Regression Task 2020II

Nikolay Prieto Ph.D(c)

October 6, 2020

1 Regression task for Intelligence Systems

Regression tasks in Artificial Intelligence are not common, they are being used in fields as Finance, Surrogate Modelling [1] and Sensitivity Analysis [2]. This time your work will be focused in design, especifically, design of transtibial (Ankle-foot) prosthesis.

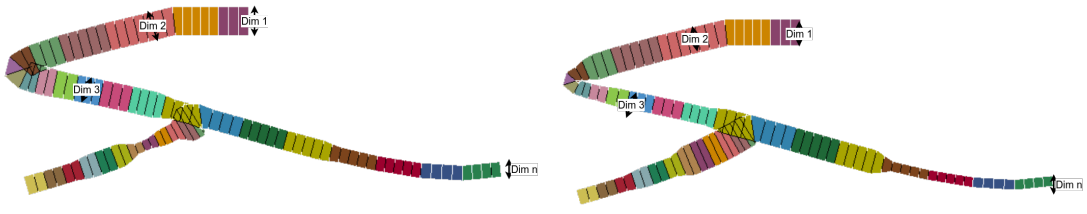
The Design Variables in prosthesis have been studied along decades. Those variables can be grouped in the following according to [3]:

- Siffness/Flexibility.
- Damping.
- Roll-over characteristics
- Active push-off in late stance phase

The half part of the course will do the design focused on the prosthesis stiffness, in other words, variation of the thickness along the laminates.The other half of the course will be focused on the Roll-over characteristics, this is achieved with variations in the prosthesis shape.

1.1 Design for thickness

As it was explained, there are two datasets, those are simulations already performed with a Latin Hyper Cube (LHS) experiment [**?**] design. The first one is about the influence of the thickness variation with respect to the outputs. For thickness 120 simulations were run, varying the thickness on different locations of the prosthesis. The following figure 1 explains how thickness is being varied along the device. In total, there are twelve dimensions as features for the dataset.



[6]:
```
import pandas as pd
Thickness_df = pd.read_csv('AnkleFoot_thickness.csv', index_col=0)
```

```
Thickness_df.index = Thickness_df['Experiment No.'].astype('int64')␣
 ↪#Estabiliching index experiment
Thickness_df_df =  Thickness_df.drop(['Experiment No.'], axis=1) #Dropping index␣
 ↪column
Thickness_df.head().iloc[:,:10]
```

[6]:

Experiment No.	dim 1	dim 2	dim 3	dim 4	dim 5	dim 6	dim 7	\
1	14.696	19.446	6.146	2.346	9.312	10.737	18.179	
2	10.262	2.979	10.262	17.546	10.896	17.071	10.104	
3	15.013	9.629	18.021	8.679	6.146	19.921	11.688	
4	9.629	7.571	1.396	8.204	17.229	13.904	1.238	
5	4.404	3.771	15.329	18.654	13.904	10.896	8.521	

Experiment No.	dim 8	dim 9	dim 10
1	2.029	7.254	9.471
2	11.688	8.046	15.013
3	15.329	8.521	15.487
4	12.954	5.196	14.221
5	18.338	7.413	1.871

The dimensions are given in mm.

1.2 Design for shape

On the other hand, we have manipulated the foot shape due to the control points in the cartesian space. The following figure indicates which dimension corresponds to each point:

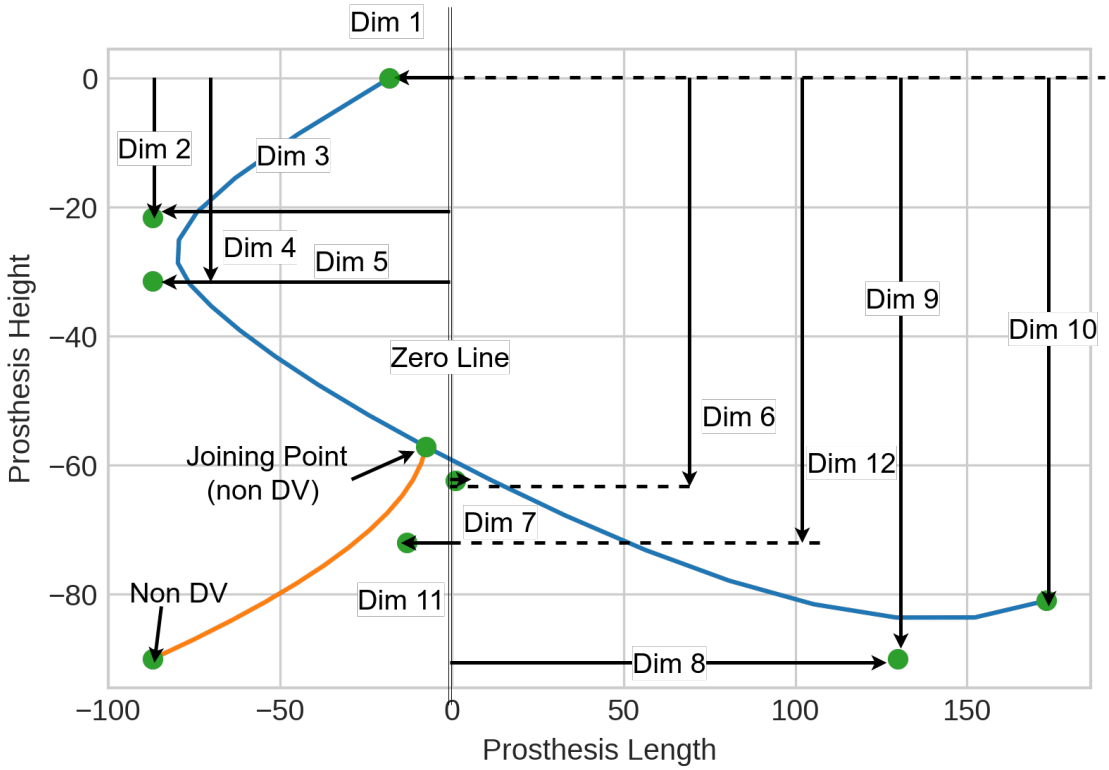
The respective dimensions of the dimensions mentioned above, are depicted in the following dataframe:

[5]:
```
Shape_df = pd.read_csv('AnkleFoot_shape.csv', index_col=0)
Shape_df.index = Shape_df['Experiment No.'].astype('int64') #Estabiliching index␣
 ↪experiment
Shape_df =  Shape_df.drop(['Experiment No.'], axis=1) #Dropping index column
Shape_df.head().iloc[:,:12]
```

[5]:

Experiment No.	dim 1	dim 2	dim 3	dim 4	dim 5	dim 6	dim 7	dim 8	dim 9	\
1	0.738	1.987	0.119	1.762	0.590	48.25	0.148	0.625	1.079	
2	0.862	1.979	0.461	1.142	0.230	-55.25	0.988	0.886	0.989	
3	0.388	0.824	0.449	1.630	0.874	-7.55	0.742	0.764	0.946	
4	0.788	2.181	0.399	1.398	0.966	31.15	0.587	0.627	0.981	
5	0.162	2.134	0.491	1.181	0.902	14.95	0.872	0.672	0.789	

Experiment No.	dim 10	dim 11	dim 12



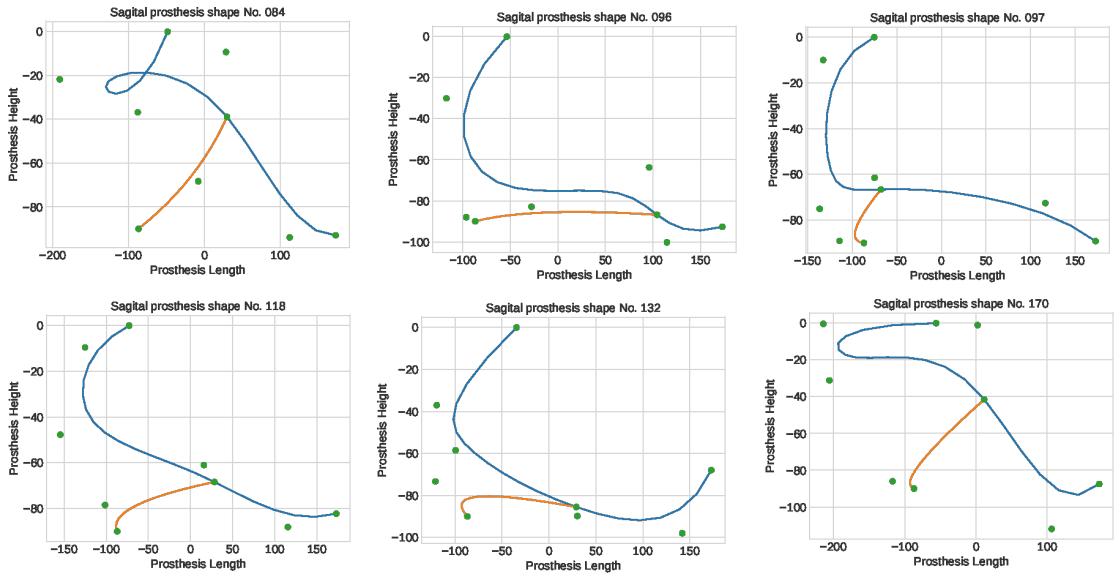
Experiment No.			
1	0.942	1.136	0.753
2	0.609	1.286	0.948
3	0.796	1.489	0.963
4	0.862	1.016	0.854
5	1.041	1.271	0.992

In order to maintain the proportion, most of the prosthesis preserve 25% of the shape to the left of the zero line (being this straight to the tibia position) and the distal point may be positioned up to 75% of the total length. Hence, dimensions 1, 3, 5 and 11 are measured proportionally with respect to the maximum left size, which means 25% of the total length. In other words, if dim 1 is 0.738 and the total length of the foot is 250 mm, the position of this point in x will be $0.738 \times -0.25 \times 250mm = -46mm$

Similarly, dimension 8 will be evaluated with respect to the remaining 75% right proportion, whereas for dimensions 2, 4, 9, 12 and 10 will be the proportion of the max height of the foot, it means: dimension 9 is 1.079 for a 130 mm prosthesis height, in milimeters is translated to: $1.079 \times 130mm = 140.27mm$.

Varying those points can give us many design shapes, as shown in Fig. 3.

The unique point that does not work relative to the foot proportion are the dims 6 and 7. this point may vary either before or after the zero line, that is why those points are being controlled in mm.



1.3 Outputs

Either for shape or thickness design, the outputs are the same in both dataframes. First of all, in the dynamic process you will need to know how much stress is exerted in the prosthesis, this is an important variable that cannot exceed the yield stress of the material. Also, these laminates are composite materials, they are exposed to cyclic loops and the safety for the user must be guaranteed. These variable in the column **MPa** in the dataframe. For your design, consider that you cannot exceed the $900MPa$

[8]:
```
Shape_df.head().iloc[:,12:]
```

[8]:

Experiment No.	MPa	Error_GRF	MechWork	ErrorMom	RoM
1	639.5904	0.035172	0.002248	2.442129	2.542826
2	678.8516	0.015182	0.015457	1.480014	19.319449
3	697.6824	0.017248	8.633470	1.303972	27.260984
4	617.7568	0.037799	7.744428	2.079074	26.291950
5	698.1921	0.044156	14.864731	1.539913	31.093966

1.3.1 Error in the GRF and Moment trajectory

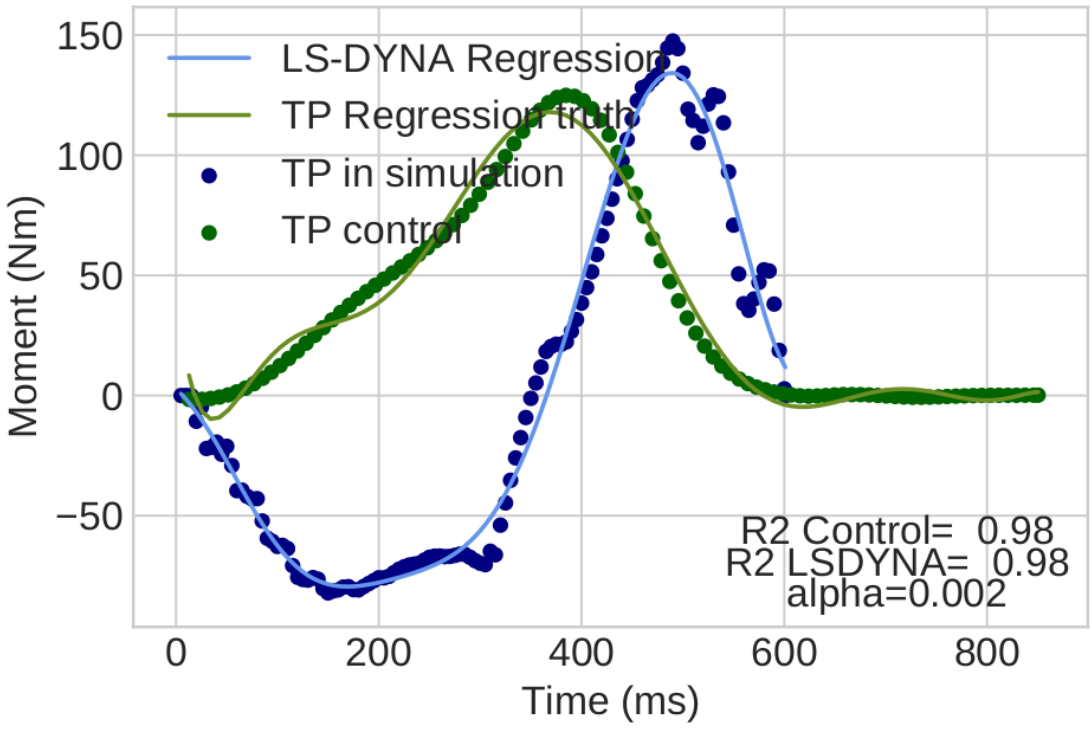
As you see in the data, there are two columns with names **Error_GRF** and **ErrorMom**. Those are the Mean Square Error (MSE) measurements for the Ground Reaction Force (GRF) and dynamic moment trajectory, respectively.

- GRF Error.

They were measured through reading raw data on each simulation. The desired output is the blue one. For the raw data we have applied a butterworth filter in order to smooth the real output and measure the error through the minimum Mean Square Error. See Fig **??** for more details

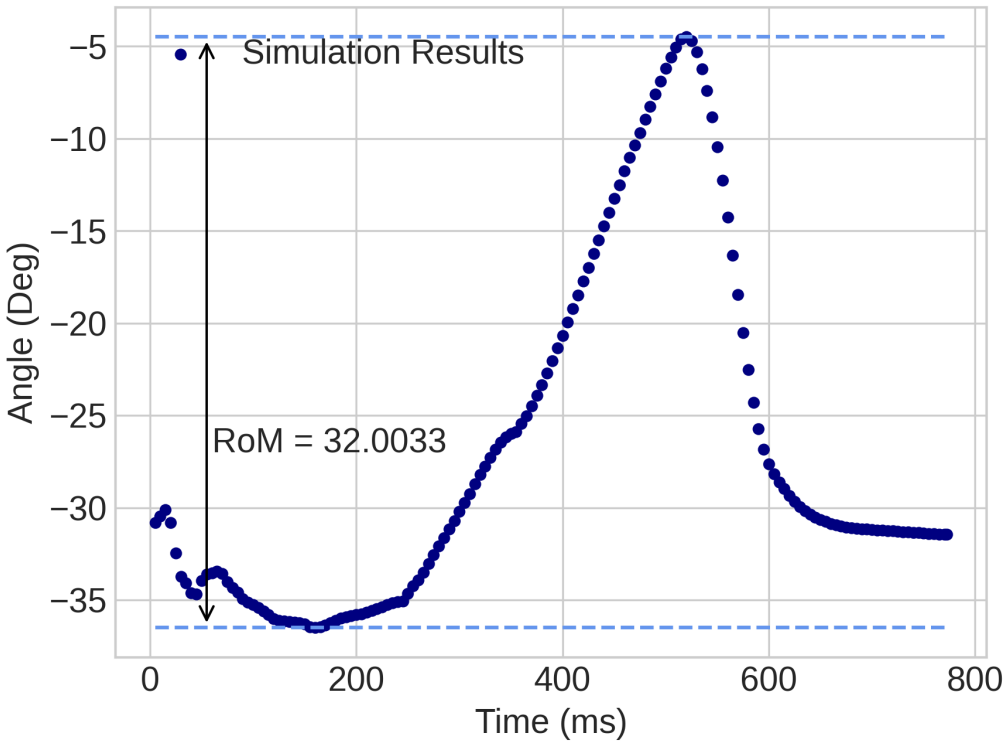
- Errors in moments.

One of the most important outputs is the torque generated in the ankle joint. The main purpose is to replicate a natural ankle dynamics with the prosthesis thus we are comparing the torque generated in the prosthesis (blue) with a target (green line). therefore, we also have measured the MSE between both functions, see Fig. **??**.



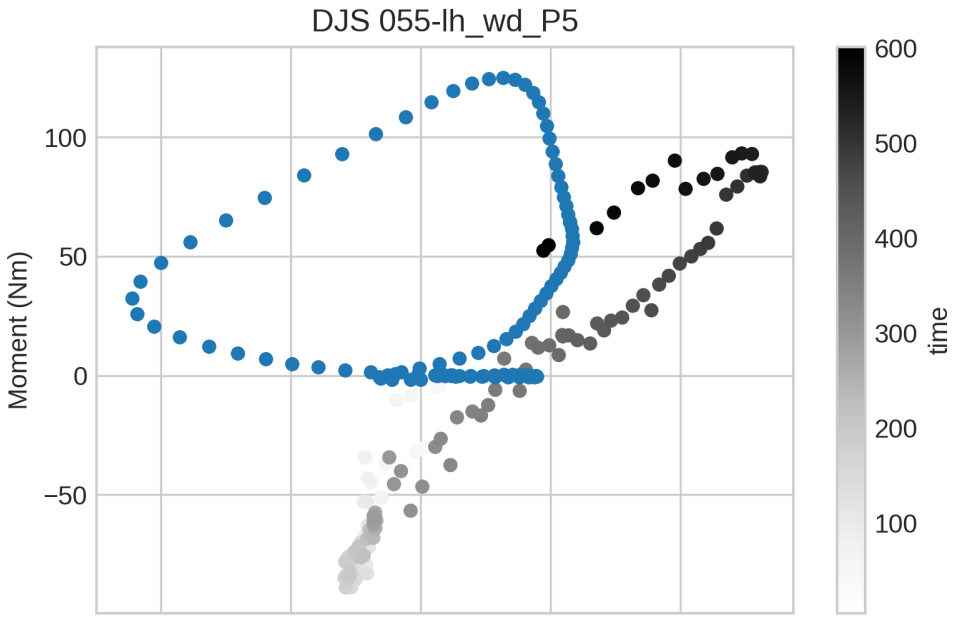
1.3.2 Range of Motion

Another important feature is to see how compliant is the prosthesis, this is measured through the Range of Motion (RoM), which is obtained through the maximum amplitude in the angle dynamic behavior, as shown in Fig **??**, the units are degrees.



1.3.3 Mechanical work

The remaining output is the mechanical work, this is obtained through the loop integral of the moment respect to the angle. In other words, is the area of the closed loop of the moment-angle plot. See Fig **??**.



1.4 Data analysis

As you can percieve in the dataset, this is a multi-label regression problem. Then, on each output you may apply a regression algorithm to fit the data and predict another future value. The following is an example using random forest for only one output in the shape design dataframe:

[12]:
```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import RandomForestRegressor

#Only for Von misses stress
X_train, X_test, Y_train, Y_test = train_test_split(Shape_df.iloc[:,:12],
                                                    Shape_df.loc[:,'MPa'],␣
 ↪test_size=0.2)

# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 50, stop = 2000, num = 50)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt', None]
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 200, num = 20)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 3, 4, 5, 10, 20]
```

```
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4, 8]
# Method of selecting samples for training each tree
bootstrap = [True, False]

# Creating the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
# Use the random grid to search for best hyperparameters
# First create the base model to tune

rf = RandomForestRegressor(random_state = 42)
# Random search of parameters, using 3 fold cross validation,
# search across 100 different combinations, and use all available cores
rf_random = RandomizedSearchCV(estimator=rf, param_distributions=random_grid,
                           n_iter = 50, scoring='neg_mean_absolute_error',
                           cv = 3, verbose=1, random_state=42, n_jobs=-1,
                           return_train_score=True)


rf_random.fit(X_train, Y_train)

#concatenating best params obtained with the accuracy
results = rf_random.best_params_
model = rf_random.best_estimator_
```

```
Fitting 3 folds for each of 50 candidates, totalling 150 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed:   34.4s
[Parallel(n_jobs=-1)]: Done 150 out of 150 | elapsed:  1.2min finished
```

the best model for this output is:

[13]:
```
model
```

[13]:
```
RandomForestRegressor(max_depth=50, max_features=None, min_samples_leaf=2,
                      min_samples_split=3, n_estimators=408, random_state=42)
```

1.5 Sensitivity Analysis

One of the important characteristics of the ensemble algorithms is to be able to measure the importance of the variables, in some cases there are features that are not relevant to the output. Thus

that features (or column) can be omited in future analysis. These is an example of the permutation importance with the previuos model:

[ ]:
```
from sklearn.inspection import permutation_importance
model.fit(X_train, Y_train)
perm = permutation_importance(model, X_train, Y_train, n_repeats=10,
                              random_state=42, n_jobs=-1)
```

The higher the number the most important is, some methods can give values greater than 1, however most of them vary between 0 and 1.

[17]:
```
pd.Series(perm['importances_mean'])
```

[17]:
```
0     0.021687
1     0.028002
2     0.031456
3     0.114980
4     0.037173
5     0.044599
6     0.159323
7     0.021985
8     0.126900
9     0.882708
10    0.025654
11    0.015289
dtype: float64
```

1.6 Objective design

Your goal is to design the best prosthesis as possible with the given data, building metamodels (ML algorithms) for each output, you are free to implement whatever algorithm available in the literature (XGBoost, RF, SVM, etc). in order to choose the best design you must consider the constrains and goal.

- Constrains:
- The maximum Von Misses stress cannot exceed $1000MPa$
- The Range of Motion should be between 22 and 28 degrees.
- The GRF error should be lower than 0.03.
- Goals:
- The least Moment error as possible.
- The Maximum Mechanical work as possible.

1.7 Activities

This is an individual task, your job is the following:

- Perform the best ML algorithm that fits the Test set greater than 0.92. Remind the advantages and disadvantages of each algorithm. For instance, one disadvantage of RF is that they cannot predict values greater or lower than the values implied in the dataset, however they provide good scores whatever the metric implemented. On the other hand, SVM is able to predict values higher than the bounds, however scaling the data is needed to provide coherent results. (40% of the grade)
- Perform Sensitivity analysis procedure: Do this analysis with the following techniques: Permutation importance, feature importance and one the Salib packages. Compare the results and give conclusions. (20% of the grade)
- Generate an estrategy for obtaining the best values. For instance, you can set a meshgrid with numpy and predict all possible configurations, later you can filter the best. After that, compare if you have obtained higher values for the goals and preserving the constrains. (10% of the grade)
- Provide 5 new samples, I will simulate your dimensions and see how far are from the predicted answers (10% of the grade).
- Presentation and visualization: Remind that good plots and coherent conslusions makes your work stronger (20% of the grade)

.

Regards

2 References

References

[1] Alexander I J Forrester, András Sóbester, and Andy J Keane. *Engineering Design via surrogate optimization*. 2008.

[2] Bertrand Iooss and Paul Lemaître. Uncertainty Management in Simulation-Optimization of Complex Systems. *Operations Research/ Computer Science Interfaces Series*, 59:101–122, 2015.

[3] Andrew Hansen and Felix Starker. Prosthetic Foot Principles and Their Influence on Gait. *Handbook of Human Motion*, pages 1–15, 2016.