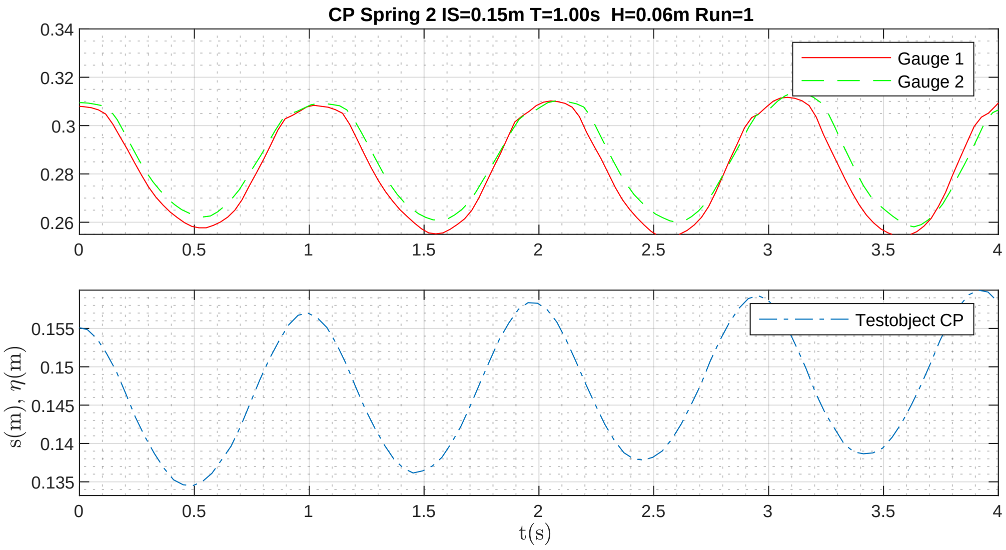

CP Spring 2 IS=0.15m T=1.00s H=0.01m Run=1

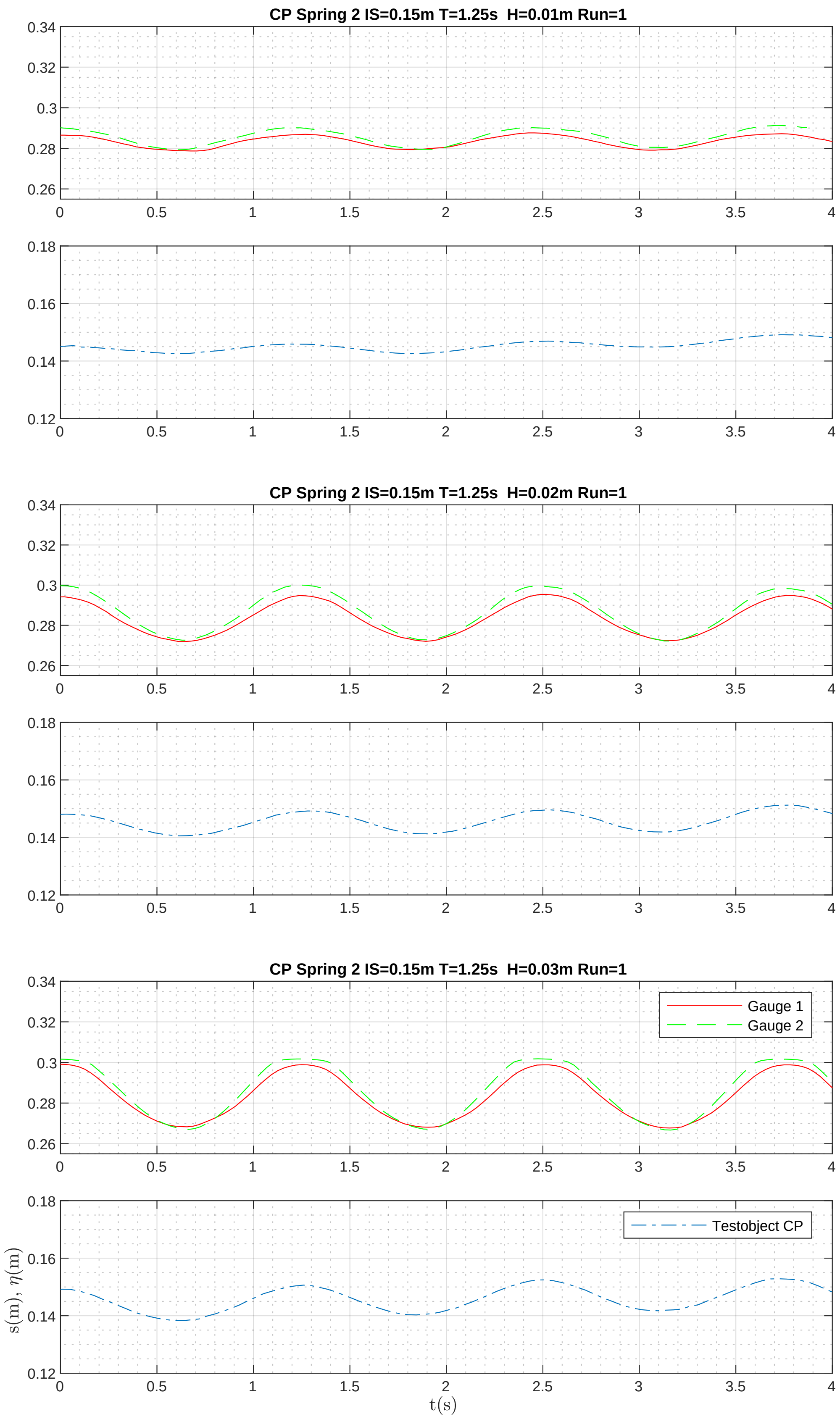

CP Spring 2 IS=0.15m T=1.00s H=0.02m Run=1

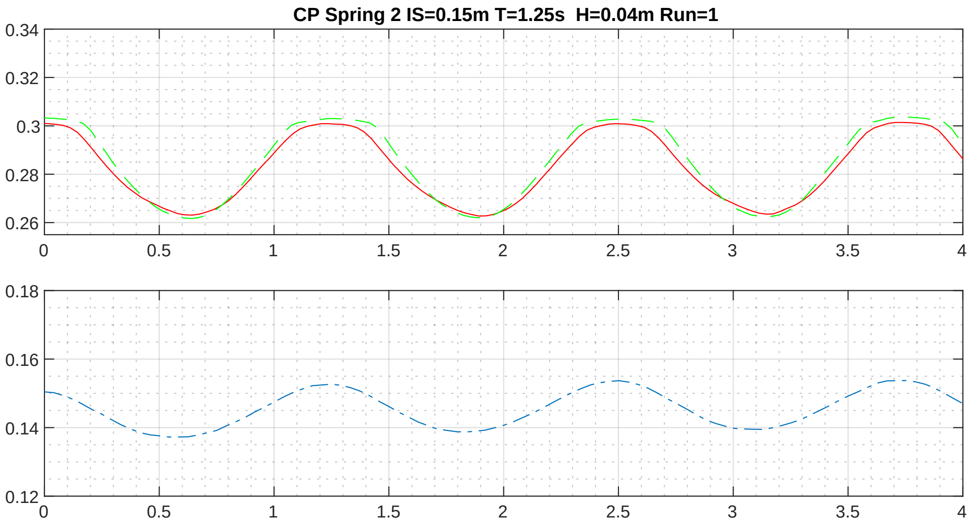

CP Spring 2 IS=0.15m T=1.00s H=0.03m Run=1

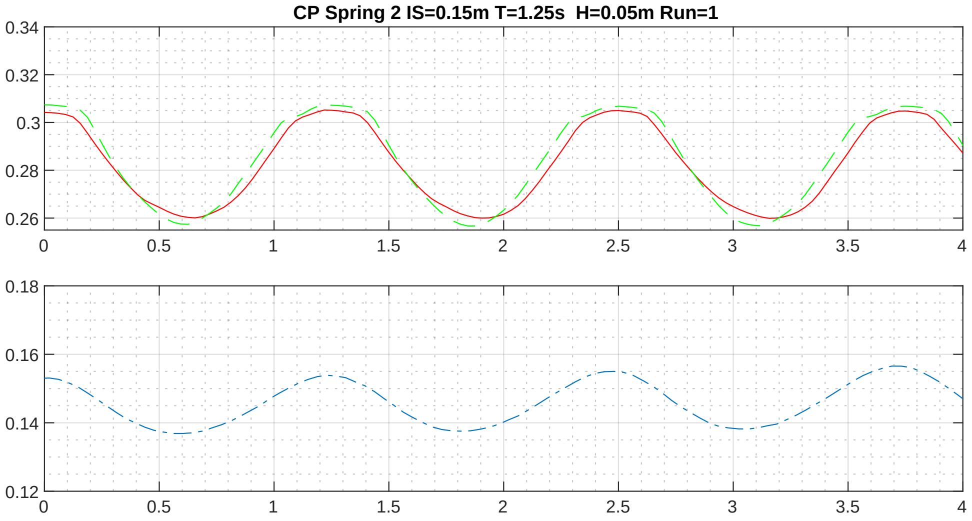

CP Spring 2 IS=0.15m T=1.00s H=0.04m Run=1

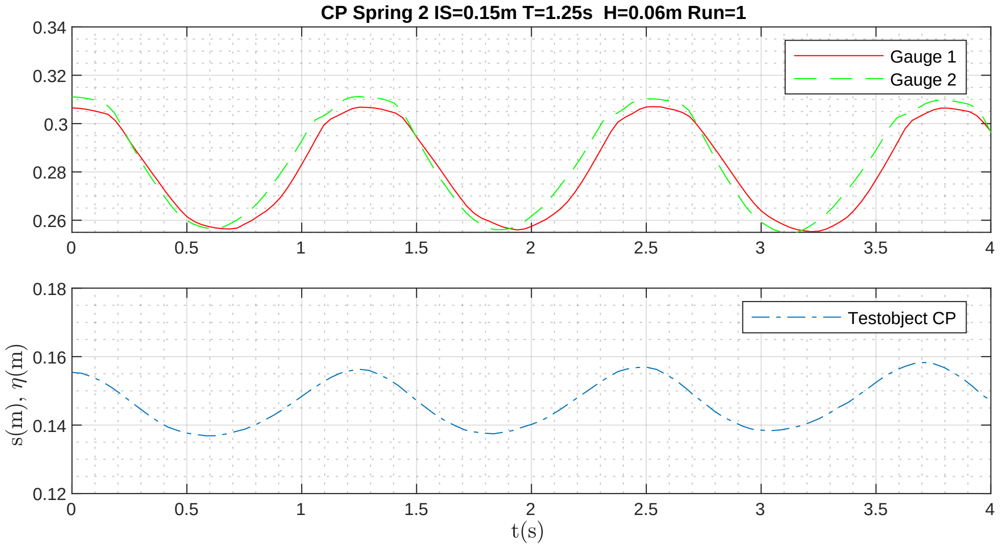

CP Spring 2 IS=0.15m T=1.00s H=0.05m Run=1

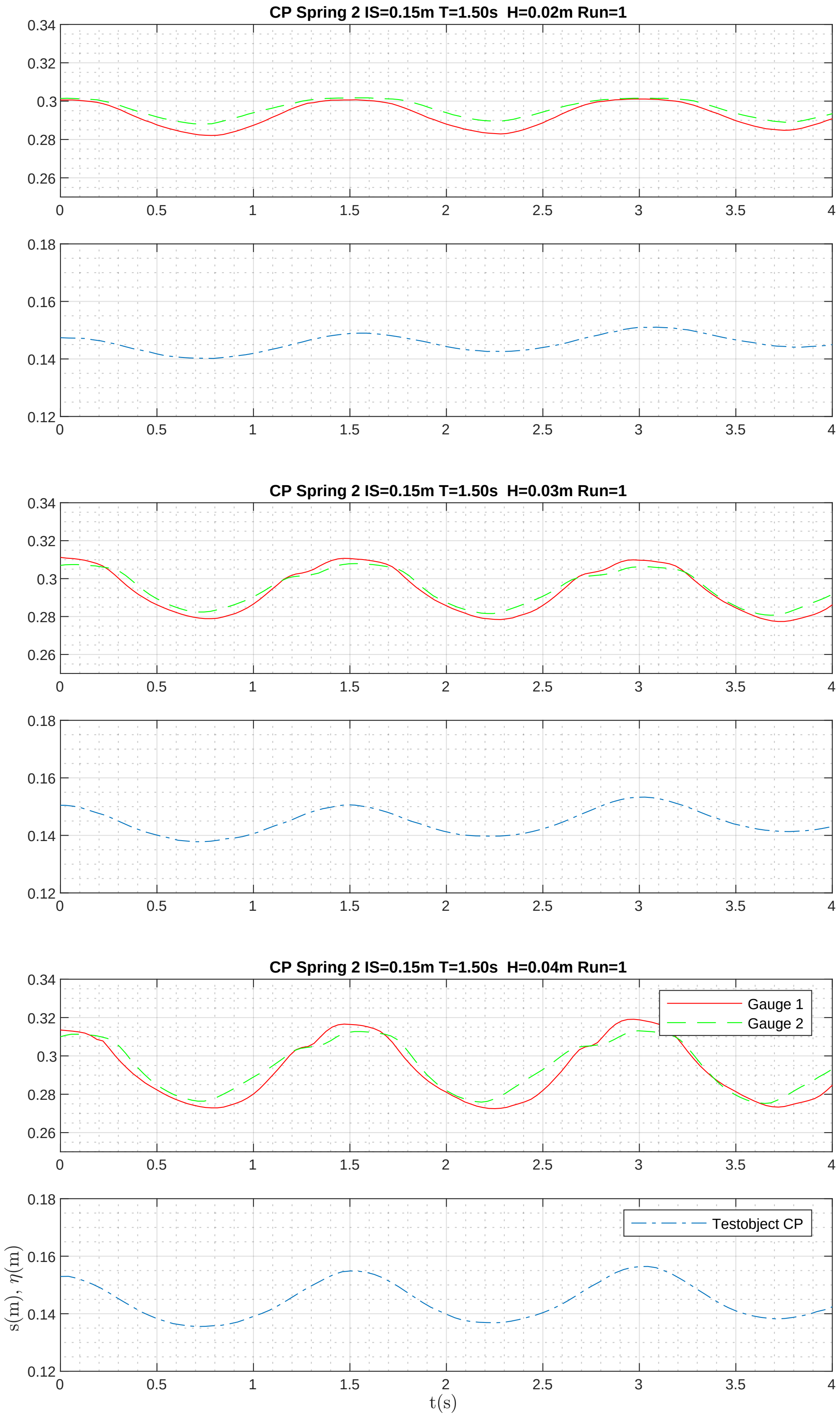

CP Spring 2 IS=0.15m T=1.00s H=0.06m Run=1

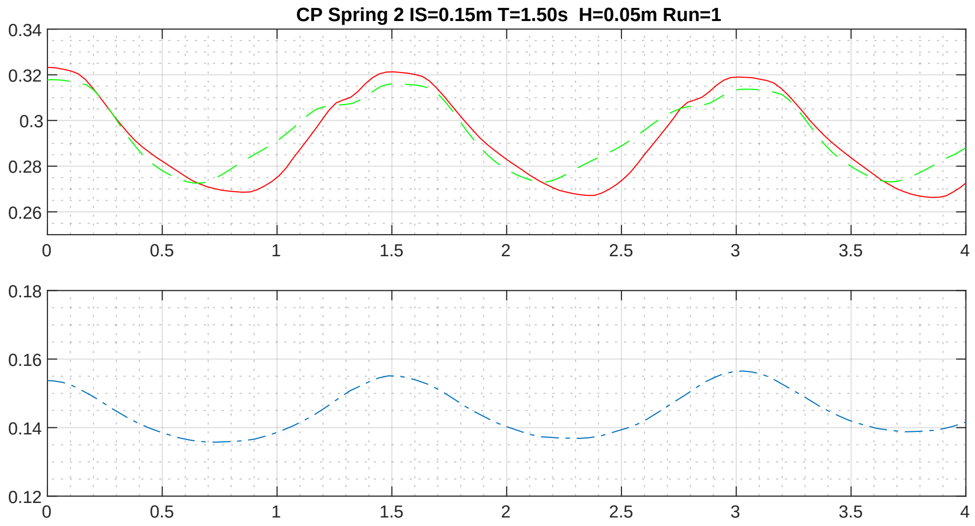

CP Spring 2 IS=0.15m T=1.25s H=0.01m Run=1
CP Spring 2 IS=0.15m T=1.25s H=0.02m Run=1
CP Spring 2 IS=0.15m T=1.25s H=0.03m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

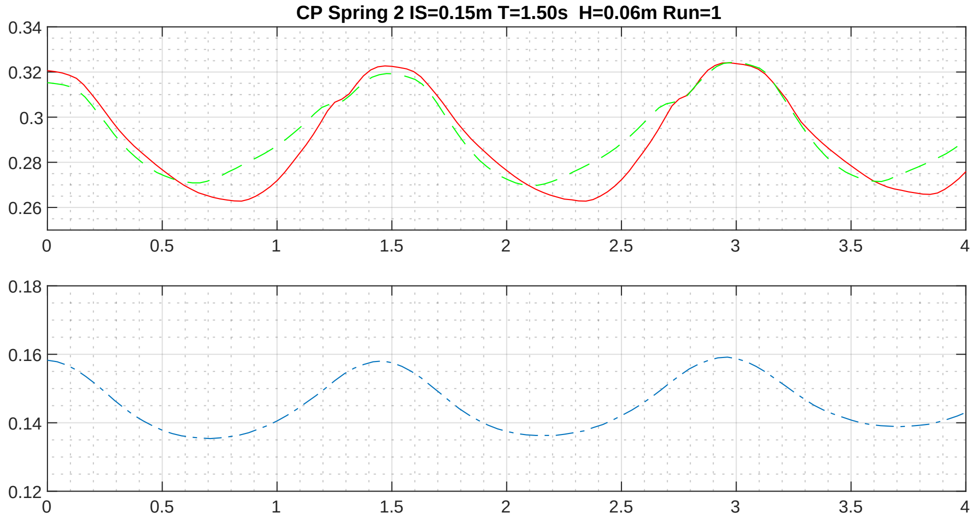

CP Spring 2 IS=0.15m T=1.25s H=0.04m Run=1

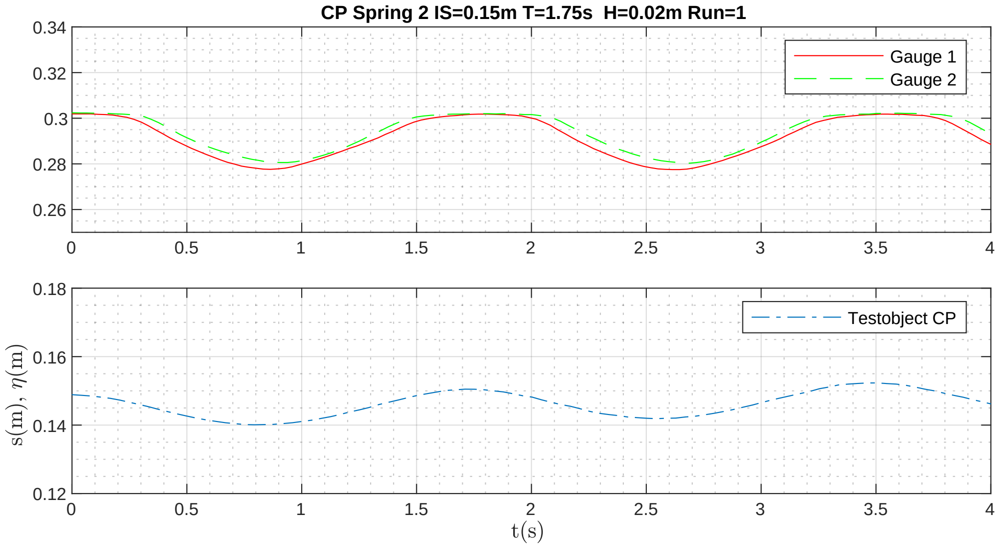

CP Spring 2 IS=0.15m T=1.25s H=0.05m Run=1

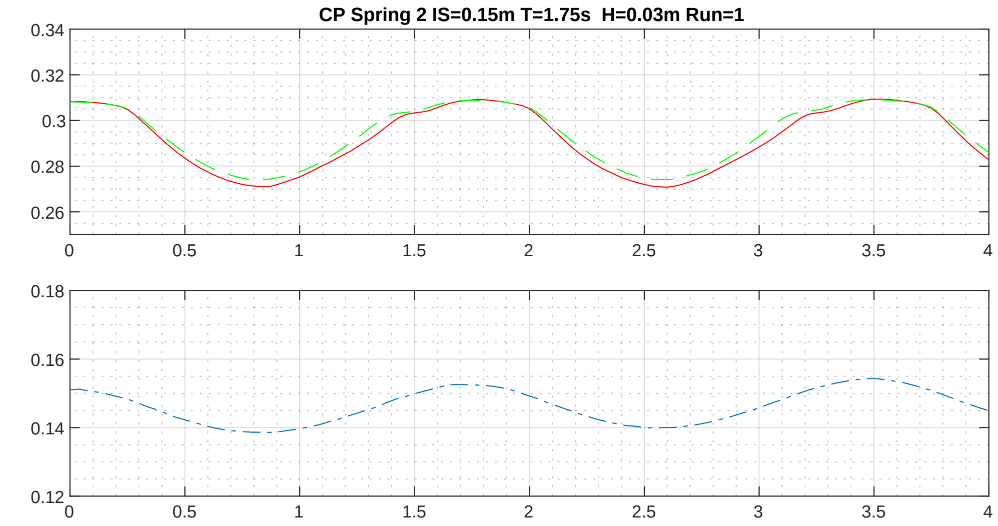

CP Spring 2 IS=0.15m T=1.25s H=0.06m Run=1

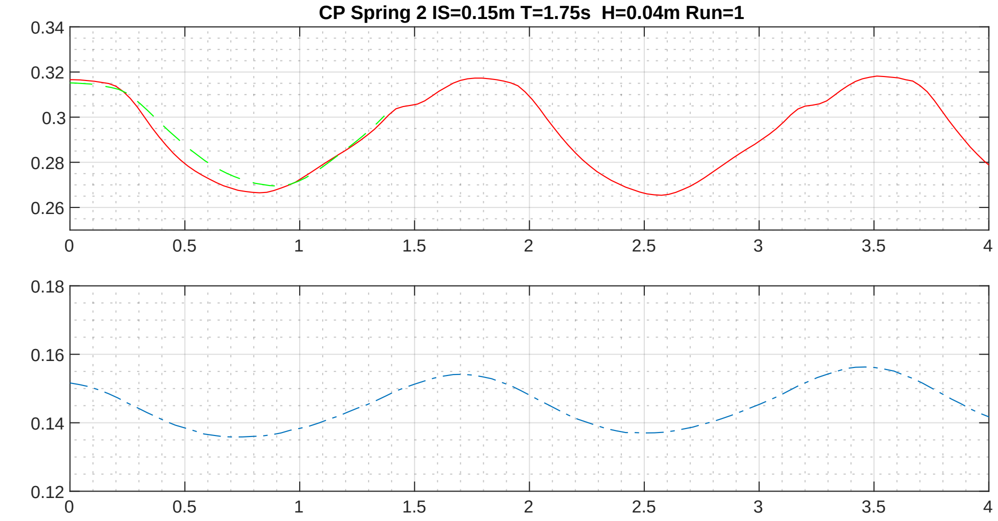

CP Spring 2 IS=0.15m T=1.50s H=0.02m Run=1
CP Spring 2 IS=0.15m T=1.50s H=0.03m Run=1
CP Spring 2 IS=0.15m T=1.50s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

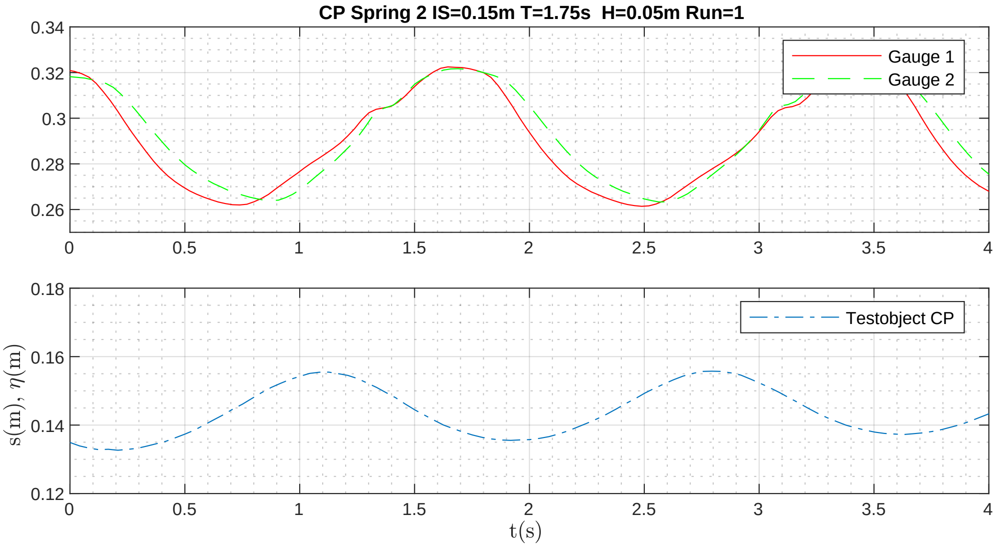

CP Spring 2 IS=0.15m T=1.50s H=0.05m Run=1

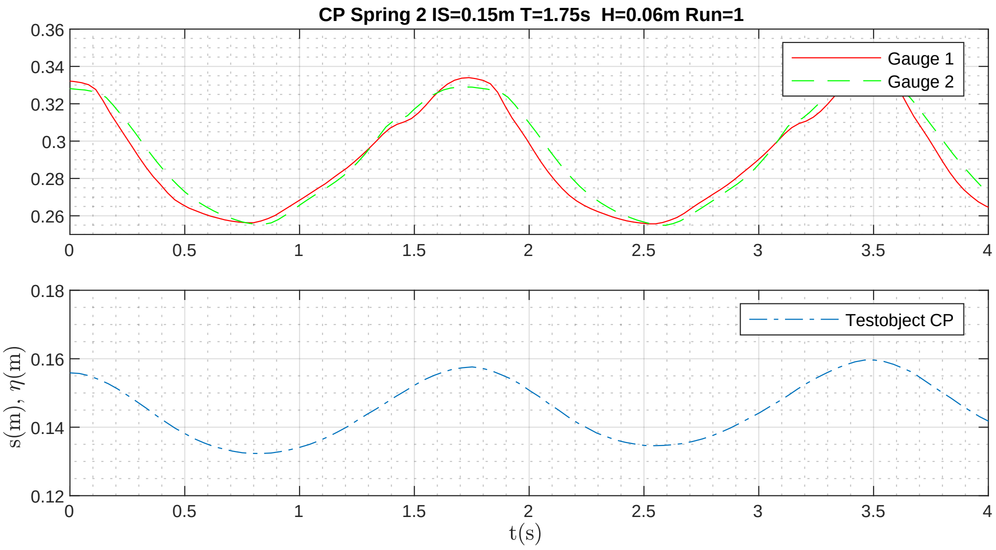

CP Spring 2 IS=0.15m T=1.50s H=0.06m Run=1

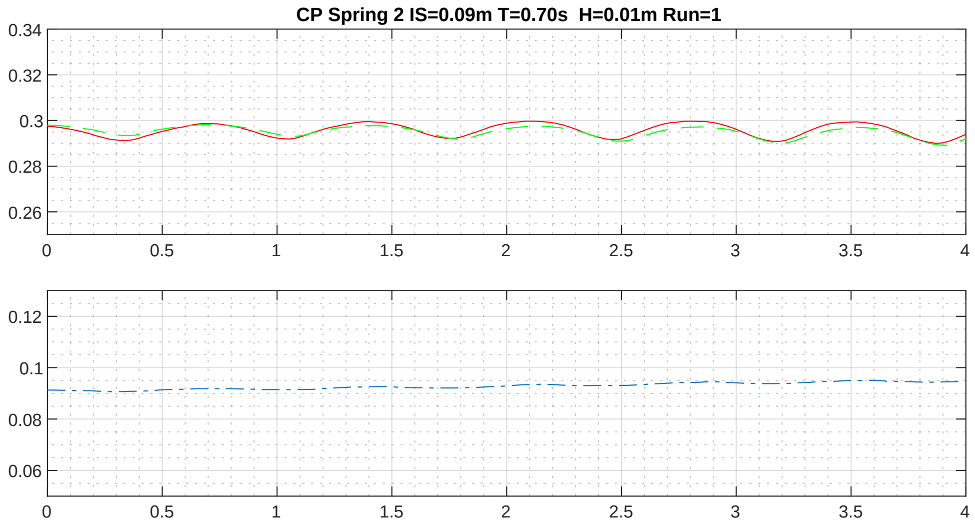

CP Spring 2 IS=0.15m T=1.75s H=0.02m Run=1

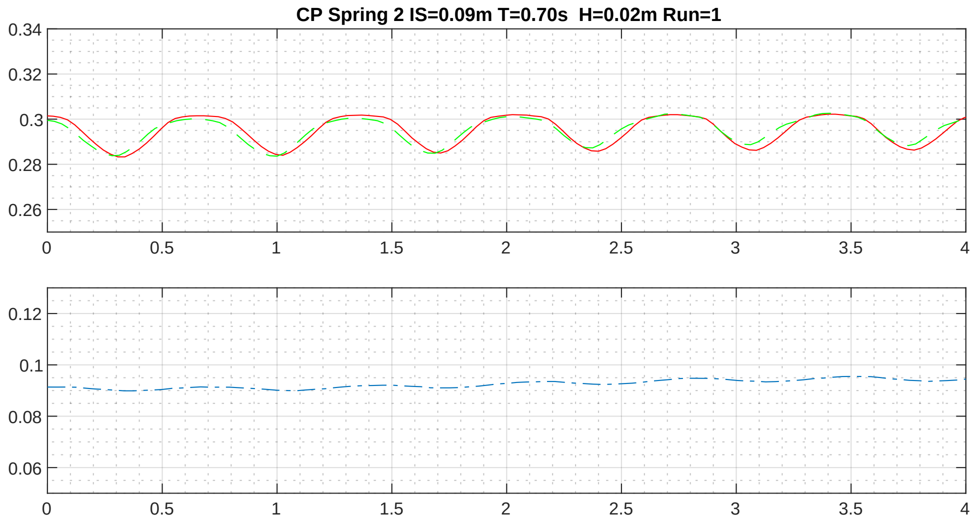

CP Spring 2 IS=0.15m T=1.75s H=0.03m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

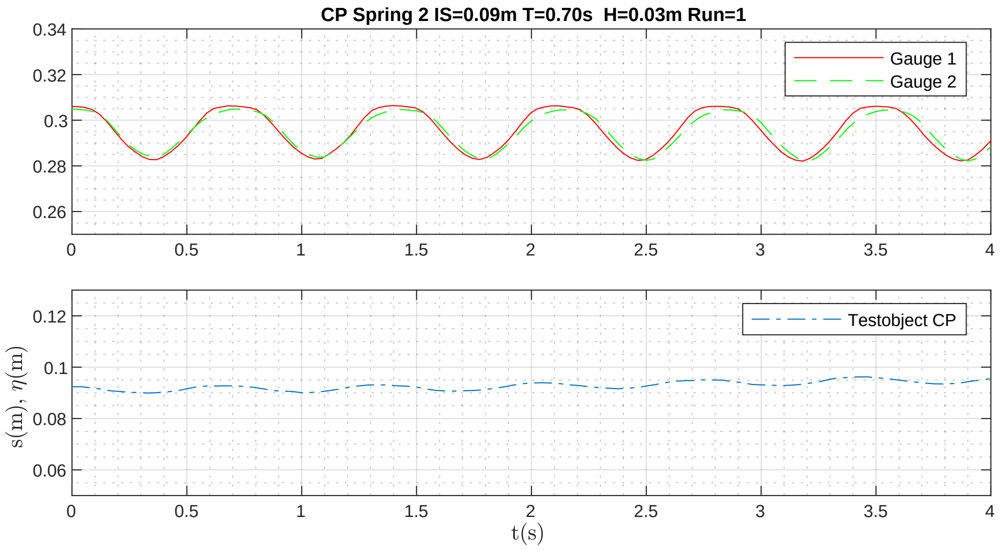

CP Spring 2 IS=0.15m T=1.75s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

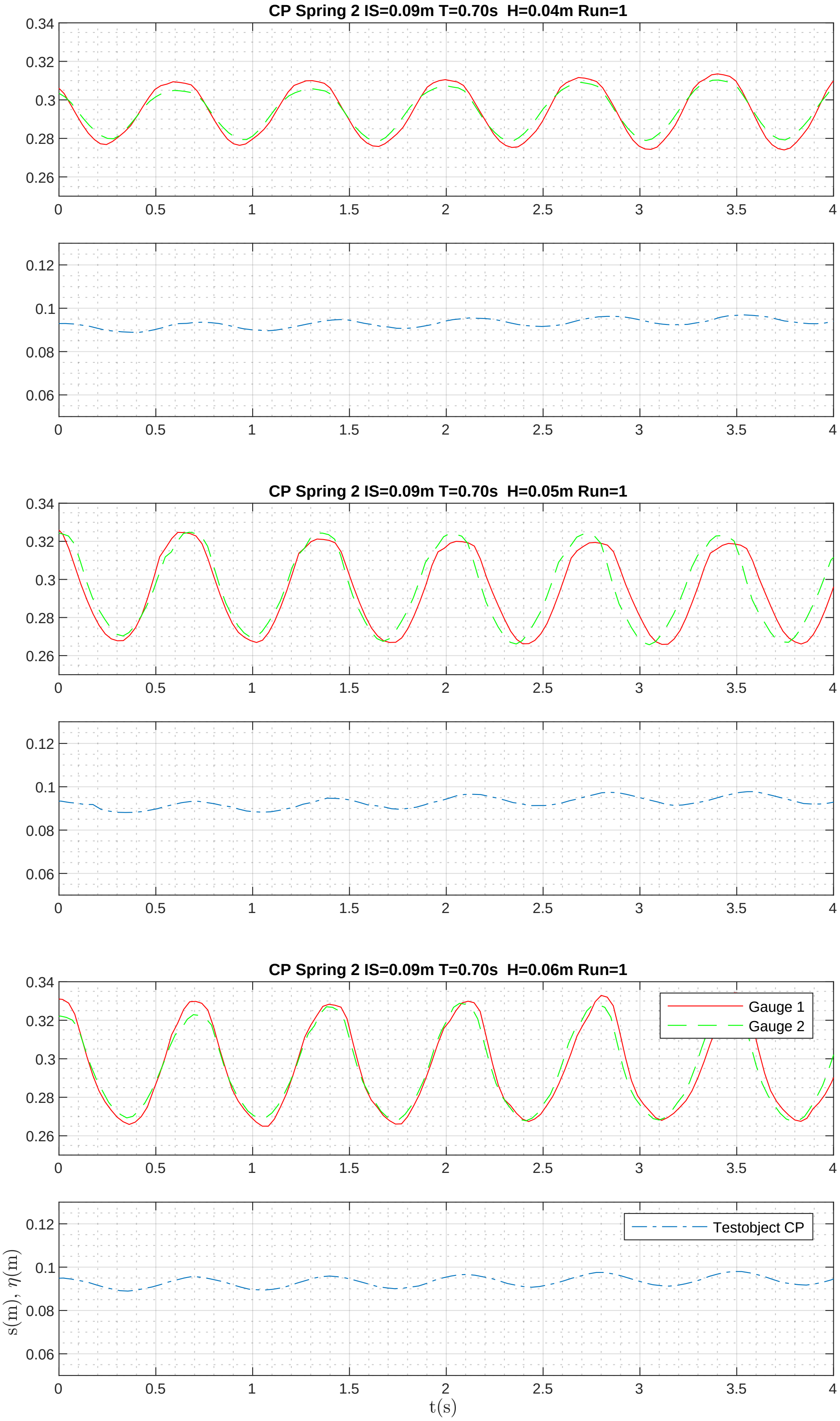

CP Spring 2 IS=0.15m T=1.75s H=0.05m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.18
0.16
0.14
0.12
0
0.5
1
1.5
2
2.5
3
3.5
4

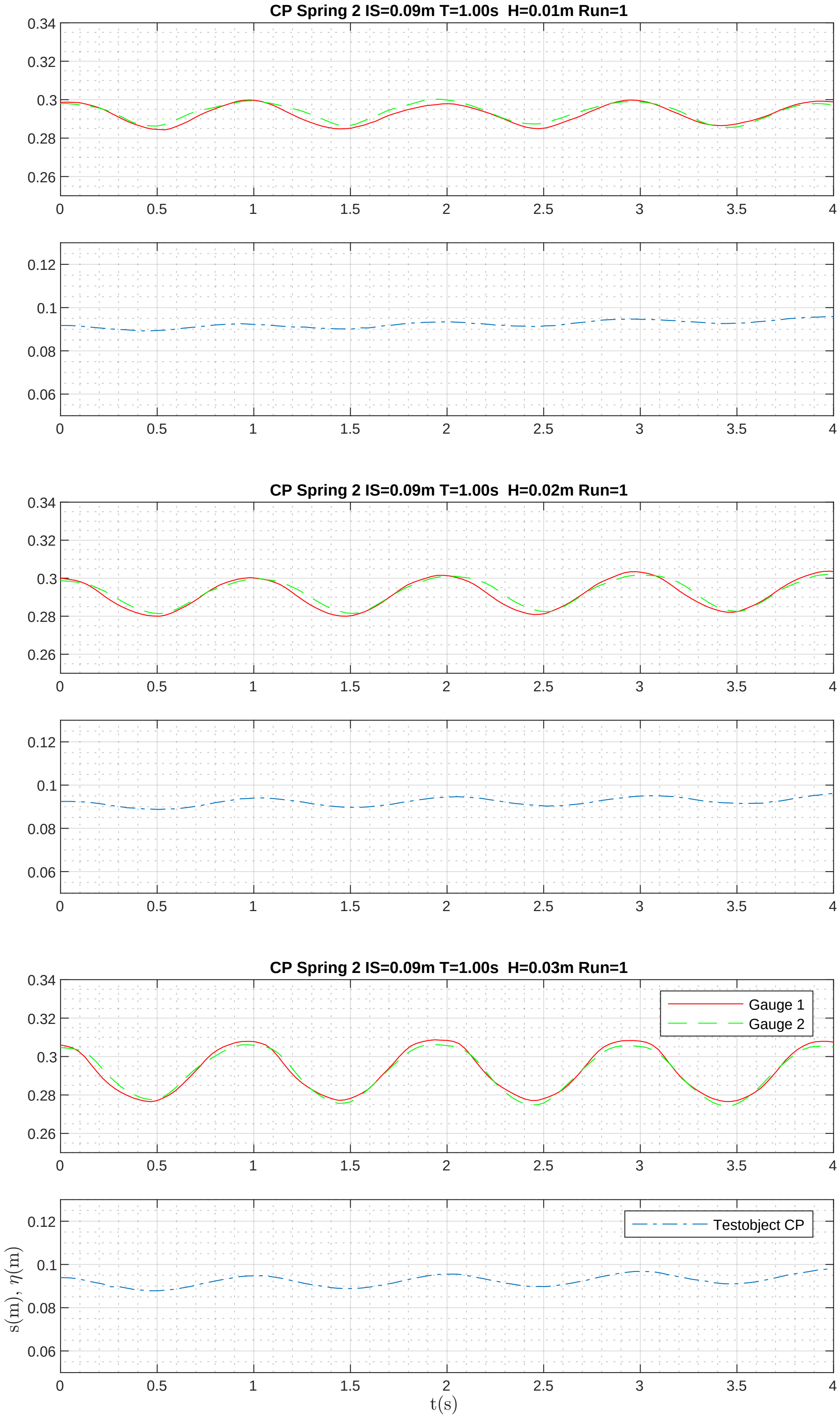

CP Spring 2 IS=0.15m T=1.75s H=0.06m Run=1
Gauge 1
Gauge 2
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
Testobject CP
0.18
0.16
0.14
0.12
s(m), η(m)
t(s)

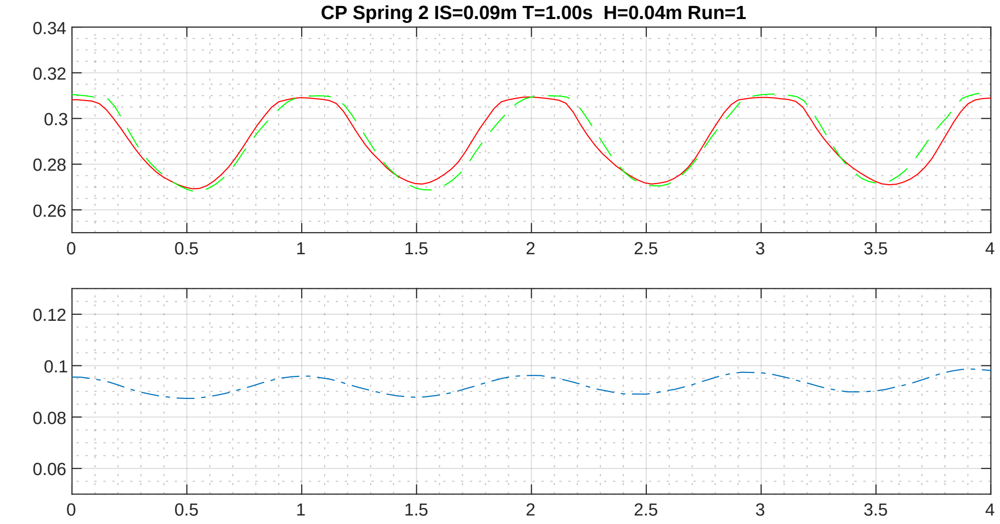

CP Spring 2 IS=0.09m T=0.70s H=0.01m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4

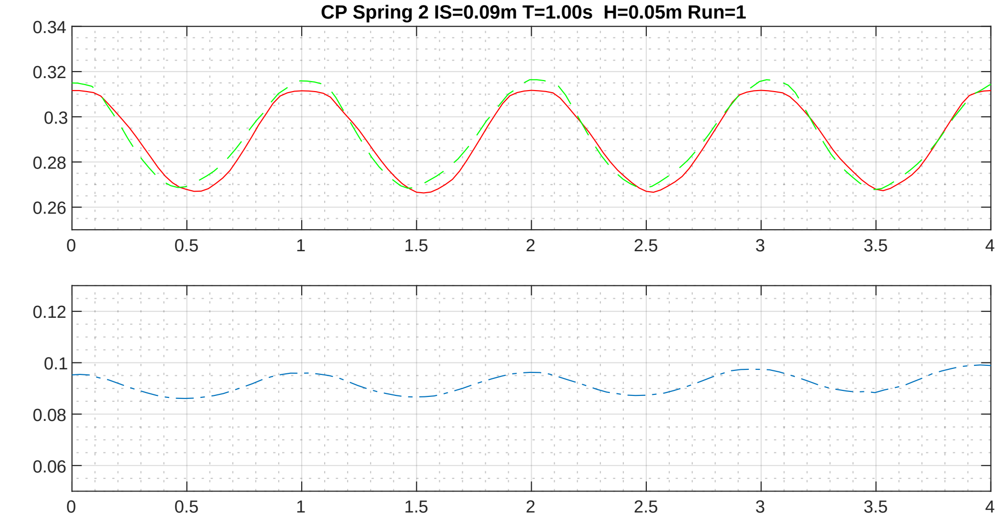

CP Spring 2 IS=0.09m T=0.70s H=0.02m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4

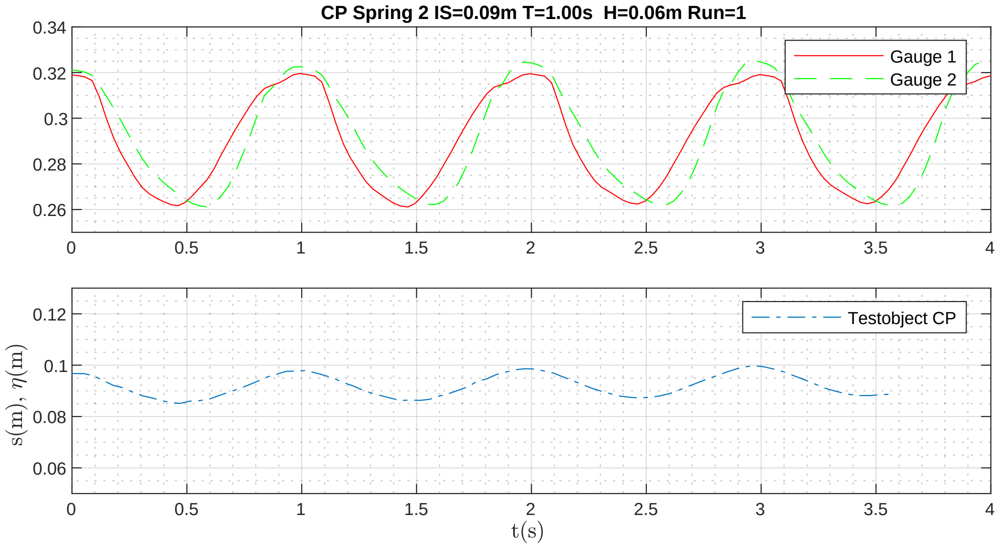

CP Spring 2 IS=0.09m T=0.70s H=0.03m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4

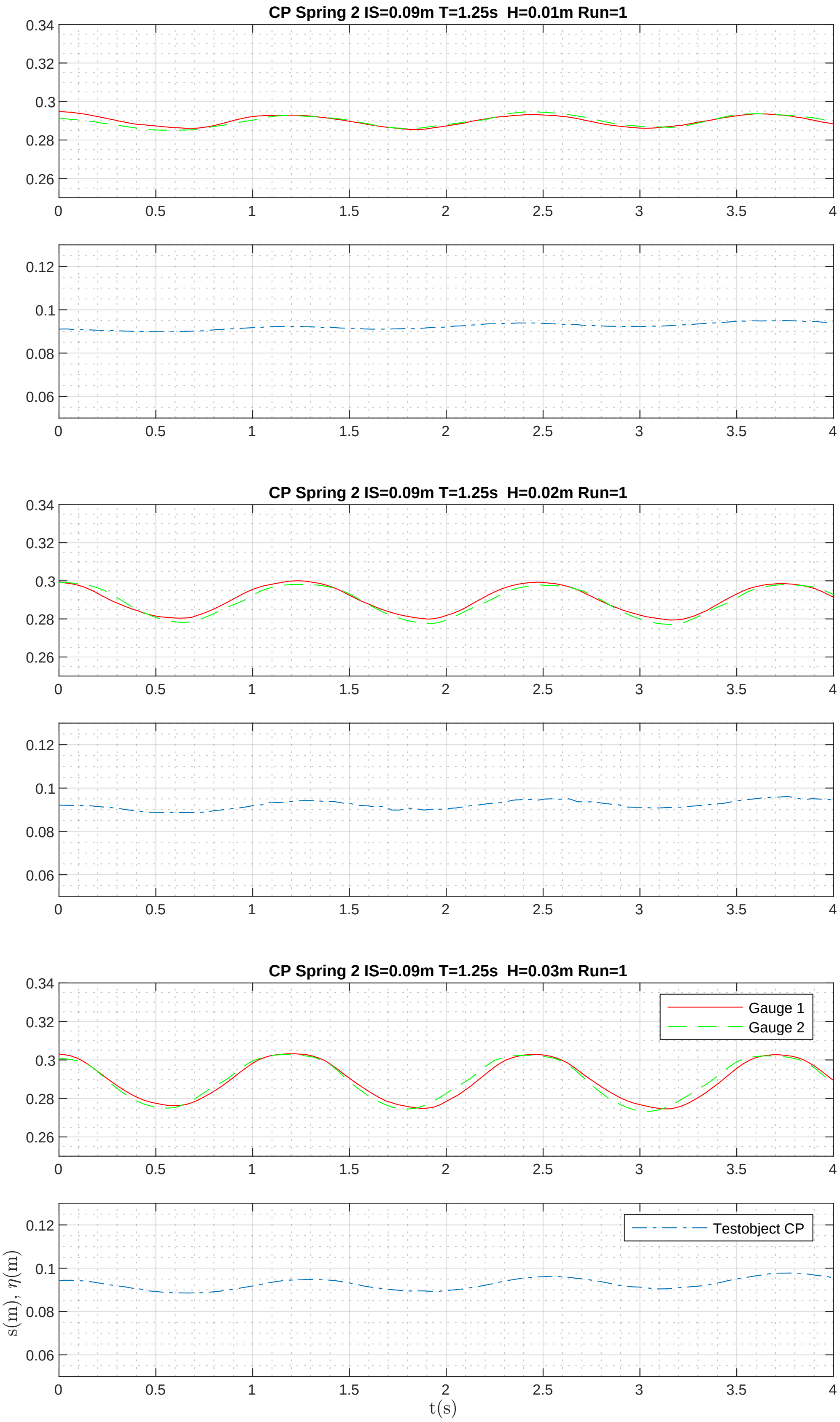

CP Spring 2 IS=0.09m T=0.70s H=0.04m Run=1
CP Spring 2 IS=0.09m T=0.70s H=0.05m Run=1
CP Spring 2 IS=0.09m T=0.70s H=0.06m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

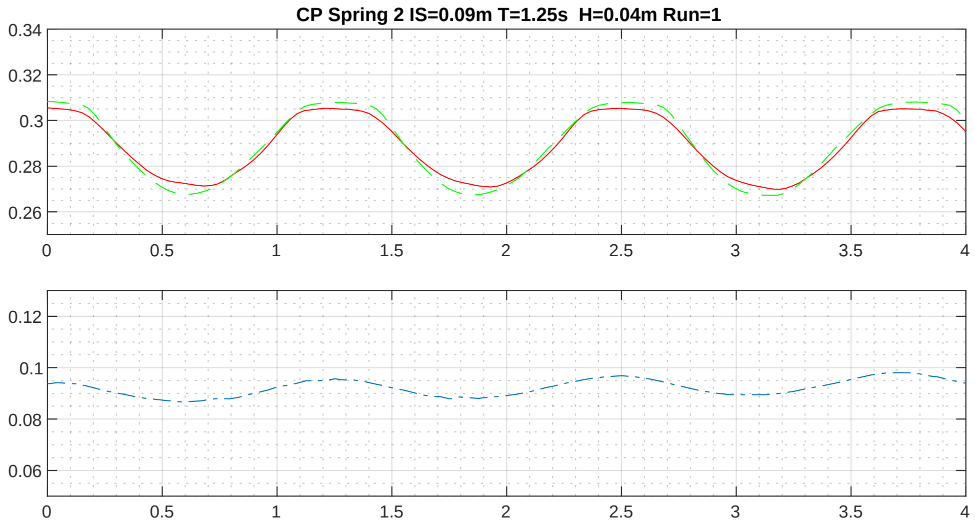

CP Spring 2 IS=0.09m T=1.00s H=0.01m Run=1
CP Spring 2 IS=0.09m T=1.00s H=0.02m Run=1
CP Spring 2 IS=0.09m T=1.00s H=0.03m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

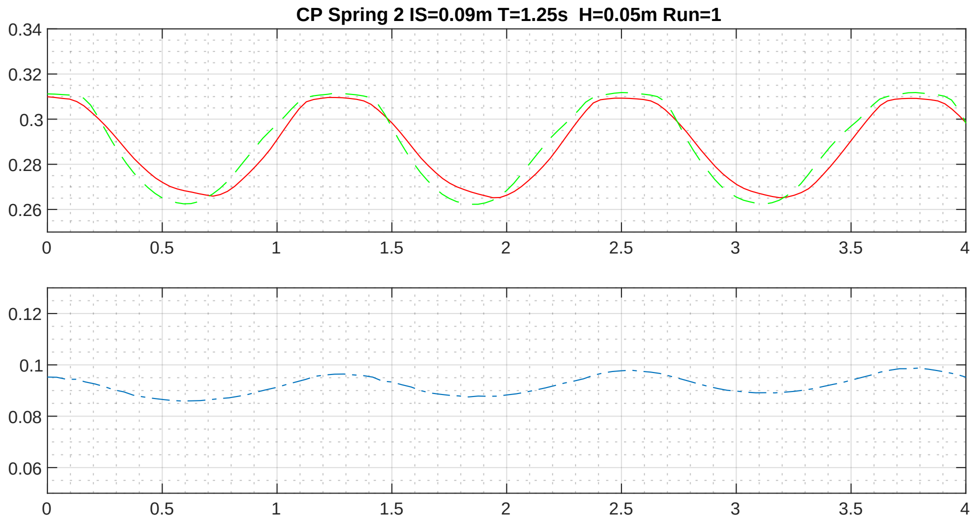

CP Spring 2 IS=0.09m T=1.00s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4

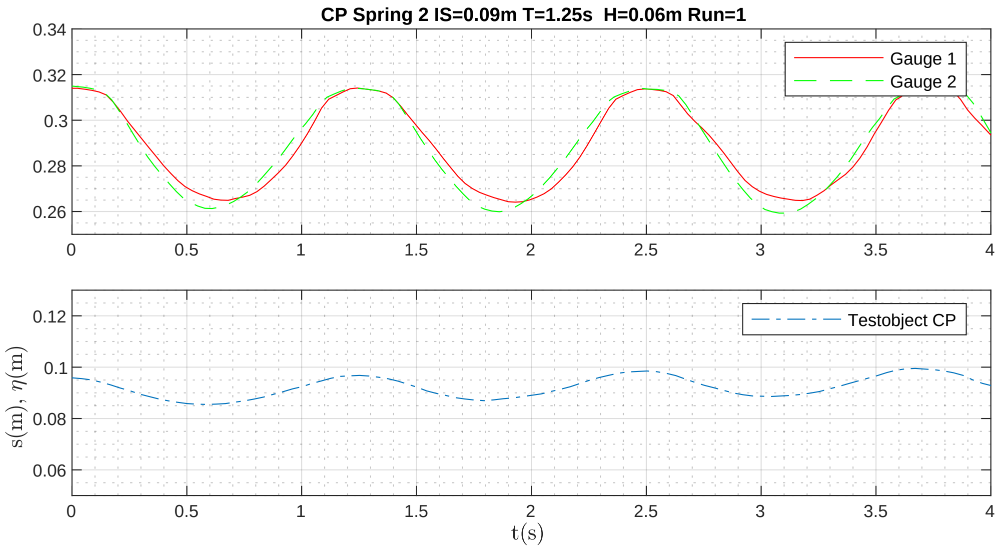

CP Spring 2 IS=0.09m T=1.00s H=0.05m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4

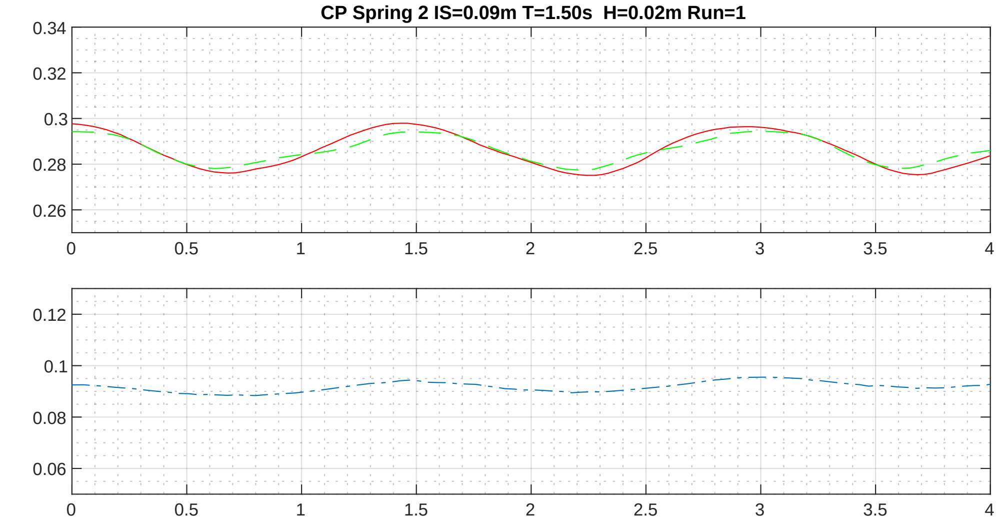

CP Spring 2 IS=0.09m T=1.00s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4


CP Spring 2 IS=0.09m T=1.25s H=0.01m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4
CP Spring 2 IS=0.09m T=1.25s H=0.02m Run=1
CP Spring 2 IS=0.09m T=1.25s H=0.03m Run=1
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

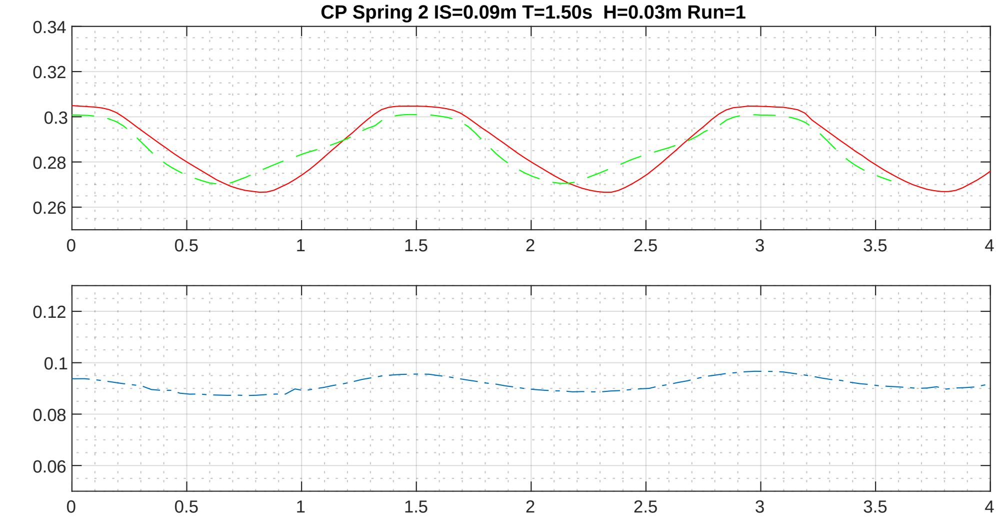

CP Spring 2 IS=0.09m T=1.25s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4


CP Spring 2 IS=0.09m T=1.25s H=0.05m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4

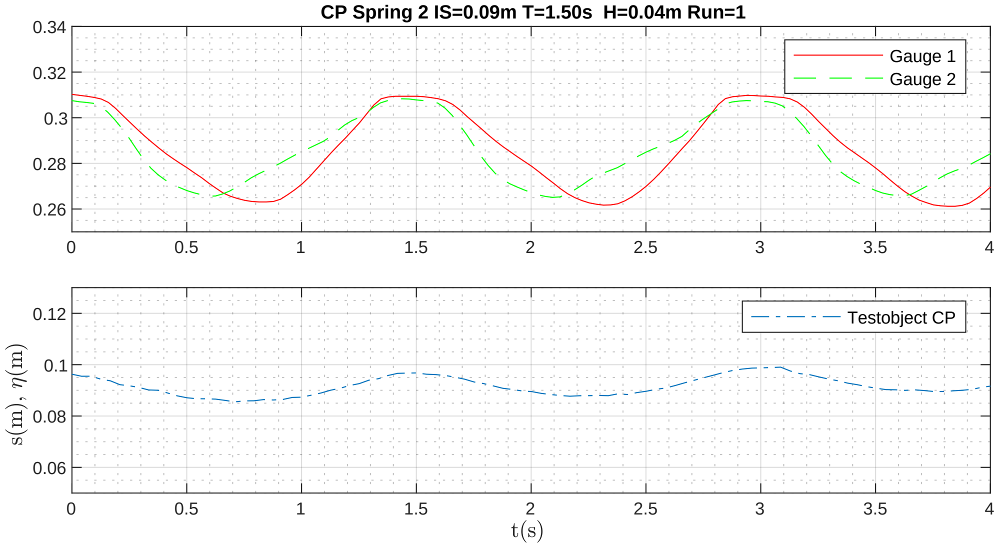

CP Spring 2 IS=0.09m T=1.25s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject CP
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
s(m), η(m)
0
0.5
1
1.5
2
2.5
3
3.5
4
t(s)



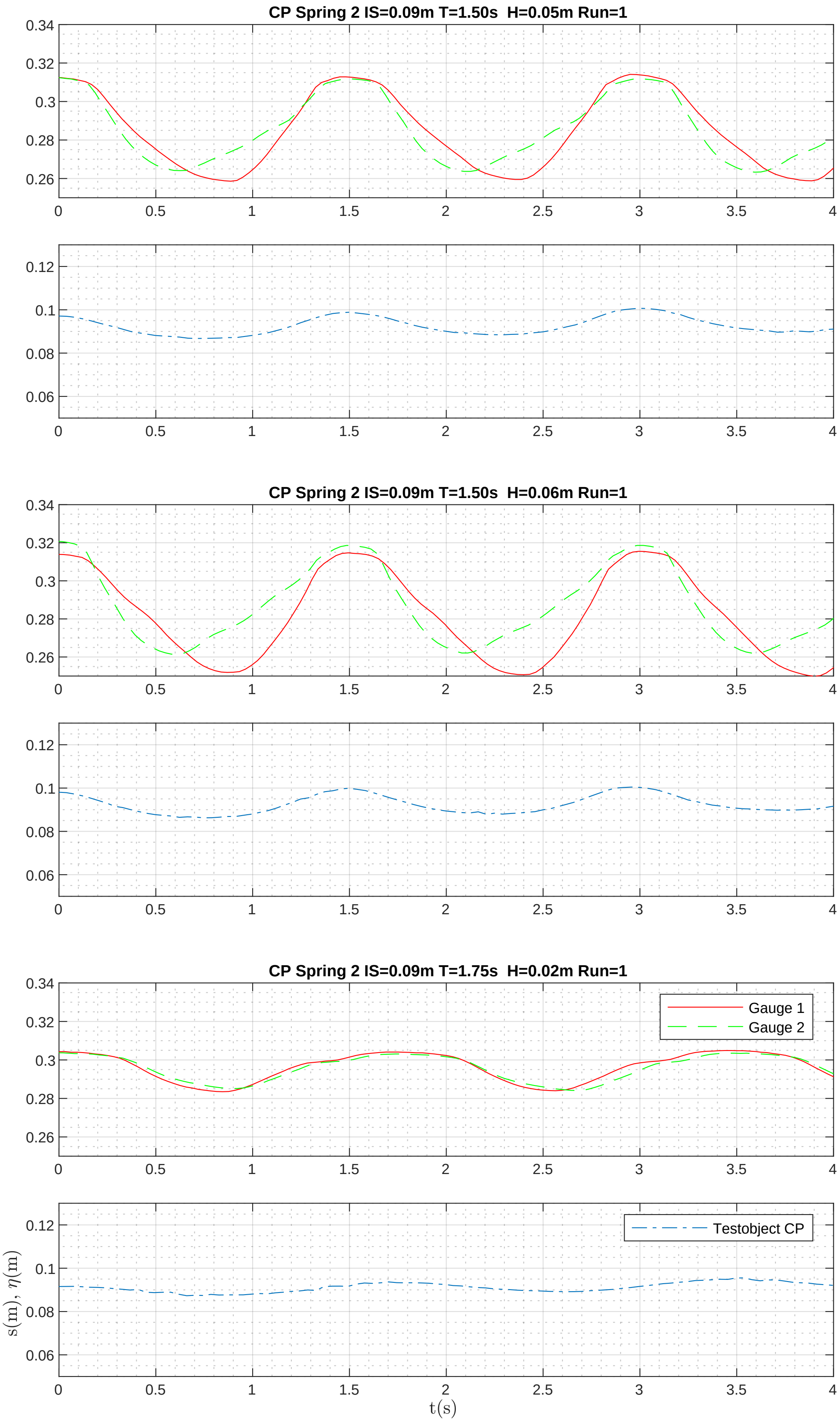
CP Spring 2 IS=0.09m T=1.50s H=0.02m Run=1

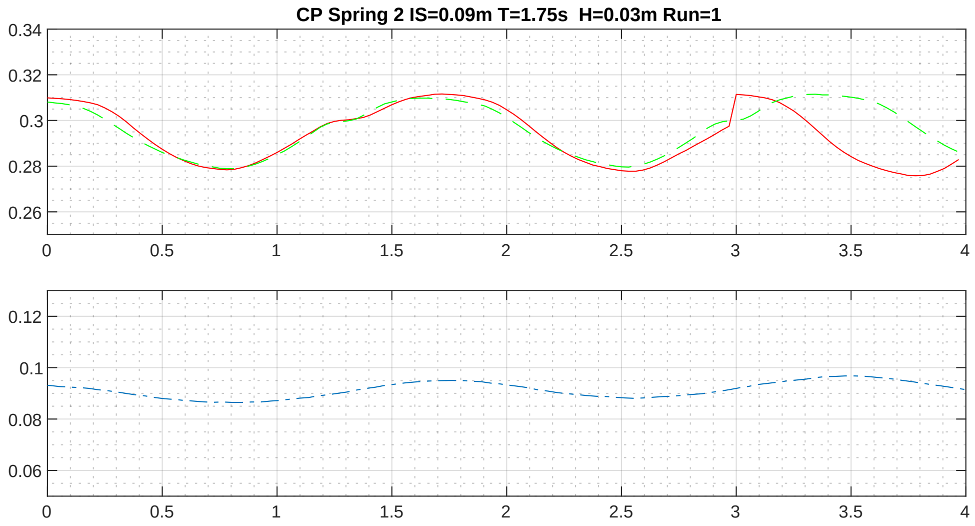


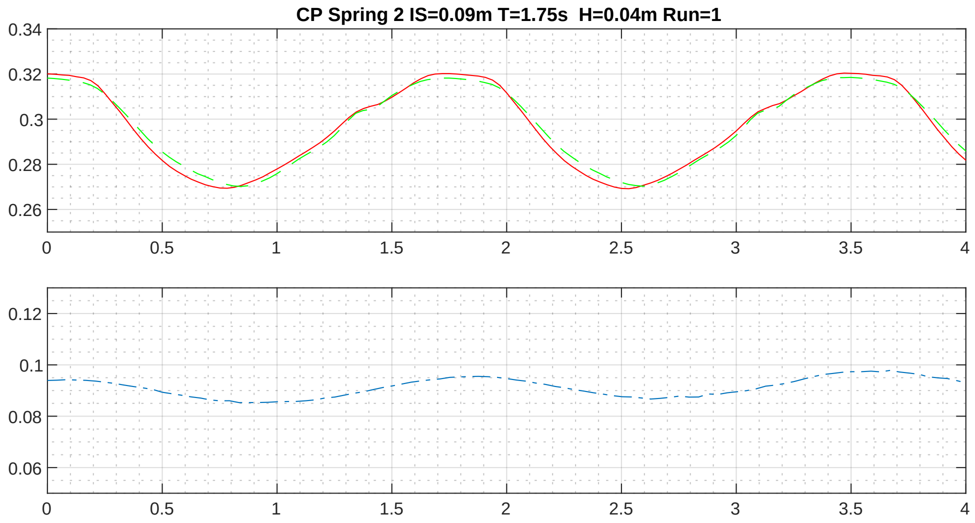
CP Spring 2 IS=0.09m T=1.50s H=0.03m Run=1

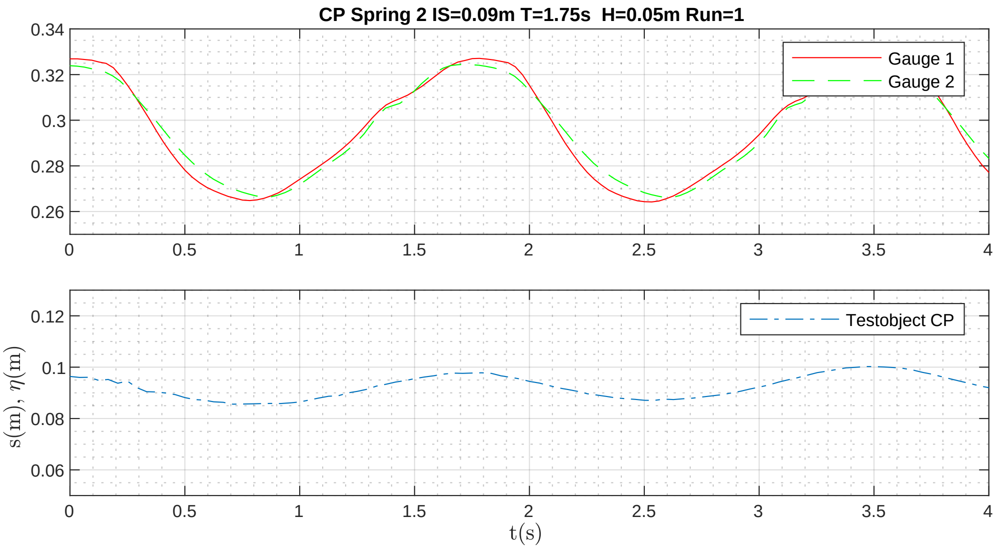


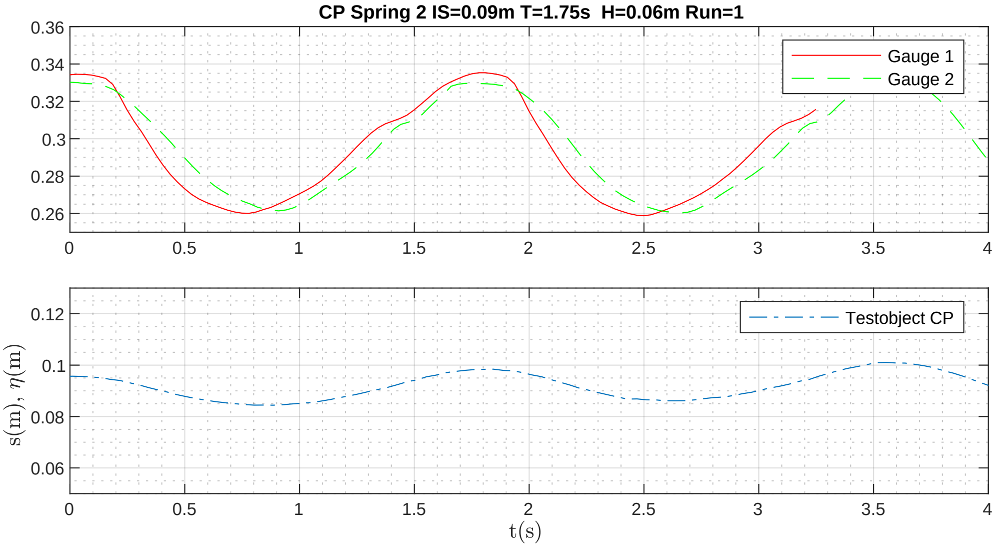
CP Spring 2 IS=0.09m T=1.50s H=0.04m Run=1


CP Spring 2 IS=0.09m T=1.50s H=0.05m Run=1
CP Spring 2 IS=0.09m T=1.50s H=0.06m Run=1
CP Spring 2 IS=0.09m T=1.75s H=0.02m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4
Gauge 1
Gauge 2
Testobject CP
s(m), η(m)
t(s)

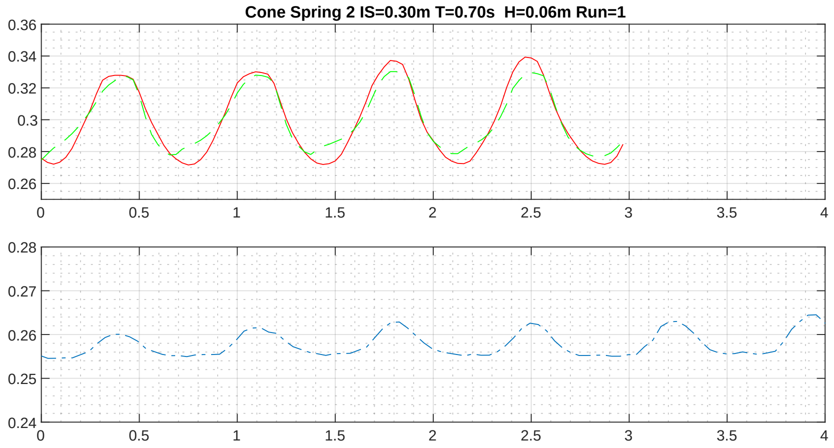

CP Spring 2 IS=0.09m T=1.75s H=0.03m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4

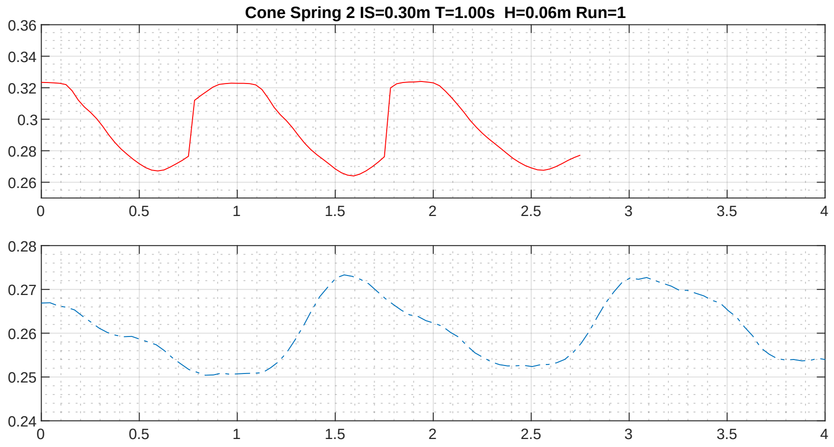

CP Spring 2 IS=0.09m T=1.75s H=0.04m Run=1
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4

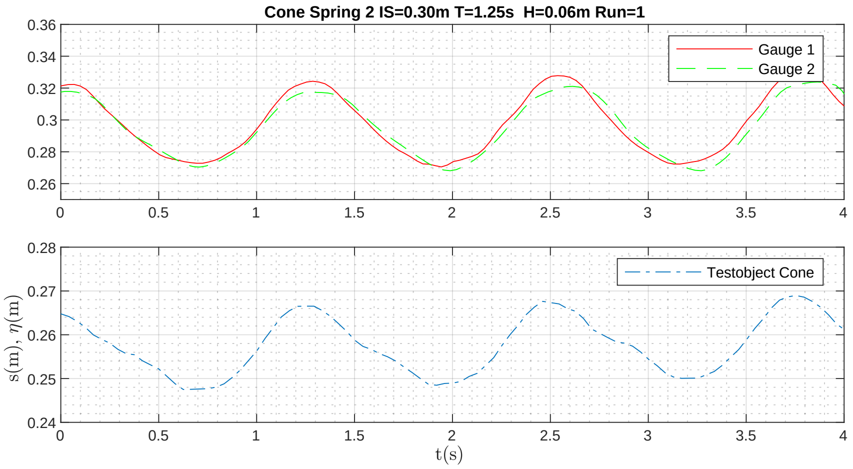

CP Spring 2 IS=0.09m T=1.75s H=0.05m Run=1
Gauge 1
Gauge 2
Testobject CP
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
s(m), η(m)
0
0.5
1
1.5
2
2.5
3
3.5
4
t(s)

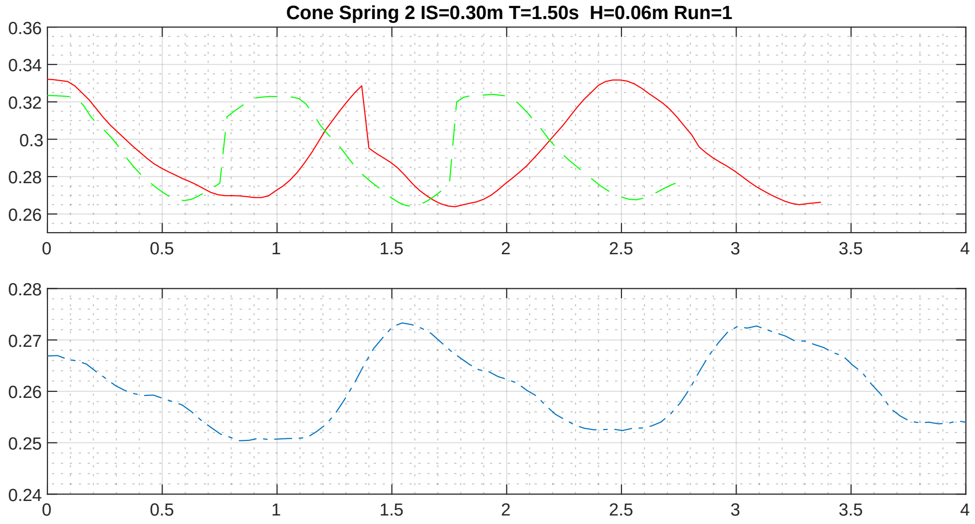

CP Spring 2 IS=0.09m T=1.75s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject CP
0.36
0.34
0.32
0.3
0.28
0.26
0.12
0.1
0.08
0.06
0
0.5
1
1.5
2
2.5
3
3.5
4
s(m), η(m)
t(s)

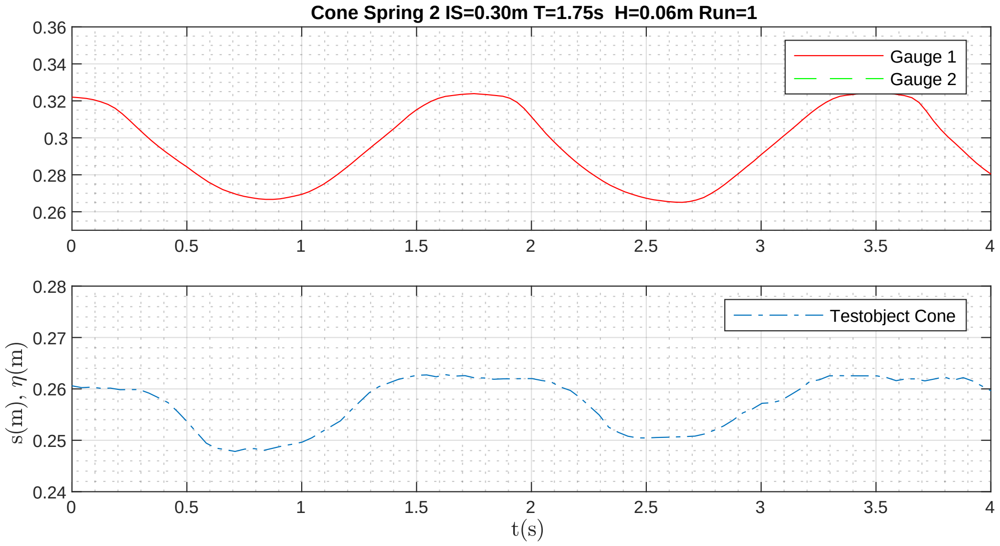
H. Plots Oscillation Height Testobject Cone

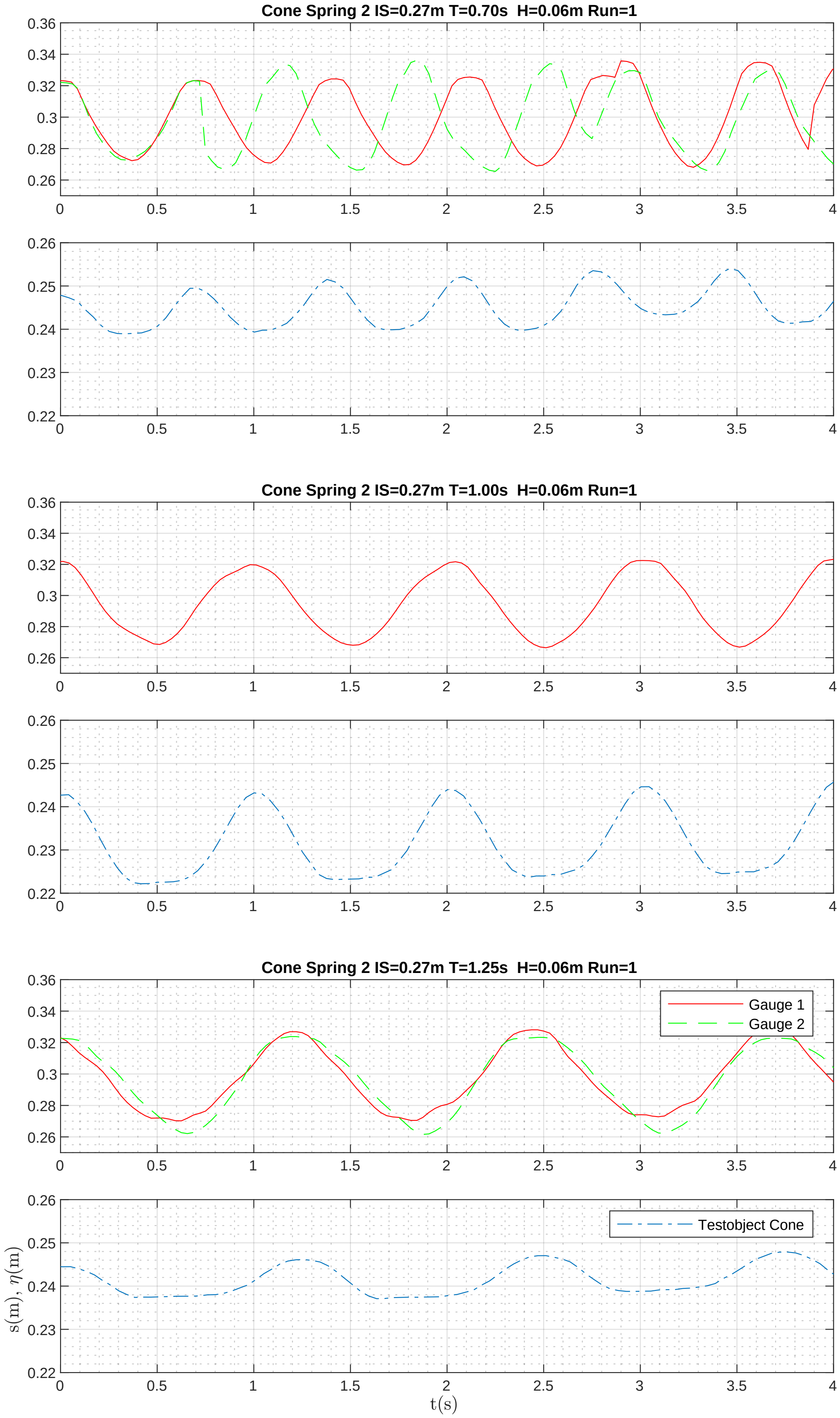

Cone Spring 2 IS=0.30m T=0.70s H=0.06m Run=1

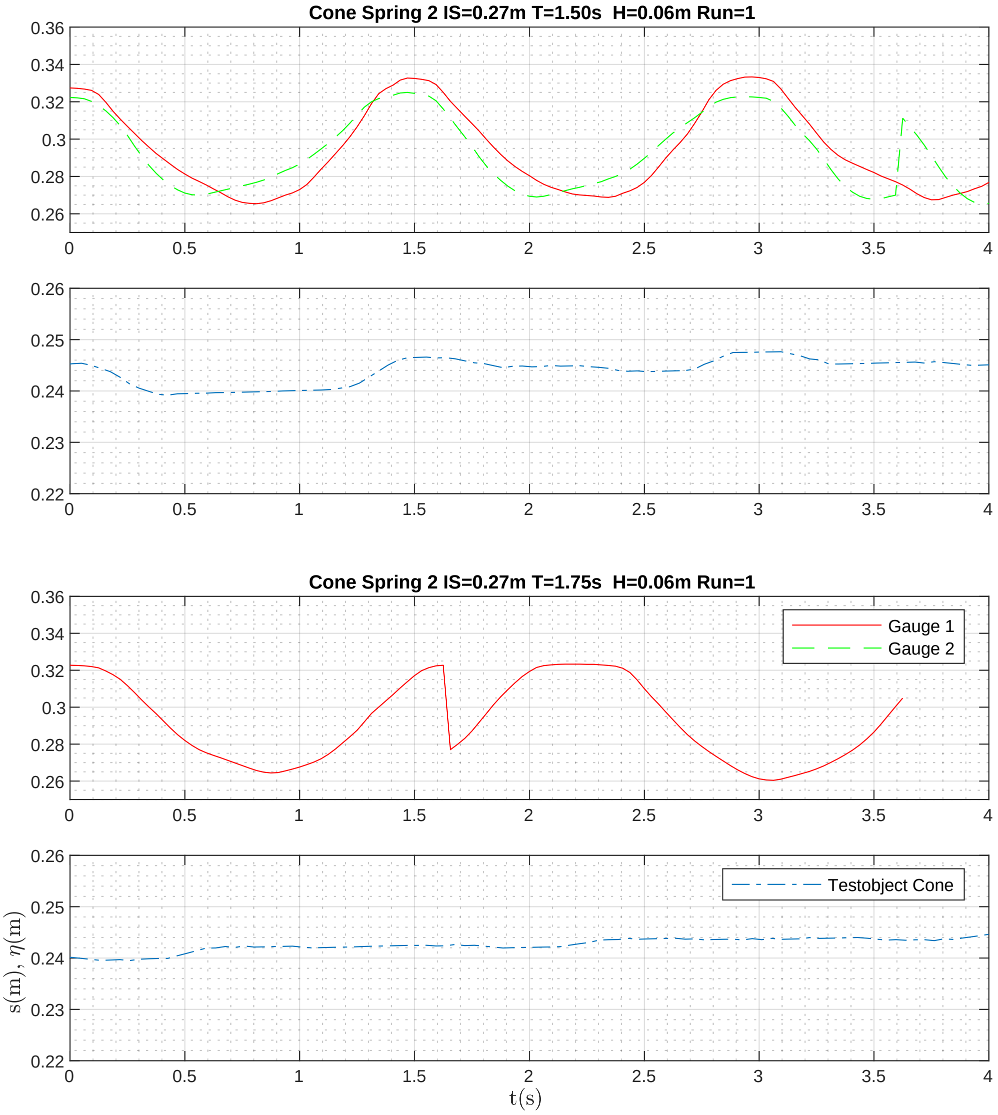

Cone Spring 2 IS=0.30m T=1.00s H=0.06m Run=1


Cone Spring 2 IS=0.30m T=1.25s H=0.06m Run=1


Cone Spring 2 IS=0.30m T=1.50s H=0.06m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.28
0.27
0.26
0.25
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4


Cone Spring 2 IS=0.30m T=1.75s H=0.06m Run=1
Gauge 1
Gauge 2
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
Testobject Cone
s(m), η(m)
0.28
0.27
0.26
0.25
0.24
0
0.5
1
1.5
2
2.5
3
3.5
4
t(s)


Cone Spring 2 IS=0.27m T=0.70s H=0.06m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0
0.5
1
1.5
2
2.5
3
3.5
4
0.26
0.25
0.24
0.23
0.22
Cone Spring 2 IS=0.27m T=1.00s H=0.06m Run=1
Cone Spring 2 IS=0.27m T=1.25s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject Cone
s(m), η(m)
t(s)


Cone Spring 2 IS=0.27m T=1.50s H=0.06m Run=1
0.36
0.34
0.32
0.3
0.28
0.26
0.25
0.24
0.23
0.22
0
0.5
1
1.5
2
2.5
3
3.5
4
Cone Spring 2 IS=0.27m T=1.75s H=0.06m Run=1
Gauge 1
Gauge 2
Testobject Cone
s(m), η(m)
t(s)