

Prevodová charakteristika

Obsah

1	O meraní prevodovej charakteristiky	1
1.1	Úlohy	2
1.2	Surové dáta...	3
1.3	Spracovanie surových dát	3
2	Doplňkový text: o aproximácii prevodovej charakteristiky lineárnym modelom	6
2.1	Model	6
2.2	Jednoduché hľadanie parametrov (koeficientov) polynómu	6
2.2.1	Funkcia polyfit	6
2.2.2	Funkcia polyval	7
2.2.3	Používanie modelu	7
2.3	Čo vlastne robí funkcia polyfit	7
2.3.1	Preurčená sústava rovníc	8
2.3.2	Názorná ukážka	10
2.3.3	Analytické riešenie	10
2.4	Všetko uvedené realizované v jazyku Python	11

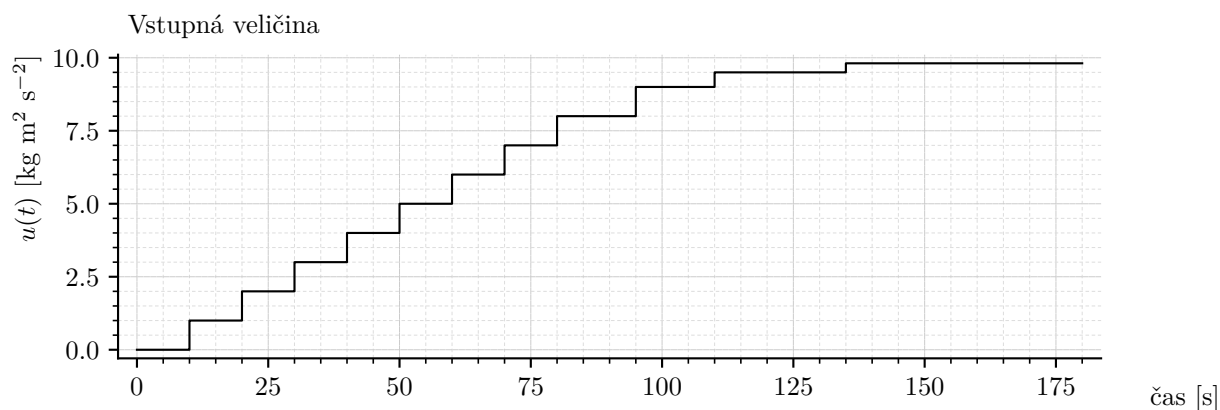
1 O meraní prevodovej charakteristiky

Prevodová charakteristika je závislosť ustálených hodnôt výstupnej veličiny od ustálených hodnôt vstupnej veličiny. Zhruba povedané. Prevodová charakteristika, niekde sa nazýva aj statická charakteristika, teda charakterizuje systém len v ustálených stavoch. Neobsahuje informáciu o dynamike systému.

V nasledujúcom sa pokúsime načrtnúť meranie prevodovej charakteristiky - tu sa však nebude naozaj niečo merať, ale reálny systém bude nahradený simulovaným. Proces získavania „surových“ dát, ktoré sú potrebné pre určenie prevodovej charakteristiky a proces spracovania týchto dát bude však rovnaký akoby išlo o reálny systém.

Tabuľka 1: Hodnoty vstupného signálu

čas [s]	hodnota [$\text{kg m}^2 \text{s}^{-2}$]
0	0,0
10	1,0
30	2,0
20	3,0
40	4,0
50	5,0
60	6,0
70	7,0
80	8,0
95	9,0
110	9,5
135	9,81

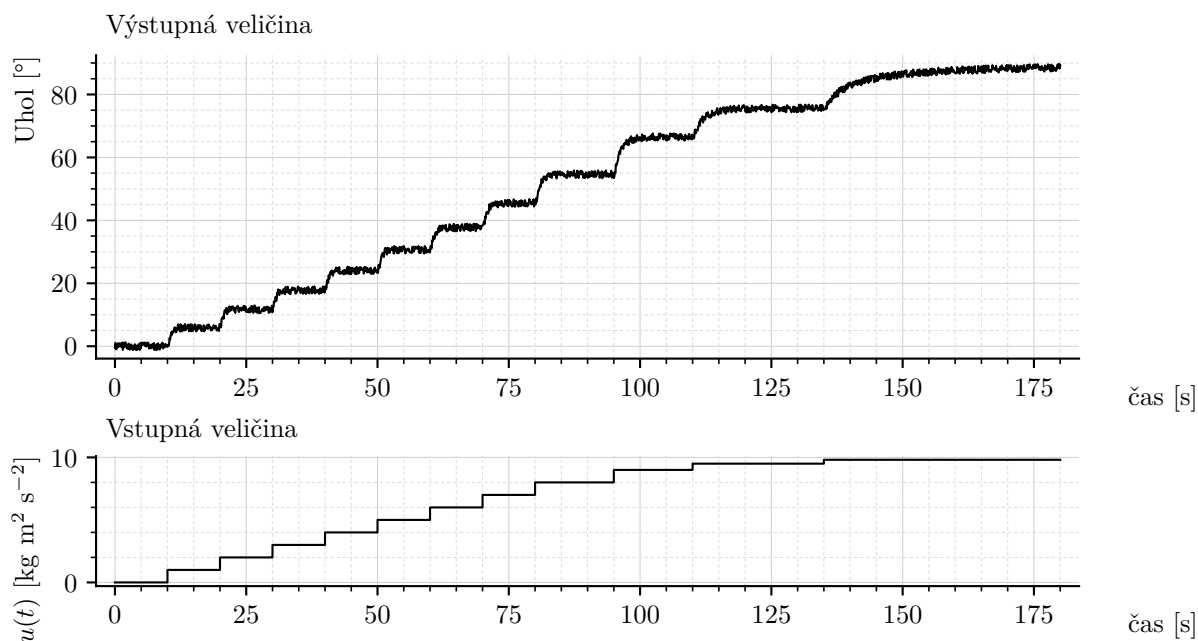


Obr. 1: Priebeh vstupného signálu

Systém, ktorý bude predmetom skúmania, ktorého prevodovú charakteristiku budeme merať, je kyvadlo, tak ako bolo už skôr opísané. Navyše však bude k výstupnej veličine simulovaného systému pridaný čitateľovi neznámy šum. Dôvodom je lepšie napodobenie situácie v akej by bol čitateľ ak by sa zaoberal výstupnou veličinou reálneho systému.

1.1 Úlohy

1. Vytvorte numerickú simuláciu časového priebehu výchylky kyvadla. Je zrejmé, že nie je dôvod uvažovať nenulové začiatkové podmienky, a teda $\varphi = 0$ [rad] a $\dot{\varphi} = 0$ [rad/s]. Vstupný signál $u(t) = 0$ [kg m² s⁻²] sa bude meniť podľa potreby.
2. Odmerajte (simulačne) prevodovú charakteristiku uvažovaného systému.



Obr. 2: Surové dáta

1.2 Surové dáta...

Ako už bolo uvedené, predmetom záujmu sú ustálené hodnoty výstupného signálu. Ak na vstup privedieme vstupný signál s nejakou konštantnou vstupnou hodnotou, následne počkáme istý čas nech sa výstupný signál ustáli, tak potom môžeme odčítať (odmerať) ustálenú hodnotu výstupného signálu. Takto sa získa jeden bod z prevodovej charakteristiky.

Hneď potom je však možné zmeniť hodnotu vstupného signálu a opäť čakať, kým sa výstup ustáli.

Tento postup, hodnoty vstupného signálu a doby, počas ktorých sa „čaká“ na ustálenie výstupu, možno vyjadriť tabuľkou nasledovne. Prvý stĺpec je čas, v ktorom sa zmení (prepne) vstup a druhý stĺpec je hodnota (konštanta), na ktorú sa zmení (prepne).

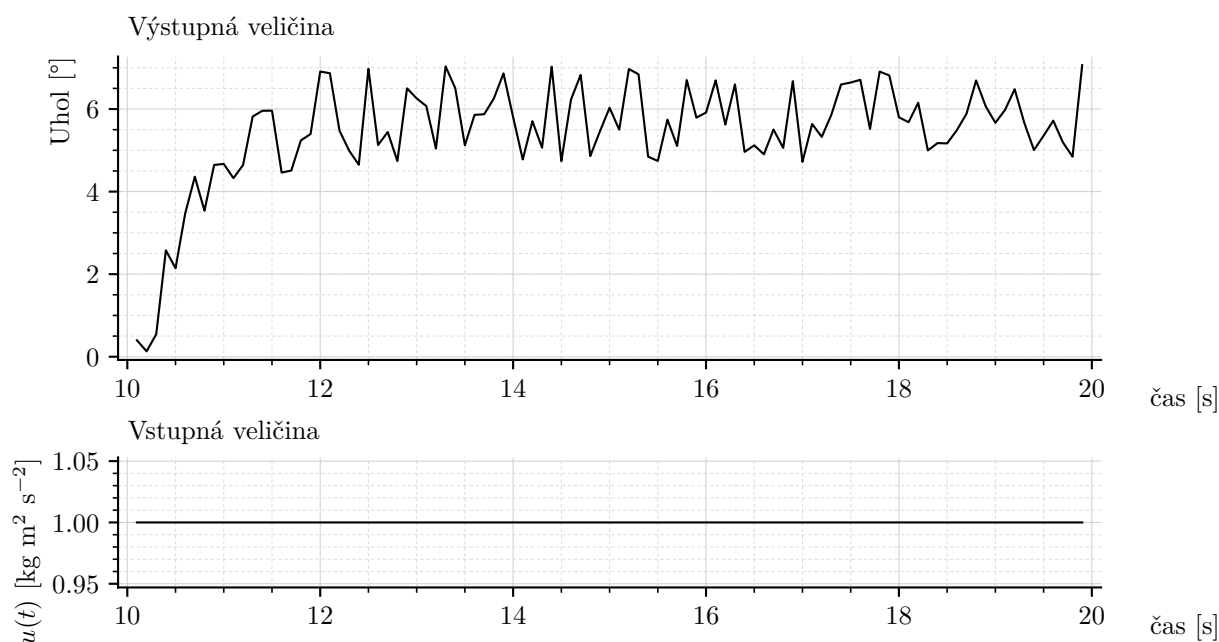
Z uvedeného je zároveň jasné, že interval vstupných hodnôt, pre ktorý zisťujeme ustálené hodnoty výstupu je 0 až 9,81 [kg m² s⁻²]. Iné by vzhľadom na konkrétne kyvadlo, ktoré sa tu uvažuje, malo pramalý význam.

Vstupný signál tak ako je definovaný tabuľkou 1 možno znázorniť ako na obr. 1.

Simulujeme teraz priebeh výstupnej veličiny kyvadla (výchylky kyvadla) pri uvedenom vstupnom signály. Výsledok je na obr. 2.

1.3 Spracovanie surových dát - odčítanie („z grafu“, z dát) bodov prevodovej charakteristiky

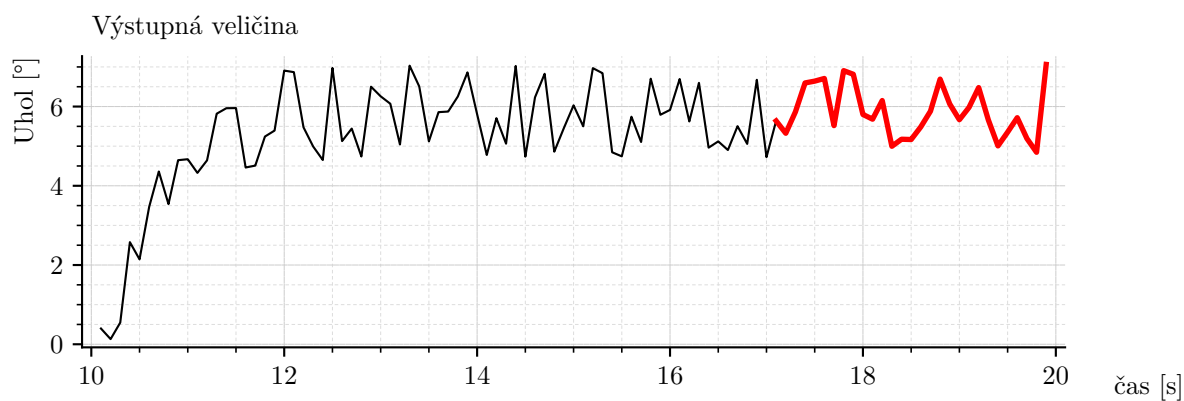
Zo surových dát je potrebné získať jednotlivé body prevodovej charakteristiky. To v prvom rade znamená byť schopný odčítať ustálenú hodnotu výstupného signálu (zo surových dát). Pre ilustráciu, venujme sa nameraným dátam od desiatej sekundy po dvadsiatu sekundu, teda pre interval, počas ktorého bola na vstupe hodnota $u = 1$ [kg m² s⁻²]. Táto časť dát je nakreslená na obr. 3.



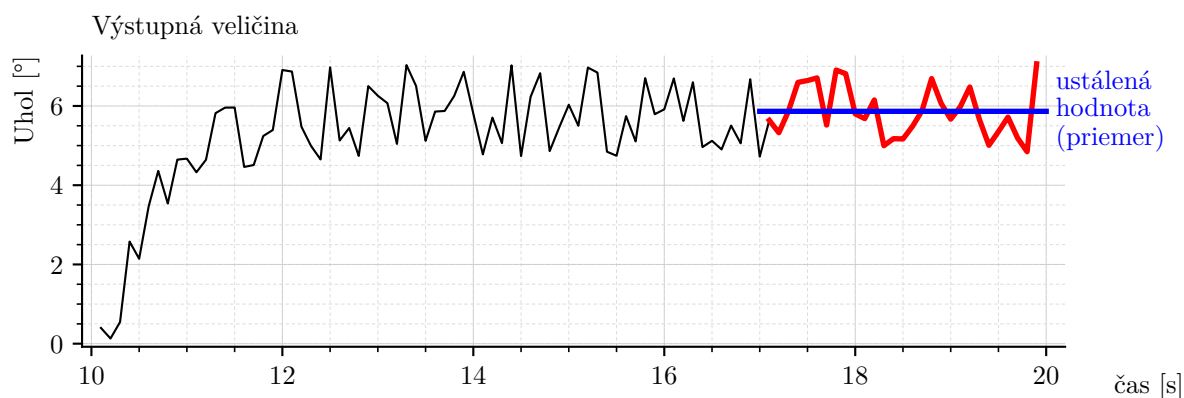
Obr. 3: Surové dáta - detail

Z obr. 3 je zrejmé, že v poslednej tretine intervalu je už možné považovať hodnotu výstupu za ustálenú. Vyznačme túto časť dát - viď obr. 4.

Otázka je, ako z vyznačeného úseku určiť jednu hodnotu - ustálenú hodnotu. Prirodzenou voľbou je urobiť priemer z vybranej (vyznačenej) časti dát. Priemer je vyznačený na obr. 5

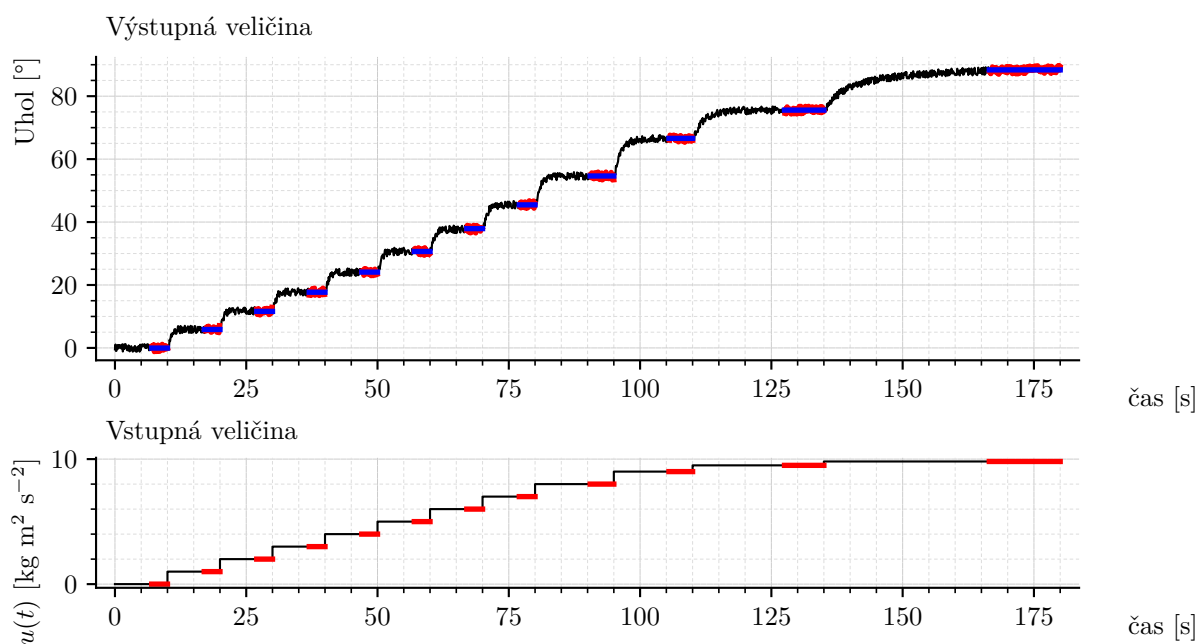


Obr. 4: Surové dáta - detail



Obr. 5: Surové dáta - detail

Rovnako je samozrejme možné postupovať pri všetkých bodoch prevodovej charakteristiky. Všetky ustálené hodnoty odčítané zo „surových dát“ sú znázornené na obr. 6.

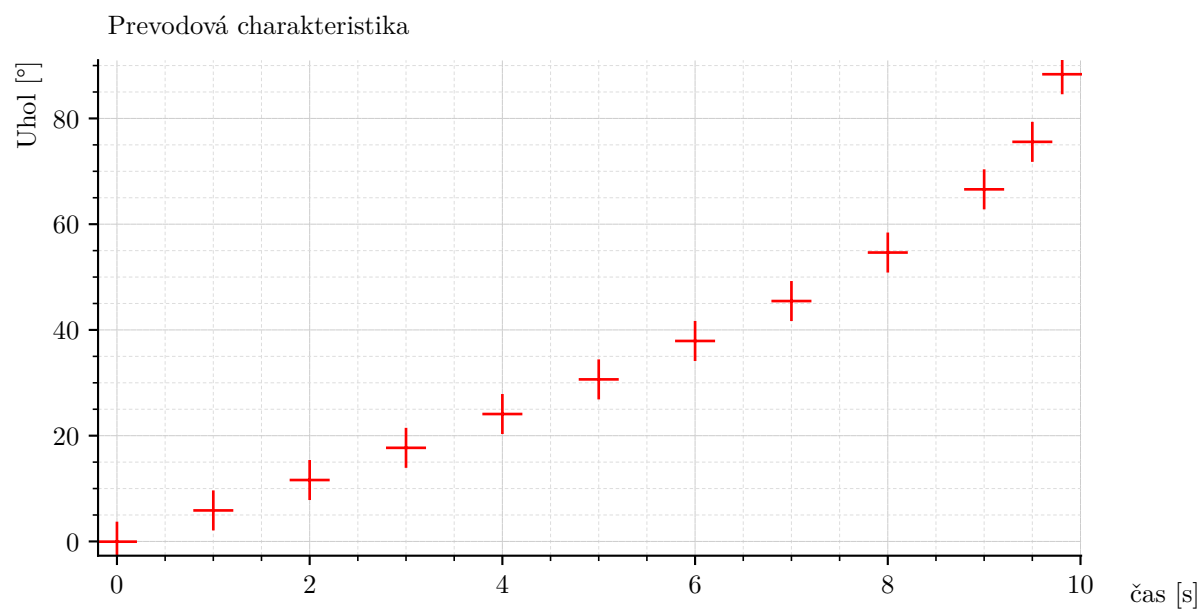


Obr. 6: Surové dáta

Odčítané hodnoty sú následne uvedené v tabuľke 2. Prevodová charakteristika je graficky znázornená na obr. 7.

Tabulka 2: Prevodová charakteristika

vstup [$\text{kg m}^2 \text{s}^{-2}$]	výstup [$^\circ$]
0,0	0,00691
1,0	5,7
2,0	11,8
3,0	17,7
4,0	24,3
5,0	30,6
6,0	37,6
7,0	45,2
8,0	54,5
9,0	66,5
9,5	75,6
9,81	88,7



Obr. 7: Prevodová charakteristika

2 Doplnkový text: o aproximácii prevodovej charakteristiky lineárnym modelom

Majme nameranú prevodovú charakteristiku, dáta sú v tabuľke 2. Prevodová charakteristika je graficky znázornená na obr. 7.

2.1 Model

Body prevodovej charakteristiky zodpovedajú istej vlastnosti reálneho systému (reálne existujúceho systému). Zodpovedajú závislosti výstupu systému od vstupu systému, samozrejme v ustálenom stave. Nameraných je však len niekoľko bodov. V týchto bodoch je daná vlastnosť systému známa. Čo však v prípade ak by bolo potrebné poznať danú vlastnosť mimo nameraných bodov? Teda mimo hodnôt vstupu, pre ktoré bola prevodová charakteristika nameraná.

Aj pre tieto účely je výhodné využiť model. Model reálnej vlastnosti systému. Samotnej vlastnosti systému zodpovedá nameraná závislosť (prevodová charakteristika). Aproximáciou tejto závislosti je možné získať model.

Model nech je v tomto prípade matematický vzťah, funkčná závislosť, istý predpis... Ak hodnota na vstupe modelu bude rovnaká ako hodnota na vstupe reálneho systému, potom hodnota na výstupe modelu nech je „približne rovnaká“ ako hodnota reálna. Toto nech však platí pre všetky namerané body prevodovej charakteristiky. Teda model nech sa približne zhoduje s reálnymi dátami. Ak toto platí v nameraných bodoch, potom to, zrejme, platí aj v iných bodoch. Platí to pre akúkoľvek hodnotu na vstupe modelu - že výstup modelu sa približne zhoduje s reálnym výstupom systému.

Takto všeobecne opísaný model možno skonkretizovať napríklad nasledovne: Nech modelom je polynomiálna funkcia

$$\hat{y} = \Theta_3 u^3 + \Theta_2 u^2 + \Theta_1 u + \Theta_0 \quad (1)$$

kde „vstup modelu“ je u a „výstup modelu“ je \hat{y} . Parametrami modelu sú koeficienty (čísla) Θ_3 , Θ_2 , Θ_1 a Θ_0 .

Mimochodom, ide o lineárny model. Parametre modelu sú v lineárnom vzťahu k „signálom“ modelu (k vstupom modelu).

2.2 Jednoduché hľadanie parametrov (koeficientov) polynómu

V tejto časti sa použije MATLAB pre nájdenie parametrov (koeficientov) polynomiálnej funkcie (1). Pre tieto účely majme premennú `prevodChar`, ktorej prvý stĺpec sú hodnoty vstupnej veličiny a druhý stĺpec sú hodnoty výstupnej veličiny. Teda obr. 7 by sme v MATLABe nakreslili takto

```
figure(1);  
plot(prevodChar(:,1), prevodChar(:,2), 'r');
```

2.2.1 Funkcia polyfit

Funkcia `polyfit` slúži vo všeobecnosti na hľadanie koeficientov polynómu (polynomiálnej funkcie) daného stupňa tak aby polynomiálna funkcia aproximovala dané dáta (napr. nameranú x-y závislosť). Kritériom pre hľadanie koeficientov je minimalizácia štvorcov (kvadrátov) odchýlok medzi nameranou hodnotou a jej aproximáciou. Presnejšie, minimalizácia sumy štvorcov odchýlok.

Použitie funkcie `polyfit` v tu uvažovanom konkrétnom prípade by bolo nasledovné

```
polyKcoef = polyfit(prevodChar(:,1),prevodChar(:,2), 3)
```

a premenná `polyKcoef` obsahuje hodnoty koeficientov polynómu. Rovnica (1) s nájdenými koeficientami je:

$$\hat{y} = 0,1111u^3 - 1,1150u^2 + 8,9102u - 1,1770 \quad (2)$$

2.2.2 Funkcia polyval

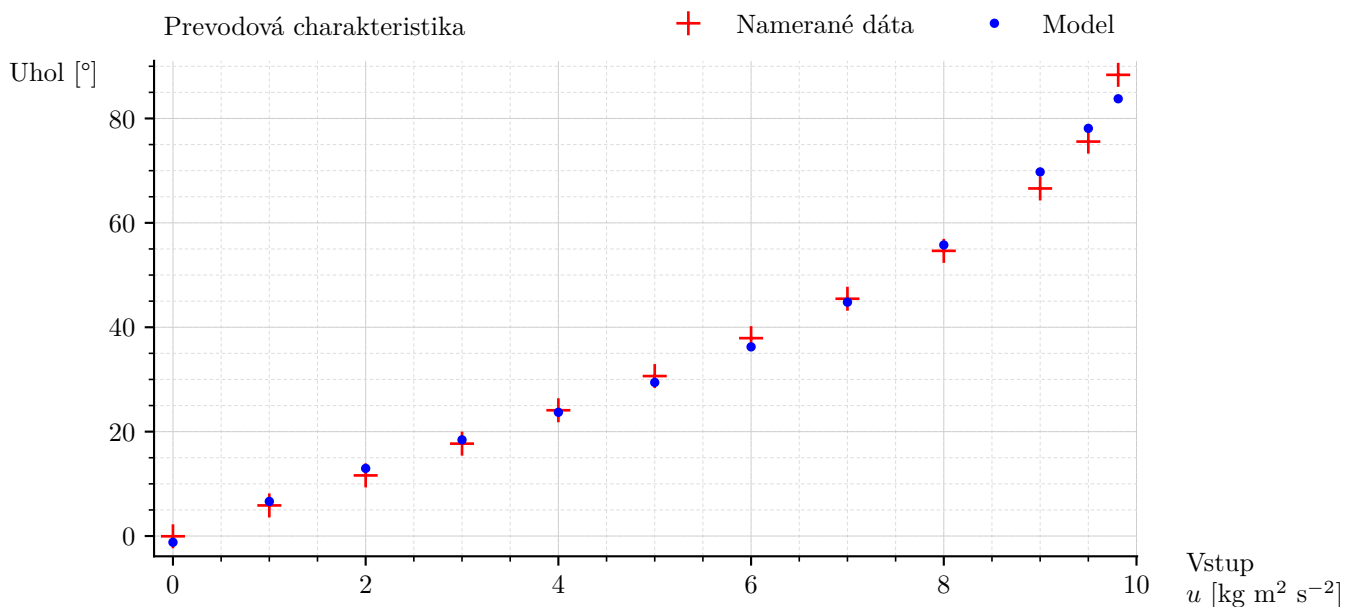
Pre vypočítanie hodnôt (výstupov) \hat{y} pre požadované vstupy u je možné použiť funkciu `polyval`. Ak teda chceme ku každému vstupu, pre ktorý bola nameraná hodnota výstupu, vypočítať jej aproximáciu \hat{y} podľa modelu (2), potom stačí zavolať:

```
y_hat = polyval(polyKoeff, prevodChar(:,1));
```

Obrázok, na ktorom sú naraz zobrazené namerané dáta aj výstup modelu (2) možno nakresliť nasledovne:

```
figure(2);  
hold on;  
plot(prevodChar(:,1), prevodChar(:,2), '+r')  
plot(prevodChar(:,1), y_hat, '.b')
```

Samotný obrázok by bol podobný obr. 8



Obr. 8: Prevodová charakteristika

2.2.3 Používanie modelu

Model, samozrejme, umožňuje vypočítať aproximáciu reálneho výstupu pre akúkoľvek hodnotu vstupu - nie len pre hodnoty vstupov, pre ktoré boli merané hodnoty skutočného systému. Vypočítajme výstupy modelu pre tieto vstupné hodnoty (dané vektorom):

```
u_ine = [0:0.1:9.81];
```

Teda zavoláme funkciu `polyval`.

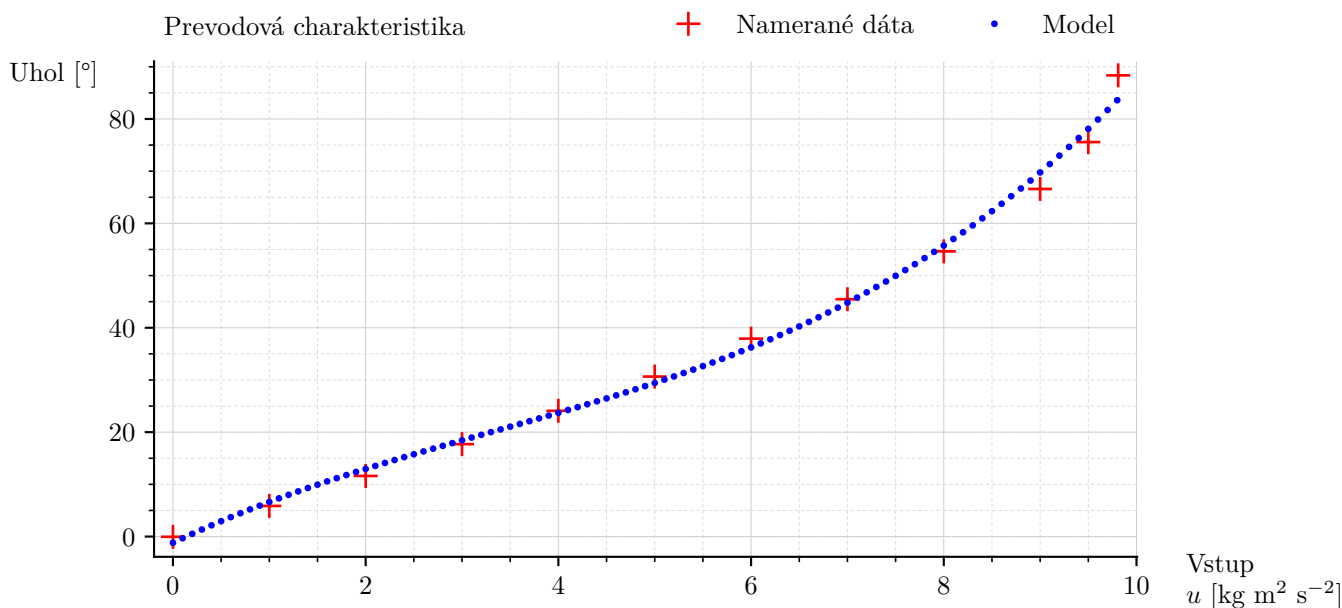
```
y_ine_1 = polyval(polyKoeff, u_ine);
```

Nakreslime obrázok (viď obr. 9)

```
figure(3);  
hold on;  
plot(prevodChar(:,1), prevodChar(:,2), '+r')  
plot(u_ine, y_ine_1, '.b')
```

2.3 Čo vlastne robí funkcia polyfit

V MATLABe funkcia `polyfit` zostaví istú sústavu rovníc a následne sa použije, pre MATLAB ikonický, príkaz „backslash“: `\`. Dokonca niektorí ľudia nosia na tričku nápis (niektorí nie, pretože také tričko nemajú): $x = A \backslash b$;



Obr. 9: Prevodová charakteristika

O tom, že funkcia `polyfit` používa príkaz „backslash“, sa čitateľ môže ľahko presvedčiť otvorením súboru, v ktorom je funkcia definovaná. Stačí zadať príkaz (do Command Window):

```
open polyfit
```

Otvorí sa súbor `polyfit.m` a na riadku číslo 65 sa používa príkaz „backslash“.

2.3.1 Preurčená sústava rovníc

Tu sa pokúsime naznačiť, akú sústavu rovníc zostaví a potom rieši funkcia `polyfit` aby našla to čo hľadá - hľadá koeficienty polynómu v zmysle istého kritéria (minimalizovať sumu štvorcov odchýlok medzi nameranými dátami a výstupom modelu).

Podobne sa dá postupovať nie len práve pri hľadaní nejakých koeficientov polynómu. Podobný postup je možné použiť pri hľadaní parametrov lineárneho modelu vo všeobecnosti. Pritom však dôkladnejšie vysvetlenie pojmu *lineárny model* a *identifikácia* jeho parametrov je pradáleko nad rámec tohto textu, predmetu, a asi aj bakalárskeho študijného programu.

Akokoľvek, aspoň náznak pre ilustráciu istotne má zmysel.

Preurčená sústava rovníc je sústava algebraických rovníc, taká, že počet rovníc je väčší ako počet neznámych. Tu by bolo možné pridať značné množstvo detailov. Na niektoré si azda čitateľ spomína zo svojho štúdia lineárnej algebry.

Ak má sústava rovníc toľko neznámych koľko rovníc, potom je riešenie jednoznačné (iba ak by nebolo).

Ak má sústava rovníc menej neznámych ako je počet rovníc, potom riešenie pri prvom intuitívnom priblížení neexistuje. Avšak sú prípady, keď je potrebné „aspoň nejako“ vyriešiť takúto sústavu rovníc. Taký prípad sa vyskytuje napríklad aj pri používaní funkcie `polyfit` tak ako bola použitá vyššie.

Preurčenú sústavu rovníc je možné riešiť v zmysle tzv. *metódy najmenších štvorcov*. Algebraický pohľad na vec tu vynechajme, ale aplikovanie tejto „metódy“ sa v našom prípade prejaví tak, že suma štvorcov odchýlok medzi nameranými dátami a výstupom modelu je minimálna možná.

Pripomeňme, že sa zaoberáme prípadom, keď modelom je polynomiálna funkcia v tvare

$$\hat{y} = \Theta_3 u^3 + \Theta_2 u^2 + \Theta_1 u + \Theta_0 \quad (3)$$

čo je rovnica (1) a teda „vstup modelu“ je u a „výstup modelu“ je \hat{y} . Parametrami modelu sú koeficienty (čísla) $\Theta_3, \Theta_2, \Theta_1$ a Θ_0 .

Ak by sme na vstup modelu priviedli hodnotu zodpovedajúcu prvému bodu prevodovej charakteristiky (súradnice bodu označme $[u_1, y_1]$) potom na výstupe modelu by bola hodnota, ktorú označme \hat{y}_1 . Presnejšie napísané:

$$\hat{y}_1 = \Theta_3 u_1^3 + \Theta_2 u_1^2 + \Theta_1 u_1 + \Theta_0 \quad (4)$$

Pre druhý bod prevodovej charakteristiky by to bolo

$$\hat{y}_2 = \Theta_3 u_2^3 + \Theta_2 u_2^2 + \Theta_1 u_2 + \Theta_0 \quad (5)$$

atď. . .

Samozrejme, model má v každom prípade len jednu sadu parametrov: $\Theta_3, \Theta_2, \Theta_1$ a Θ_0 . Tie, samozrejme, ostávajú rovnaké bez ohľadu na to aký je vstup modelu.

Model má aproximovať reálne dáta (reálnu vlastnosť systému). Odchýlky medzi modelom a dátami v jednotlivých nameraných bodoch sú:

$$e_1 = y_1 - \hat{y}_1 \quad (6a)$$

$$e_2 = y_2 - \hat{y}_2 \quad (6b)$$

$$e_3 = y_3 - \hat{y}_3 \quad (6c)$$

$$\vdots \quad (6d)$$

Odchýlky teda sú e_1, e_2 , atď. Za \hat{y}_i , pre $i = 1, \dots, N$, možno dosadiť vyššie uvedené, teda

$$e_1 = y_1 - \Theta_3 u_1^3 + \Theta_2 u_1^2 + \Theta_1 u_1 + \Theta_0 \quad (7a)$$

$$e_2 = y_2 - \Theta_3 u_2^3 + \Theta_2 u_2^2 + \Theta_1 u_2 + \Theta_0 \quad (7b)$$

$$e_3 = y_3 - \Theta_3 u_3^3 + \Theta_2 u_3^2 + \Theta_1 u_3 + \Theta_0 \quad (7c)$$

$$\vdots \quad (7d)$$

$$e_N = y_N - \Theta_3 u_N^3 + \Theta_2 u_N^2 + \Theta_1 u_N + \Theta_0 \quad (7e)$$

Ak by sa model presne zhodoval s dátami, potom by platilo

$$e_1 = 0 \quad (8a)$$

$$\vdots \quad (8b)$$

$$e_N = 0 \quad (8c)$$

Toto by však bol úplne ideálny prípad a dá sa tušiť, že takýto prípad nenastane.

Akokoľvek, čo sa týka číselných hodnôt e_i považujeme ich na chvíľku za známe, také, že sú nulové. Potom práve vznikla sústava rovníc. Ich počet je N . Neznáme v rovniciach sú $\Theta_3, \Theta_2, \Theta_1$ a Θ_0 , teda 4 neznáme. Všetko ostatné je v rovniciach známe, teda známe sú e_i, y_i aj u_i . V tomto prípade, počet rovníc je väčší ako počet neznámych. To je veľmi dôležité, avšak prečo je to dôležité je ďalšia vec čo začína byť nad rámec tohto textu.

Uvedená sústava rovníc sa dá zapísať maticovo:

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} u_1^3 & u_1^2 & u_1 & 1 \\ u_2^3 & u_2^2 & u_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ u_N^3 & u_N^2 & u_N & 1 \end{bmatrix} \begin{bmatrix} \Theta_3 \\ \Theta_2 \\ \Theta_1 \\ \Theta_0 \end{bmatrix} \quad (9)$$

Pre zjednodušenie zavedme označenia:

$$0 = y - H\Theta \quad (10)$$

kde je snáď zrejmé čo je čo. Inak zapísané:

$$H\Theta = y \quad (11)$$

Toto je (pri predpoklade o vyššom počte rovníc ako je neznámych) sústava rovníc, ktorú je možné riešiť v zmysle metódy najmenších štvorcov (kvadrátov). Podrobnosti sú ďaleko pradáleko nad rámec tohto textu. Výsledkom však je, že vektor y a vektor $H\Theta$ (áno je to vektor) sa „rovnajú“ v takom zmysle, že suma kvadrátov odchýlok medzi ich súradnicami je minimálna možná. Inak, a užitočnejšie, povedané, $y - H\Theta$ je vlastne e . Vektor e obsahuje všetky odchýlky medzi modelom a nameranými dátami. Vektor e sú ale aj odchýlky medzi súradnicami vektorov, o ktorých sme práve hovorili. A suma kvadrátov týchto odchýlok je minimálna! Takže aj suma odchýlok medzi modelom a dátami je minimálna!

2.3.2 Názorná ukážka

Zostavme teraz, v MATLABe, maticu H a vektor y podľa toho čo ukazuje rovnica (9).

```
H = [prevodChar(:,1).^3, prevodChar(:,1).^2, prevodChar(:,1).^1,
      ones(length(prevodChar(:,1)), 1)];
y = prevodChar(:,2);
```

Neznámu nazvime Θ a riešenie preurčeného systému rovníc (11) hľadáme príkazom:

```
Theta = H\y
```

Výsledkom je:

```
Theta =
    0.1111
   -1.1150
    8.9102
   -1.1770
```

a to sú také isté hodnoty koeficientov ako sa uvádza v rovnici (2).

2.3.3 Analytické riešenie

V predchádzajúcej časti sme použili istý príkaz („backslash“) pre hľadanie riešenia preurčeného systému algebraických rovníc v zmysle metódy najmenších štvorcov. Čo presne robí tento príkaz, aby našiel riešenie, je opäť raz ďaleko nad rámec tohto textu. Ak by sme však hľadali riešenie formulovaného problému analyticky, je možné ukázať (ale nebudeme to robiť), že to vedie na nasledujúcu rovnicu

$$\Theta = (H^T H)^{-1} H^T y \quad (12)$$

Táto rovnica je známa pod názvom *Gaussov vzorec*.

Pri použití tejto rovnice je potrebné urobiť inverziu istej matice - ako je zrejmé... Práve táto inverzia matice môže byť (a v praxi veľmi často je) z hľadiska numerických výpočtov veľmi náročnou operáciou (a napríklad aj toto rieši efektívnejšie príkaz „backslash“).

Použime tento vzorec (rovniciu) priamo v MATLABe:

```
Theta_1 = inv(H'*H)*H'*y
```

Výsledkom je:

```
Theta_1 =
    0.1111
   -1.1150
    8.9102
   -1.1770
```

a to sú opäť také isté hodnoty koeficientov ako sa uvádza v rovnici (2).

2.4 Všetko uvedené realizované v jazyku Python

Nasledujúce je ukážkou príkazov v jazyku Python, kde sa využíva knižnica (modul) `numpy`. Príkazy realizujú to isté ako bolo uvedené doposiaľ (pomocou príkazov v MATLABe). Kreslenie obrázkov je tu vynechané...

```
import numpy as np

# použitie funkcie polyfit (z modulu numpy)
polyKoeef = np.polyfit(prevodChar[:,0],prevodChar[:,1], 3)

# funkcia polyval pouzita pre vypocet aproximacie nameranych
# hodnot vystupu
y_hat = np.polyval(polyKoeef, prevodChar[:,0])

# vypocitanie aproximacie vystupu pre rozne vstupne hodnoty
u_ine = np.arange(0,9.81,0.1)
y_ine_1 = np.polyval(polyKoeef, u_ine)

# zostavenie matice H a vektora y
H = np.hstack([
    (prevodChar[:,0].reshape(-1,1))**3,
    (prevodChar[:,0].reshape(-1,1))**2,
    prevodChar[:,0].reshape(-1,1),
    np.ones([prevodChar[:,0].shape[0],1]),
])

y = prevodChar[:,1].reshape(-1,1)

# Riesenie preurceného systému rovníc v zmysle metódy najmenších
# stvorcov
Theta = np.linalg.lstsq(H, y, rcond=None)[0]

# Použitie Gaussovhó vzorca
Theta_1 = np.matmul(np.matmul(np.linalg.inv(np.matmul(H.T, H)), H
    .T), y)
```