Technion – Israel Institute of Technology

Faculty of Mechanical Engineering



Project – Part 1

Vibration Theory

034011

June 20, 2021

# Abstract

A wind turbine with three blades was modeled as a point mass atop a cantilever beam.   
To compute the maximal stress in the turbine’s mast as a function of wind speed, and the system’s modal properties, the force exerted on the turbine was modeled as a sum of two harmonic forces where one has a frequency three times as the other in correlation to the number of blades on the turbine.  
In addition, two types of systems were checked:   
a linear one, where only the mast, transmission, generator and blades were modeled, and a none-linear one, where tracked springs were added, and the damping force was defined more broadly.  
Stress was calculated using Euler-Bernoulli static beam and the deflection was solved by ODE45 or transfer function, depending on the system at hand.  
We have found that the maximal stress in the linear system crosses the allowed stress of 108Mpa around two different wind speeds, while the maximal stress in the none-linear system never amounts to much more than half the allowed stress.

Contents

[1 Abstract 1](#_Toc531455970)

[2 Modeling a Linear System 3](#_Toc531455971)

[2.1 Calculate equivalent stiffness of the beam 3](#_Toc531455972)

[2.2 Write the equation of motion of the point mass using power terminology 3](#_Toc531455973)

[2.3 Calculate natural frequency, damping ratio, damped frequency and frequency of maximal amplitude 4](#_Toc531455974)

[2.4 Analytically calculate the system’s response to an impulse, given that there is no damping 4](#_Toc531455975)

[2.5 Analytical calculation of the complex frequency response with damping: 7](#_Toc531455976)

[2.6 Maximal stress in beam as a function of wind speed 9](#_Toc531455977)

[3 Adding non-linear elements to the system 10](#_Toc531455978)

[3.1 Write the equation of motion of the point mass using power terminology 11](#_Toc531455979)

[3.2 Find the equilibrium points of the system and classify them 12](#_Toc531455980)

[3.3 Find the natural frequency, damping ratio and unconstrained oscillation frequency of the point mass around the stable equilibriums 12](#_Toc531455981)

[3.4 Repeat 1.6 for the none-linear system solving with ODE45 13](#_Toc531455982)

[4 Conclusion 15](#_Toc531455983)

[5 Appendix 16](#_Toc531455984)

[5.1 Bibliography 16](#_Toc531455985)

[5.2 Matlab code 16](#_Toc531455986)

# Modeling a Linear System

We modeled the turbine as a point mass, representing the transmission, generator and blades, and a massless cantilever beam as the mast.

Where:



The movement of the turbine’s blades causes a damping force to be exerted on them:



|  |  |
| --- | --- |
| Figure 1 Turbine description   1. Turbine mast 2. Transmission and generator 3. Turbine blades | Figure 2 Turbine Model |

## Calculate equivalent stiffness of the beam

Calculating the beams stiffness was done using Euler-Bernoulli static beam theory (פרופ' י. ליפשיץ)

Where the following is assumed:

1. Planes perpendicular to the neutral axis before the load was applied remain so after
2. The deflections are small regarding the length of the beam
3. The beam deflects only in ****



## Write the equation of motion of the point mass using power terminology

The energy loss of the system is attributed only to the damping force, the power loss of the system can be written as such:



The terms T, V relate to kinetic and potential energy accordingly:



Plugging in the energy terms into the power equations yields the equation of motion:



We are not interested in the trivial solution , and we will analyze the none-trivial one:



## Calculate natural frequency, damping ratio, damped frequency and frequency of maximal amplitude

As the derived equation of motion is linear, natural frequency, damping ratio and other such properties can be calculated:

Table 1 Linear system - Modal properties



## Analytically calculate the system’s response to an impulse, given that there is no damping



It is known that the response for a system of the form  to a force unit impulse  is:



A convolution integral was used to solve the system’s steady state response to the given impulse.





1. To determine the value of  which will result in maximal steady state displacement we have plotted a solution surface  for a minimum of 3 periods and projected it to the  plane. The circumference of the project surface is .

|  |  |
| --- | --- |
|  | Figure 3 Linear system w/o damping - Maximal Displacement in steady state as a function of force shock input |

it is evident that for  a maximal displacement is obtained, and it can be computed analytically with a limit:



Note: It was assumed that the maximal displacement during the shock is of no interest as .

1. Calculating the extreme beta limits of the solution yields:



1. The solution we have plotted for  predicts that for some  the deflection is time independently zero.

This can be explained through the amplitude of the sine-based function  that zeros out when    
A physical approach to the question at hand would determine that for  the impulse will have done net zero work on the system. In other words:



To prove this, we first calculated the displacement during the shock using a convolution integral and taken its derivative.



We then expressed the integrand of the previously defined power integral:

   
Plugging in  has enabled us to write the integrand as a sum of sines



For the specific integral to be net zero, all three sine functions of the integrand must complete an integer number of cycles during the shock.





Solving the three equations above for n yields the following:



And the solution:



Q.E.D

## Analytical calculation of the complex frequency response with damping:

we now consider the damped system obtained in ‎2.2 with a generic harmonic input force 



When computing the complex frequency response, a complex exponential representation of the input force and output displacement is possible and simplifies the computation (William T.Thomson, Marie Dillon Dahleh, p. 52)



We then can write the equation of motion as follows:







Defining the transfer function  using the relations from 1.3:



denoting  simplifies the equation



For an input  the output can then be written (פלמור, 1993):



Alternatively, the transfer function could be defined through the Laplace transform as the following  where 

We had chosen to show the frequency response of the system with a Bode diagram, created by the alternative approach. Decibels were computed with velocity of 20.



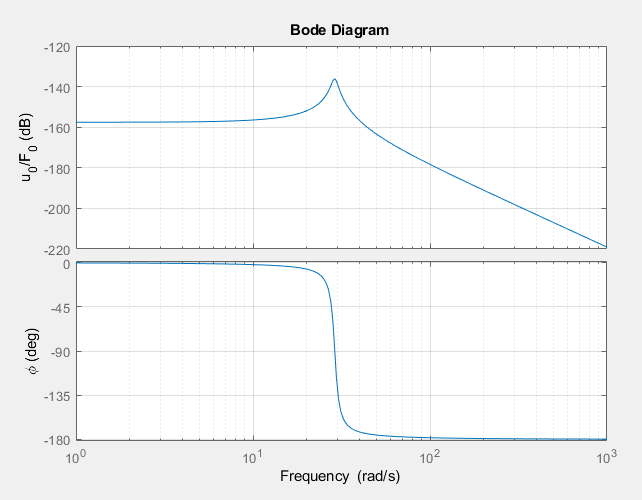


Figure 4 Linear system Bode diagram

## Maximal stress in beam as a function of wind speed

A more accurate model of the force acting on the turbine blades is given by the term



1. The displacement  was calculated by the superposition principle of linear systems and the transfer function we have derived in 1.5:



1. The maximal stress in the mast’s structure is obtained in the outermost shell of the mast’s base and was derived from static stiffness equations for the modeled cantilever beam (פרופ' י. ליפשיץ). Stress resulting from shear was neglected.



Under the current modeling conditions,  is only a function of  , and we can derive the maximal stress in the mast’s structure as a function of the wind speed.



Given that the allowed maximal stress is  , the following plot has been graphed for wind speeds that fulfill  :

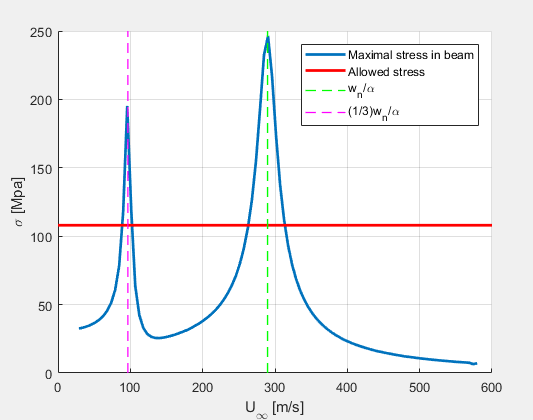


Figure 5 Linear system maximal stress as a function of wind speed

There are two sections where the maximal stress in the beam goes past the allowed stress. The first one, when  , and the second one when  .It should be noted that from our personal experience as enthusiast meteorologists, wind speeds around  are rare and near  are none-existed. Still, one should make sure to take the appropriate precautions to halt the turbine’s work when dangerous wind speeds are abounding.

# Adding non-linear elements to the system

In order to limit the movement of the turbine in the  direction, two springs of stiffness  (not to be confused with the stiffness of the beam ) were added to the system. One of each of the springs ends can move on track given by .



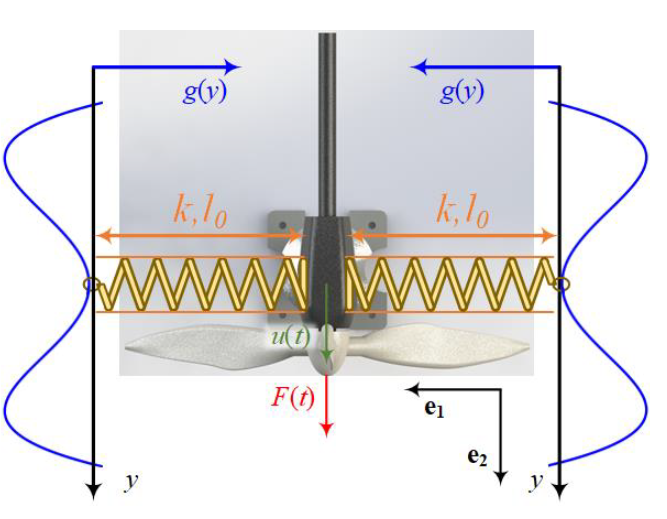


Figure 6 None linear system description

In addition, the damping force takes the following form:



## Write the equation of motion of the point mass using power terminology

We will apply the same methodology from 1.2





Plugging in the potential and kinetic energy terms into the power equation yields:



We will analyze the none-trivial solution:



## Find the equilibrium points of the system and classify them

We obtained the system’s equilibrium points by finding the roots of the displacement derivative of the potential energy . This was done in Matlab with cubic spline curve fitting.

We then classified the equilibrium points through the potential energy itself.

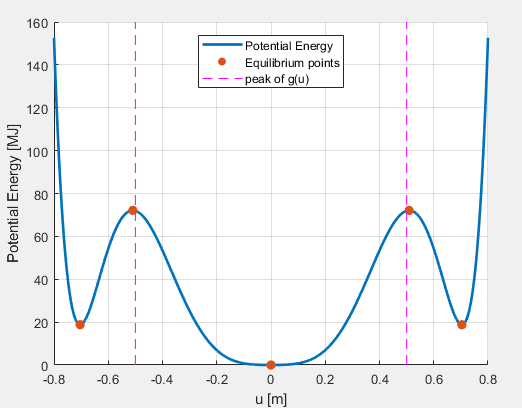


Figure 7 None-linear system potential energy and equilibrium points

The equilibrium points location and class are described in the table below:

Table 2 None linear system equilibrium points classification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | -0.7037 | -0.5094 | 0 | 0.5094 | 0.7037 |
|  | Stable | Unstable | Stable | Unstable | stable |



## Find the natural frequency, damping ratio and unconstrained oscillation frequency of the point mass around the stable equilibriums

For the natural frequency and other such properties to be defined, the system first must be linear. To linearize the system around the equilibrium points, we first define an implicit function from the equation of motion.



The linearized system will fulfill the following:



Where 

In our case:



1. Around the stable equilibrium :

The linearized system will take the exact same form of the system we solved in 1.3 and Its modal properties and can be seen in table 1.

1. Around the stable equilibriums :

Table 3 None linear system modal properties for distant stable equilibriums



## Repeat 1.6 for the none-linear system solving with ODE45

We were asked again to find the maximal stress in the turbine’s mast as a function of wind speed where the force exerted on the point mass is modeled as follows:



To confront the problem, we had first decided to solve the deflections around the stable equilibriums using ODE45 (Matlab function). The stress was then to be calculated from displacement through static stiffness equations – the same as in section 1.6.  
It was quickly evident that there is no real motivation for solving the maximal stress in the turbine’s mast as a function of wind speed for the distant stable equilibrium points as it will always surpass the allowed stress.



In addition, the problem for the linearized system around the stable equilibrium  was already solved in 1.6.  
For those reasons, we have decided to solve the none linear system for initial conditions:



We have plotted the solution graph for steady state conditions to allow for comparison for the parallel solution of the linear system.  
It should be noted that the transient solution offers no new revelations and for that reason it is not in presence.

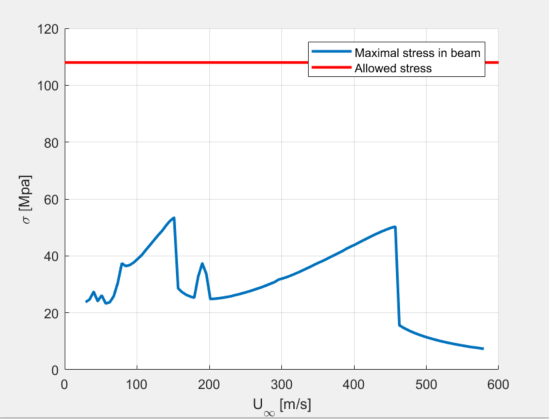


Figure 8 Non- linear system maximal stress as a function of wind speed

# Conclusion

The none-linear system stress analysis clearly shows that adding the tracked springs, and accounting for more of the damping force is good design, and smart modeling - the maximal stress in the beam never crosses its allowed threshold, as opposed to the state in the linear system.

# Appendix

## Bibliography

William T.Thomson, Marie Dillon Dahleh. (1998). *Theory of Vibration with Applications.* Prentice Hall.

פלמור, ז. (1993). *מערכות לינאריות.*

פרופ' י. ליפשיץ. (n.d.). *מכניקת מוצקים 2.* הפקולטה להנדסת מכונות טכניון.

## Matlab code

%% initalize

mili=1e-3; micro=1e-6;

E=210\*1e9; %Pa

L=65; %m

m=90\*1e3; %kg

r\_out=3; %m

r\_in=2.5; %m

c1=22\*1e4; %Ns/m, Fd=-c1\*du/dt

%% A1

I=pi\*(r\_out^4-r\_in^4)/(4); %m^4

k=(3\*E\*I)/(L^3); %N/m

%% A2 - no code needed.

%motion eq: mu''+c1u'+ku=0

%' represents time derivitive

%% A3

wn=sqrt(k/m); %rad/s

zeta=c1/(2\*sqrt(k\*m)); %unitless

wd=wn\*sqrt(1-zeta^2); %rad/s. free osciliation radial frequency

wr=wn\*sqrt(1-2\*zeta^2); %rad/s. maximal output/input 4 steady state radial frequency

%% A4

fn=wn/(2\*pi);

t=linspace(0,3/fn,100); %s, we want to show 3 machzorim

beta=linspace(0,2\*wn,100);

[T,Beta]=meshgrid(t,beta); %mesh matrix form coordinates

% u\_InShock=(1040\*Beta.\*sqrt(Beta)./(wn\*pi\*(Beta.^2-wn.^2))).\*(wn\*sin(Beta.\*T)-Beta.\*sin(wn.\*T));

u=(1040\*Beta.\*sqrt(Beta)./(wn\*pi\*(Beta.^2-wn.^2))).\*cos((pi\*wn)./(2\*Beta)).\*sin(wn.\*T-(pi\*wn)./(2.\*Beta));

%Plot u(beta,t)

srffig=figure;

srfAx=axes(srffig);

surf(srfAx,T,Beta,u)

xlabel(srfAx,'t [s]');

ylabel(srfAx,'\beta [rad/s]');

zlabel(srfAx,'u [m]');

%plot max(u(beta,t),beta) and wn

Maxu=max(u,[],2);

Srspfig=figure;

SrspAx=axes(Srspfig); grid(SrspAx,'on'); hold(SrspAx,'on');

plot(SrspAx,beta,Maxu,'linewidth',2)

plot(SrspAx,[wn,wn],SrspAx.YLim,'--g','linewidth',2)

xlabel(SrspAx, '\beta [rad/s]');

ylabel(SrspAx, 'Displacement [m]');

legend('max(u(t,\beta))','w\_n','location','best');

%% PA5

%Eq: mu''+c1u'+ku=F; F=F0cos(wt)

H=tf(1,[m,c1,k]);

BodeFig=figure;

BodeAx=axes(BodeFig);

bode(BodeAx,H);

grid(BodeAx,'on');

%% A6

A=3.5\*1e6; %N

B=2.5\*1e6; %N;

alpha=0.1; %rad/m

phi=pi/3; %rad

aUinf=linspace(0.1\*wn,2\*wn,100); %rad/s

Uinf=aUinf/alpha; %m/s

t=linspace(0,2\*pi/(0.1\*wn),100); %build time vector according to slowlest frequency

[T,AUINF]=meshgrid(t,aUinf);

r=aUinf/wn; %Normalize

H=@(r) 1./(k\*(1-r.^2+2\*1i\*zeta\*r)); %normalized transfer function form

M1=(abs(H(r)))'; Psy1=(angle(H(r)))'; %hegber and phasa of alpha\*Uinf

M3=(abs(H(3\*r)))'; Psy3=(angle(H(3\*r)))'; %hegber and phasa of 3\*alpha\*Uinf

U1=A\*M1.\*cos(AUINF.\*T+Psy1); %m %AUINF changed across rows, and so Psy1 is column vector

U2=B\*M3.\*cos(AUINF.\*T+Psy3+phi); %m

U=U1+U2;

MaxU=max(U,[],2);

MaxSigma=k\*L\*r\_out\*MaxU/I; %Pa

Sigma\_allowed=108; %Mpa

%Plot Stress vs windspeed

sigmafig=figure;

sigmaAx=axes(sigmafig); hold(sigmaAx,'on'); grid(sigmaAx,'on');

plot(sigmaAx,Uinf,MaxSigma\*micro,'linewidth',2);

plot(sigmaAx,sigmaAx.XLim,[Sigma\_allowed,Sigma\_allowed],'r','linewidth',2);

plot(sigmaAx,[wn/alpha,wn/alpha],sigmaAx.YLim,'--g','linewidth',1)

plot(sigmaAx,[wn/(3\*alpha),wn/(3\*alpha)],sigmaAx.YLim,'--m','linewidth',1)

xlabel(sigmaAx,'U\_\infty [m/s]');

ylabel(sigmaAx,'\sigma [Mpa]');

legend(sigmaAx,'Maximal stress in beam','Allowed stress','w\_n/\alpha','(1/3)w\_n/\alpha');

%% B1

%nothing to compute. motion equation

%% B2

%find potenial(U) and its derivative with respect to time (dU)

%find equilibrium points from dU=0, classify them from U

%initalize new parameters

ks=4\*1e13; %N/m

c5=1e3; %N\*s^5/m^5

g=@(y) (1/25)\*(0.5\*y.^4-0.25\*y.^2);

dg=@(y) (1/25)\*(2\*y.^3-0.5\*y);

%build u,U,dU

u=linspace(-0.8,0.8,1000); %m

U=@(u) 0.5\*k\*u.^2+ks\*(g(u)).^2; %J

dU=@(u) k\*u+2\*ks.\*g(u).\*dg(u);

%find equilibriums

ForceSpline=csapi(u,dU(u)); %Cubic Spline interpolation - builds piece wise 3d degree polys.

%can be plotted with fnplt(spline)

ZeroInt=fnzeros(ForceSpline,[min(u),max(u)]); %find zeros of the spline in a given interval

%Z is a 2xm numeric vector array: each column 1:m describes the interval over which the spline is zero.

%if the spline reaches zero, but does not cross the zero line the root may

%not reveal itself in ZeroInt.

ZeroPoint=abs(ZeroInt(2,:)-ZeroInt(1,:))<eps; %bool array. 1: only the zero intervals which are of 0 length == a root.

Roots=ZeroInt(1,ZeroPoint); %take the first row (arbitrary) from ZeroInt which complies with ZeroPoint

%do some plotting

potentfig=figure;

potentAx=axes(potentfig); hold(potentAx,'on'); grid(potentAx,'on');

plot(potentAx,u,U(u)\*micro,'linewidth',2);

scatter(potentAx,Roots,U(Roots)\*micro,50,'filled');

plot([0.5,0.5],potentAx.YLim,'--m')

plot([-0.5,-0.5],potentAx.YLim,'--m')

xlabel(potentAx,'u [m]'); ylabel(potentAx,'Potential Energy [MJ]');

legend(potentAx,'Potential Energy','Equilibrium points','peak of g(u)',...

'location','north');

%% B3

%equation of motion around equlibrium:

%m\*ddu+Ceq\*du+Keq\*u=F

Keq=@(u\_eq) k+2/(25^2)\*ks\*(56\*u\_eq^6-30\*u\_eq^4+3\*u\_eq^2)/8;

Ceq=@(u\_eq) c1; %du is 0 in equilbrium

%for u\_eq1=0 the results are the same as in A2

%for u\_eq2=0.7037

wn\_eq2=sqrt(Keq(0.7037)/m); %rad/s

zeta\_eq2=Ceq(0.7037)/(2\*sqrt(Keq(0.7037)\*m)); %unitless

wd\_eq2=wn\_eq2\*sqrt(1-zeta\_eq2^2); %rad/s. free osciliation radial frequency

wr\_eq2=wn\_eq2\*sqrt(1-2\*zeta\_eq2^2); %rad/s. maximal output/input 4 steady state radial frequency

%% B4

%should I use ODE45 to solve linear equation, or non linear equation?

A=3.5\*1e6; %N

B=2.5\*1e6; %N;

alpha=0.1; %rad/m

phi=pi/3; %rad

%first stable equalbrium u0=0. second stable equilibrium is it self in

%"danger zone" stress wise. no point in checking

u0=[0,0];

aUinf=linspace(0.1\*wn,2\*wn,100); %rad/s

T=2\*pi/(0.1\*wn); %slowest frquency

tspan=linspace(0,100\*T,1000); %tons of cycles

[~,du]=arrayfun(@(aUinf) ode45(@(t,u) odefunB6(t,u,m,c1,c5,k,ks,g,dg,A,B,phi,aUinf),tspan,u0),...

aUinf,'un',0); %du = [u,dudt] output

du=cellfun(@(x) x(:,1),du,'un',0); %obtain all odd columns

u=cell2mat(du)-u0(1);

u\_ss=u(500:1000,:);

MaxU=(max(abs(u\_ss),[],1))';

Uinf=aUinf/alpha; %m/s

MaxSigma=k\*L\*r\_out\*MaxU/I; %Pa

Sigma\_allowed=108; %Mpa

%Plot Stress vs windspeed

sigmafig=figure;

sigmaAx=axes(sigmafig); hold(sigmaAx,'on'); grid(sigmaAx,'on');

plot(sigmaAx,Uinf,MaxSigma\*micro,'linewidth',2);

plot(sigmaAx,sigmaAx.XLim,[Sigma\_allowed,Sigma\_allowed],'r','linewidth',2);

xlabel(sigmaAx,'U\_\infty [m/s]');

ylabel(sigmaAx,'\sigma [Mpa]');

legend(sigmaAx,'Maximal stress in beam','Allowed stress');

function dudt=odefunB6(t,u,m,c1,c5,k,ks,g,dg,A,B,phi,aUinf)

%equation from linearization around stable equilibrium

%syntax:

%u(1)=u; u(2)=dudt

%dudt(1)=du/dt; dudt(2)=d2u/dt2

dudt=zeros(2,1);

dudt(1)=u(2);

dudt(2)=(1/m)\*(-c1\*u(2)-c5\*(u(2))^5-k\*u(1)-2\*ks\*g(u(1))\*dg(u(1))+A\*cos(aUinf\*t)+B\*cos(3\*aUinf\*t+phi));

end