

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Deák Gergely

**Szervómotorok akusztikai vizsgálata**

Önálló laboratórium I  
Msc

külső Konzulens

Kimpián Tibor,   
Thyssenkrupp Components Technology Hungary Kft.

Belső Konzulens

Orosz György

BUDAPEST, 2021

Feladatkiírás:

* A Thyssenkrupp kormányrendszerek fejlesztésével foglalkozik, ahol a kormányzás rásegítése állandómágneses szinkronmotorral történik. Ezeknek a motoroknak a zaja bejutva az utastérbe zavaró, így ennek a zajnak a csökkentése érdekében szükséges a motor viselkedésének mélyebb megértése. A motor "zajának" jelentős hányada a légrésben keletkező elektromágneses erők hatására jön létre és a motor mechanikai felépítése által módosítva jut be a korányrendszerbe, majd onnan az autó karosszériáján keresztül az utastérbe. Ebben a témában a motor mechanikai viselkedésének megértése a cél egy egyszerűsített koncentrált paraméteres modell segítségével, ami a motor állórészén és házán megjelenő hullámterjedés viselkedését írja le jól.

Tartalom

[1 Bevezetés 4](#_Toc74691479)

[2 Rezgéstani alapok 5](#_Toc74691480)

[2.1 Rezgéstani alapok: 5](#_Toc74691481)

[2.2 A rezgések eredete 5](#_Toc74691482)

[2.3 A rezgések terjedése 6](#_Toc74691483)

[2.4 A zaj csökkentése 6](#_Toc74691484)

[2.5 A lengő rendszer modellezése 6](#_Toc74691485)

[3 A tervezés részletes leírása, a választott megoldások indoklása 8](#_Toc74691486)

[3.1 A matematikai modell felállítása 8](#_Toc74691487)

[3.2 Rendszeregyenlet 9](#_Toc74691488)

[3.3 A GUI megtervezése 10](#_Toc74691489)

[4 A megtervezett műszaki alkotás értékelése, továbbfejlesztési lehetőségek 11](#_Toc74691490)

[5 Irodalomjegyzék 12](#_Toc74691491)

[6 Mellékletek (dokumentációk: kapcsolási rajz, huzalozási rajz, beültetési rajz, forrás file-ok, kész program, installálási és használati útmutatók stb.) 13](#_Toc74691492)

[6.1 Modusmátrixgenerátor 13](#_Toc74691493)

[6.2 Elmozdulásszámítás 14](#_Toc74691494)

# Bevezetés

Az önálló laboromat a ThyssenKrupp közreműködésével írtam, a kiírt feladat szervomotorok akusztikai vizsgálata volt.

Az elektromos autók terjedésével a zajkibocsátás csökken, így olyan zajok is előtérbe kerülnek, amiket eddig a belső égésű motorok hangja elnyomott, ide tartozik a kormányzásrásegítés zaja is.

Az egyre nagyobb teljesítményű autók és az ergonómiai szempontok megkövetelik a személyautókban a szervó rásegítés beépítését. A hidraulikus rásegítés helyett az alacsonyabb fogyasztás és egyszerűbb felépítés érdekében egyre elterjedtebb az elektromos szervó-motorok alkalmazása.

A motorokon végzett rezgés mérések spektrális elemzésével lehetőségünk van azonosítani az egyes zajkomponensek forrását, melyet fel tudunk használni a motorok későbbi tervezési lépéseinél.

Az előző dolgozatomban megvizsgáltam a rezgéstan alapjait, a rezgések terjedésének fizikáját és a modális elemzés alapjait. Ezeknek az ismereteknek a segítségével készítettem egy olyan programot, amivel a hullámok terjedését pontszerű tömegek között modellezhetjük. [6]

A továbbiakban ennek a programnak a felhasználásával készítek egy olyan optimalizáló algoritmust, amivel az ANSYS bemenő paramétereit lehet finomhangolni.

# Optimalizálás

A feladat elkészítéséhez szükségem volt arra, hogy megértsem az optimalizáló algoritmusok működését.

## Optimalizálás alapok:

## Négyzetes hiba

A legegyszerűbb optimalizálási módszer, ha két függvény értékeinek négyzetes különbségeit vesszük és összeadjuk a teljes értelmezési tartományban, így kapunk egy skalármennyiséget, amivel minősíteni tudjuk a két függvény kapcsolatát. Ennek a skalármennyiségnek a minimumát keressük a paramétertérben.

Paraméterek:

visszacsatolt = 1;

j = sqrt(-1);

m = 1;

k = 1;

c = 1;

Ms = 18;

force = 1;

force\_pos = 1;

omegakezdo = 0.1;

Nomega = 10000;

Kiertekeles = 1.5;

A hiba számításhoz használt function:

E\_real\_np =@(x) sum((real(FRF\_matrix(:,n))-real(elmozdulasszamitas\_optimum(m, x(1), x(2), force, force\_pos, Ms, omegakezdo, Nomega, Kiertekeles, visszacsatolt,p))).^2) ;

A hiba függvény megjelenítése

%% surface generator

% legyenek az iterációs változók

% s és d

Nk = 100; % rugómerevség

Nc = 100; % csillapítási tényező

k\_end = 5;

c\_end = 2;

k\_start = 0.1;

c\_start = 0.01;

alteration\_k = (k\_end-k\_start)/Nk;

alteration\_c = (c\_end-c\_start)/Nc;

E\_real\_plot = zeros(Nk,Nc);

ki = 0;

ci = 0;

f = waitbar(0,'Please wait...');

for x = k\_start:alteration\_k:k\_end

ki=ki+1;

ci = 0;

for y = c\_start:alteration\_c:c\_end

ci=ci+1;

E\_real\_plot(ki,ci) = E\_real\_np([x y]);

end

status = (ki\*ci)/(Nk\*Nc);

waitbar(status ,f ,'Please wait...');

pause(0.01)

end

close(f)

Graphical user interface, application

Description automatically generated

. ábra: Hiba megjelenítése

Következtetés:

A négyzetes hibára való optimalizálás nem vezetett sikerrel mert a függvénynek nincs minimuma.

## A rezgések terjedése

A motorban keletkező rezgések a motorhoz kapcsolt elemeken keresztül a motorházon a felfogató rendszeren keresztül terjednek.

* A felszerelt és a nem felszerelt motorok zaja másként hallható, más jellegűek.
* A felszerelés módja nagy mértékben játszik szerepet a motorok akusztikai vizsgálatában, mivel nagymértékben meghatározzák azok zajának "minőségét".

A rezgések a levegőben longitudinális hullámként terjednek a közegben alacsony és magas sűrűségű váltakozásokat előidézve, ami az ember számára hallható hatást eredményez. A szerkezet struktúráján a rezgések a szerkezeti elemek rugalmas alakváltozásain keresztül közlekednek. Ezek a rezgések nagy amplitúdó és aránylag kis frekvencia esetén tapintással érezhetőek.

Az egyre kényelmesebb és halkabb autóknál törekedni kell a rezgések csillapitására. A rezgések csillapításának egy lehetséges módszere, ha a rezgés energiáját hőenergiává alakítjuk. Ahol a csillapítás nem megoldható, ott a szerkezet merevségének növelésére és hangszigetelésre kell törekedni. Az előbbit a szerkezet célszerű mechanikai kiképzésével pl. bordák alkalmazásával, utóbbit pedig szivacsos anyagokkal célszerű megvalósítani. [1]

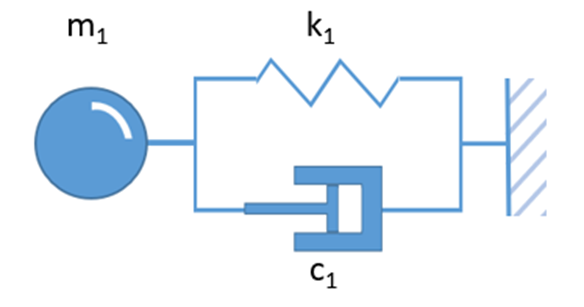
## A zaj csökkentése

A motorok felszerelésekor törekedni kell a rezgések terjedésének csökkentésére, ha a felfogatási pontokat a motor külső részeire pozícionáljuk a szerkezetünk stabilitása nő, ezzel csökkentve a rezgések mértékét. A felfogató pontokban érdemes valamilyen rugalmas alakváltozásra képes anyagot választani. Próbáljuk kerülni a motorok saját rezonanciáján való tartós használatát, ez az autóiparban egy kormány rásegítő szervó esetében elkerülhetetlen, mert a motor sebességét a kormányzás mindenkori sebessége határozza meg, így mindig lesz olyan fordulatszám az üzemi tartományban, ahol valamelyik periodikus összetevő frekvenciája megegyezik a szerkezet egy sajátfrekvenciájával, és így nagy amplitúdójú rezgést és/vagy zavaró zajt hozva létre. [1]

Aktív zajszűréssel is lehet csillapítani a motorok zaját, amihez, szoftveresen be kell avatkozni a vezérlésbe, amire szintén számos példát találunk az ipari gyakorlatban.

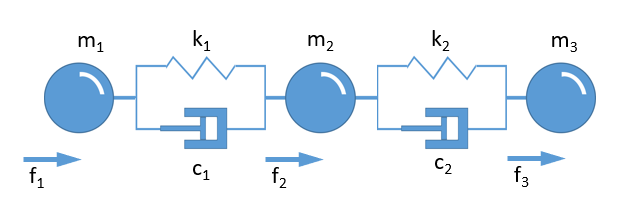
## A lengő rendszer modellezése

A lengő rendszereket egyszerű mechanikus oszcillátorként, egy tömeg és egy rugó kapcsolataként írjuk le. (*1. ábra*) A rugót egy *k* – rugóállandójú és *c* – csillapítási tényezőjű csillapító párhuzamosan kapcsolt eredőjével vesszük figyelembe, ami összeköttetésben van egy *m* – tömeggel. [2] [5]



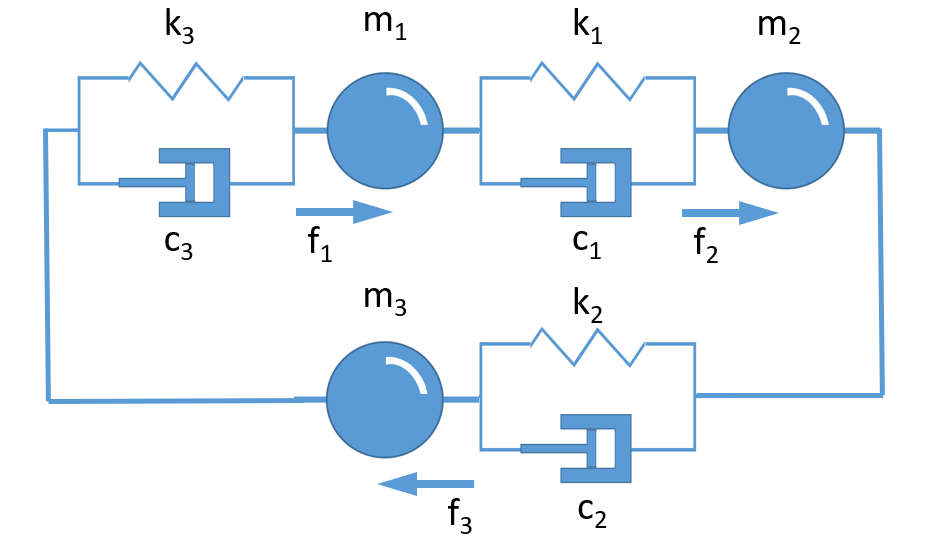
2. ábra: Csillapított tömeg rugó rendszer

A rendszer modelljét csillapított tömeg rugó rendszerek sorbakötött változatával modellezzük.



ábra: Három szabadságfokú rezgő rendszer

A mi esetünkben mivel a motor egy kör keresztmetszetű alkatrész, az utolsó és az első tömegpontokat összekötjük egy rugóval, ami a tömegek kellően nagy száma esetén jól közelíti a motor állórészén és házán tapasztalható modális viselkedést és hullámterjedést.



ábra: Három szabadságfokú „visszacsatolt rezgőkör”

# A tervezés részletes leírása, a választott megoldások indoklása

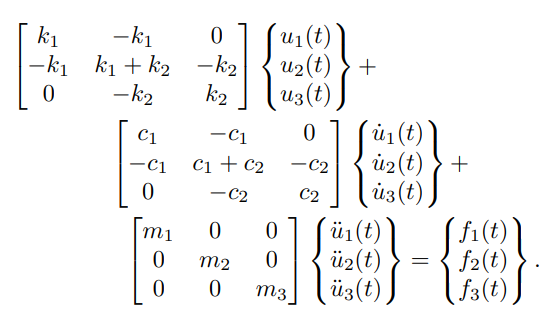
A feladatom egy olyan MatLab program írása volt, amellyel tetszőleges számú pontszerű tömeget lehet modellezni egymással sorba kapcsolt rugókkal. Az első és az utolsó tömeg egy rugóval és csillapítóval való összecsatolásával a modell egy a motor állórészéhez hasonló viselkedést fog leírni.

## A matematikai modell felállítása

Egy mechanikai rendszer mozgásegyenletét a következőképpen írhatjuk fel. A rendszer gerjesztése a tömegekre ható *f(t)* erők, a válasza pedig a tömegek *u(t)* elmozdulásai. A tömeget az *f1* erő, a *c1* csillapítás és a *k1* merevségből származó erők gyorsítják. [3]



Ahol az egyenletben a változó feletti pont a deriválást jelöli. A rugó rendszer egyenleteit mátrixalakba rendezve a következőt kapjuk. [3]



Az egyenletrendszer kompakt formája



A **K** a rendszer merevségmátrixa, **C** a csillapításmátrix, **M** pedig a tömegmátrix.

Ha feltételezzük, hogy minden időbeli változás *ejωt* alakú, akkor az idő szerinti deriválás a *jω* képzetes frekvenciával való szorzásba transzformálódik. Így a következő alakot kapjuk. [3]



A továbbiakban rendszer módusainak bevezetésével egy egyszerűbb megoldást kapunk, ahol a módusok olyan rezgésformák melyek fennmaradnak a rendszerben külső gerjesztés nélkül is. Ezeknek a kiszámításához keressük a frekvenciatartománybeli mozgásegyenlet megoldásait úgy, hogy alkalmazzuk a **C = 0**  és **f = 0** helyettesítéseket. Így a módusok a következő sajátértékfeladat megoldásai lesznek. [5]



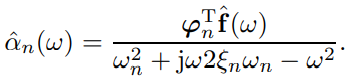
A sajátértékfeladat átrendezés után a következő alakra hozható.



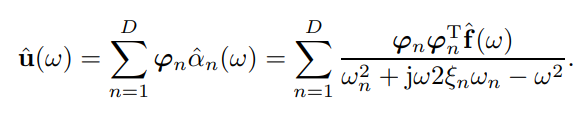
Ezen az alakon rögtön látszik, hogy a **M-1K** mátrix sajátvektorjai a ϕn vektorok, az ωn értékek pedig a sajátértékek. Egy rendszer elmozdulásai felírhatóak a módusalakokból képzett **φ** módusmátrix és **α(ω)** részesedési tényezők szorzataként. Ezt bevezetve a csillapított rendszer mozgásegyenletébe a következő egyenletet kapjuk.



A részletes levezetéstől eltekintve, az egyenletrendszer átrendezése után megkapjuk, hogy a részesedési tényezők csatolatlanok, vagyis az n-edik részesedési tényező csupán az n-edik módusalakjától függ.



A teljes elmozdulás kifejezése.



Az előző egyenlet azt mondja ki, hogy egy *D* szabadságfokú rendszer adott **f(ω)** gerjesztésre adott válasza felírható, mint *D*számú, egymástól független egyszabadságfokú rendszer elmozdulásainak szuperpozíciója, ahol a független egyszabadságfokú rendszerek a sajátfrekvenciákra vannak hangolva. [3]

## Rendszeregyenlet

Az előző fejezetben tárgyalt matematika implementálását két részletre bontva oldottam meg.

Az első függvénynek a „modusmatrixgenerator” elnevezést adtam, bemeneti értékként kéri az **m** tömeget, **k** rugómerevséget, **c** csillapítási tényezőt, **Ms** tömegpontok számát, és hogy a modellezni kívánt rendszerünk visszacsatolt-e. A megadott paraméterek alapján létrehozza a rendszermátrixokat, azokból pedig a MatLab **eig** parancsának segítségével kiszámolja a rendszer sajátértékeit és sajátvektorjait. A függvény kimeneti értékei az **M** tömegmátrix, **K** merevségmátrix, **C** csillapításmátrix, **FI** sajátvektormátrix és **OMEGA2** ω2 sajátértékek mátrixa.

A sajátvektorok és sajátértékek felhasználásával, már kitudtam számolni a rendszer részesedési tényezőit és elmozdulásait. Erre a célra létrehoztam egy „elmozdulasszamitas” nevű függvényt, ami bemenő paraméterként várja a **C** csillapításmátrixot, **FI** sajátvektormátrixot, **f** a tömegre ható erőt, **fnum** azt a számot, hogy hányas tömegre hat a gerjesztőerő, **omegakezdo** az omega változó kezdeti értékét, **Nomega** az omega változó felbontása, **Kiertekeles** az omega változó végső értékét, amit az irodalom **Ω** – val jelöl, ez az érték a sajátértékek maximumát szorozza, így a kiértékelési tartományt növeli és utolsó bemenő értékként kéri az **Ms** tömegpontok számát. Eredményként visszaadja az **U** elmozdulásmátrixot, **ALFA** részesedésitényező mátrixot, **omega** vektort és az **OMEGA** mátrixot, ami az **OMEGA2** gyöke.

A kód a mellékletek fejezetben megtalálható.

## A GUI megtervezése

A GUI tervezésénél az egyszerűségre törekedtem, hogy bárki tudja kezelni, aki tisztában van a móduselemzés alapjaival a paraméterek értelemszerű beírása után, egyszerűen tudja kezelni az alkalmazást.

Az program kezelőfelületét a MatLab appDesigner-ében készítettem el, mert egyszerűen és gyorsan lehet vele haladni.

Az app designer GUI létrehozásakor, automatikusan generálja a létrehozott elemekhez tartozó kódot, így abban csak a változókat, és funkciókat kell leírni, ami lényegesen gyorsítja a folyamatot.

Nagyon sok példavideót és példakódot meg lehet találni az appDesignerben, amik már egy-egy előre létrehozott programot tartalmaznak, ezeknek a programoknak a tanulmányozásával és megértésével kezdtem a GUI felépítését. [4]

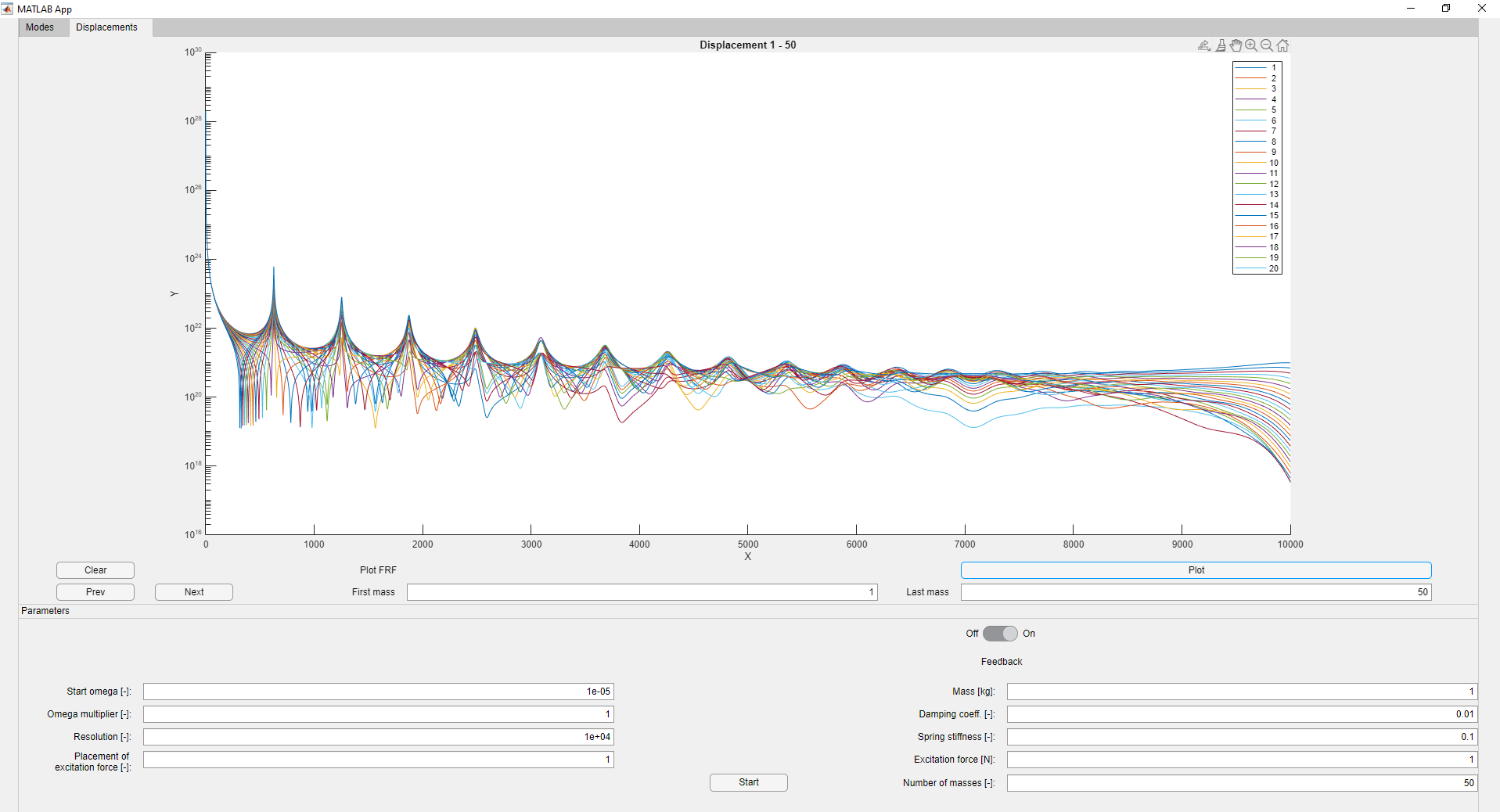
A bemenő paramétereket értelemszerűen a program erre megfelelő EditText-jeiben kell megadni, ha minden értéket kitöltöttünk a „Start” gomb lenyomásával elindíthatjuk a scriptet.

A kijelző két fülre van osztva. Az első fülön a módusok sajátvektorjait ábrázolom, a második fülön az elmozdulásokat logaritmikusan.

A „Next” és „Prev” gombokkal válthatunk a kijelzett diagrammok között, módusok szerint növekvő sorrendben.

A „Feedback” kapcsolóval lehet választani, hogy a modellünkben az utolsó és az első tömeg össze legyen-e csatolva egy rugóval. Ez a kódban a „visszacsatolt” változót állítja 0 -ra vagy 1 -re.

A „plot modes” részen be lehet állítani a kezdő és a záró módust, így több módus is megjeleníthető egyszerre.



ábra: 50 tömegből álló rugórendszer elmozdulásai

Az eredményeken látszik, hogy a sajátértékekben van a legnagyobb kitérés mind a két változatban.

# A megtervezett műszaki alkotás értékelése, továbbfejlesztési lehetőségek

A félév során megismerkedtem a modális dekompozícióval, a szervó motorok akusztikai vizsgálatának alapjaival, a rezgő rendszereket leíró egyenletekkel.

A program a céljának megfelel, egy rezgő rendszer sajátértékeit és elmozdulásait kellően egyszerűen lehet vele ábrázolni.

A megtervezett programot, később tovább lehetne fejleszteni, a beviteli adatok tömbös formátumban vagy akár függvény formájában is megadhatók legyenek, ha több helyen szeretnénk terhelni a rendszerünket tetszőleges erőeloszlással.

A későbbiekben ezek alapján egy valós rendszer rezgéseit is modellezhetjük, a mérési eredményekkel és végeselem szimulációval összevetve az analitikus modellünkkel validálhatjuk a szimulációt.

# Irodalomjegyzék

[1] Thomas Bertolini,Thomas Fuchs - Vibrations and Noises in Small Electric Motors - 2012

[2] Fiala Péter - A hangszerek fizikája - jegyzet

[3] Fiala Péter - Móduselemzés - Mérési leírás Stófiútechnika Laboratórium

[4] Matlab Help Center - <https://www.mathworks.com/help/matlab/app-designer.html>, megtekintve: 2021.06.05

[5] Agilent Technologies - Fundamentals of Modal Testing – 2000

[6] Gungl Szilárd – Elektromos kormányszervó motorok rezgésanalízise - 2018

# Mellékletek (dokumentációk: kapcsolási rajz, huzalozási rajz, beültetési rajz, forrás file-ok, kész program, installálási és használati útmutatók stb.)

## Modusmátrixgenerátor

function [M, K, C, FI, OMEGA2] = modusmatrixgenerator(m, k, c, Ms, visszacsatolt)

%m - tömeg

%c - csillapítás

%k - rugómerevség

%M - tömegmátrix

%C - csillapításmátrix

%K - rugómerevségmátrix

%Ms - tömegek száam

mm = m\*ones(1,Ms);

cm = c\*ones(1,Ms);

km = k\*ones(1,Ms);

M = eye(Ms).\*m;

K = zeros(Ms,Ms);

C = zeros(Ms,Ms);

I2 = [1 -1; -1 1];

for num = 1:Ms-1

Ksub = km(num)\*I2;

K(num:num+1,num:num+1) = K(num:num+1,num:num+1) + Ksub;

Csub = cm(num)\*I2;

C(num:num+1,num:num+1) = C(num:num+1,num:num+1) + Csub;

end

if visszacsatolt == 1

K(Ms,1) = K(Ms,1) -km(Ms);

K(1,Ms) = K(1,Ms) -km(Ms);

K(1,1) = K(1,1) + km(Ms);

K(Ms,Ms) = K(Ms,Ms) + km(Ms);

C(Ms,1) = C(Ms,1) -cm(Ms);

C(1,Ms) = C(1,Ms) -cm(Ms);

C(1,1) = C(1,1) + cm(Ms);

C(Ms,Ms) = C(Ms,Ms) + cm(Ms);

end

%FI - 1-re normált sajátvektor - Fi\_n -ek módusalakok

%OMEGA2 - sajátértékek - omega\_n\_negyzetek

[FI,OMEGA2] = eig(M\K);

end

## Elmozdulásszámítás

function [U, ALFA, omega, OMEGA] = elmozdulasszamitas(C, FI, OMEGA2, f, fnum, omegakezdo, Nomega, Kiertekeles, Ms)

j = sqrt(-1);

F = zeros(Ms,1);

F(fnum) = f;

KSZIOMEGAx2 = diag(FI.'\*C\*FI);

OMEGA = diag(sqrt(OMEGA2));

%omegakezdo = 0;

%Nomega = 1000;

%Kiertekeles = 1;

omega = linspace(omegakezdo,OMEGA(end)\*Kiertekeles,Nomega).';

ALFA = zeros(Nomega,Ms);

U = zeros(Ms,Nomega);

for n = 1:Ms

ALFA(:,n) = FI(:,n).'\*F./(OMEGA(n)^2+j\*omega\*KSZIOMEGAx2(n)-omega.^2);

U = U + FI(:,n).\*ALFA(:,n).';

end

end