

Tugas Kelompok Analisis Numerik : Stewart Problem

Kelompok B09

Maret 2019

1 Introduction

Stewart Platform adalah sebuah permukaan yang ditopang oleh enam buah kaki dengan panjang yang dapat berubah-ubah dan dihubungkan dengan sendi prismatic. *Stewart Platform* ini biasanya digunakan untuk menciptakan ruang tiga dimensi yang diinginkan oleh suatu objek dengan meletakkan muatan di berbagai titik di berbagai kemiringan. Salah satu contoh pemanfaatan *Stewart Platform* adalah alat simulasi penerbangan (FlightGlobal/Archive, 2019).

Stewart Platform ini dimungkinkan untuk dianalisis secara sederhana dalam bentuk dua dimensi. *Stewart Platform* digambarkan sebagai sebuah bidang manipulator berbentuk segitiga yang diatur posisinya oleh tiga buah penopang pada bidang *Cartesian*. Bidang manipulator tersebut memiliki sisi $L1$, $L2$ dan $L3$ dengan γ adalah sudut yang menghadap ke $L1$; sedangkan panjang tiap penopang dilambangkan dengan $P1$, $P2$ dan $P3$.

Dengan bantuan beberapa variabel lain yang akan disebutkan pada bagian **Content**, dalam proyek ini kami ingin menyelesaikan beberapa permasalahan yang disebutkan, antara lain :

1. Membuat kode *Octave/Matlab* dengan parameter bebas θ dan konstanta lainnya dimana dihipotesiskan $f(\theta) = 0$ saat $\theta = \pi/4$ atau $\theta = -\pi/4$.
2. Membuat kode *Octave/Matlab* untuk plotting $f(\theta)$ di interval $[-\pi, \pi]$.
3. Menghasilkan kembali sebuah pose yang diberikan dengan fungsi *plot* yang ada di *Octave/Matlab*.

4. Mencari kemungkinan bentuk pose dari parameter yang telah ditentukan.
5. Mencari nilai parameter tertentu dari jumlah pose yang telah ditentukan.

2 Why?

Selain untuk mengerjakan tugas dari mata kuliah Analisis Numerik yang kami ambil, kami juga tertarik dengan bagaimana harmonisasi antar penopang yang dapat menciptakan suatu kemiringan pada bidang manipulator. Kami juga tertarik untuk menyelesaikan permasalahan kinetika langsung *Stewart Platform* untuk model dua dimensi.

3 Content

3.1 Part 1

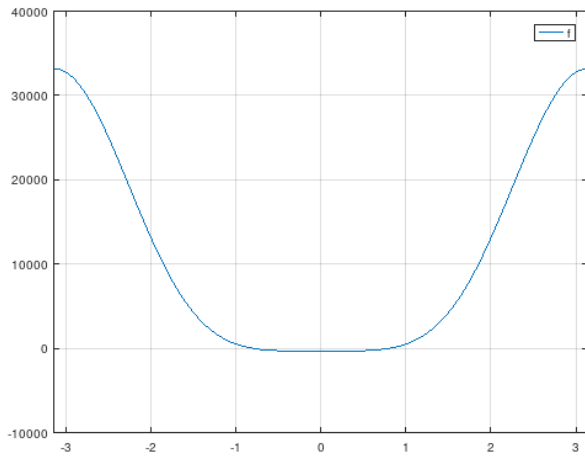
Pada bagian ini, kami mengimplementasikan fungsi yang diminta untuk dibuat menjadi sebuah kode. Fungsi ini memiliki parameter bebas θ sebagai input dan parameter terikat berupa output dari $f(\theta)$. Selain itu, terdapat panjang penopang ($p1, p2, p3$), sisi platform ($L1, L2, L3$) dan sudut *platform* menghadap $L1$ (γ) yang menjadi konstanta di persamaan ini. Kode yang kami gunakan dapat dilihat pada lampiran yang kami kumpulkan bersama dengan laporan ini (fungsi 1).

Setelah itu, untuk menguji kode yang kami buat, kami memasukkan nilai $\theta = \frac{\pi}{4}$ dan $\theta = -\frac{\pi}{4}$ pada fungsi tersebut dengan harapan akan menghasilkan nol. Ternyata kami mendapatkan nilai $f(\theta) = -4.547473508864641e - 13$ untuk kedua nilai

θ yang dimasukkan ke dalam fungsi tersebut. Terlihat bahwa nilai tersebut mendekati nol yang berarti fungsi yang kami buat cukup akurat.

3.2 Part 2

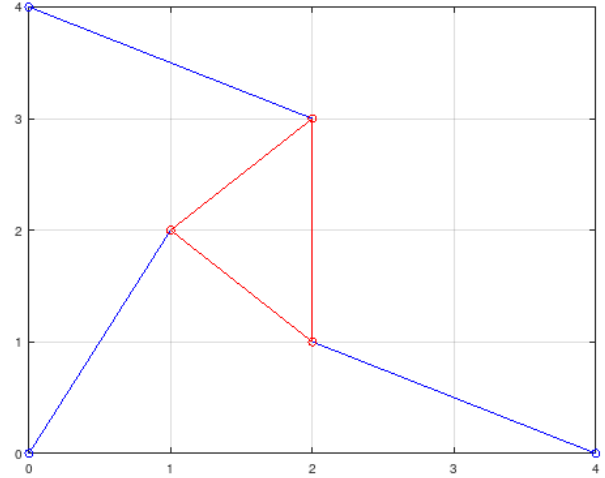
Untuk membuat plotting fungsi $f(\theta)$ pada $\theta = [-\pi, \pi]$, kami menggunakan kode `fplot(@f, [-pi,pi])` yang menghasilkan plot sebagai berikut :



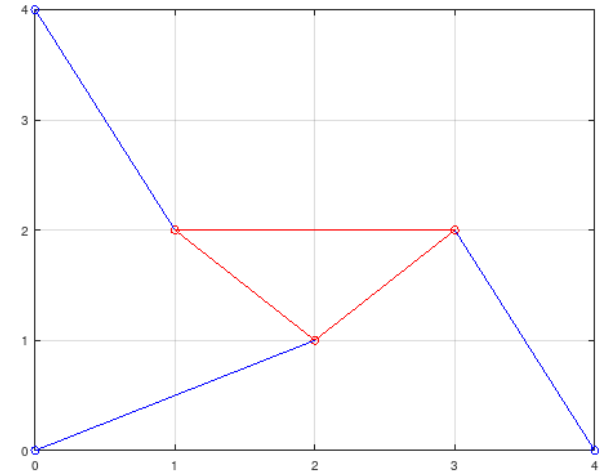
Dari gambar diatas, terlihat bahwa terdapat akar (output mendekati nilai nol) pada nilai $\theta = \pm \frac{\pi}{4}$.

3.3 Part 3

Disini, kami diminta untuk membuat plot *Stewart Platform* di bidang *Cartesian* (2D) yang sama dengan contoh yang telah diberikan pada soal sebelumnya. Untuk itu, kami membuat kode yang mengimplementasikan fungsi yang dapat menampilkan plot tersebut (fungsi 2). Fungsi yang menerima parameter berformat $(x1, x2, x3, y1, y2, y3)$ kami masukkan nilai $(1, 2, 2, 2, 1, 3)$ untuk gambar plot pertama dan nilai $(2, 3, 1, 1, 2, 2)$ untuk gambar plot kedua.



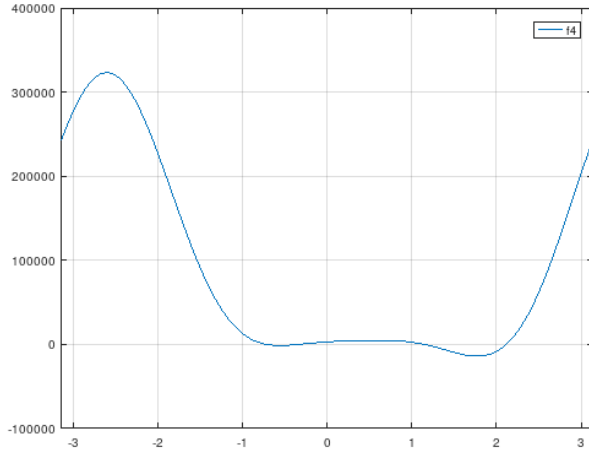
Gambar plot pertama



Gambar plot kedua

3.4 Part 4

Bagian keempat dari laporan ini menyajikan pemodelan 2 dimensi dari *Stewart Platform* dengan parameter konstan yang diganti dari kode perhitungan sebelumnya dan kami mencoba mencari 4 pose yang berbeda dari *Stewart Platform* tersebut. Sebelum mencari pose - pose tersebut, kami mencoba mencari plot $f(\theta)$ di $\theta = [-\pi, \pi]$ terlebih dahulu dengan menggunakan (fungsi 3).

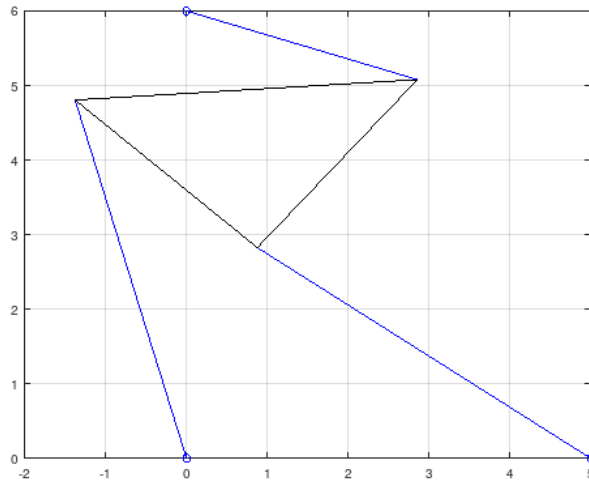


plotting $f(\theta)$

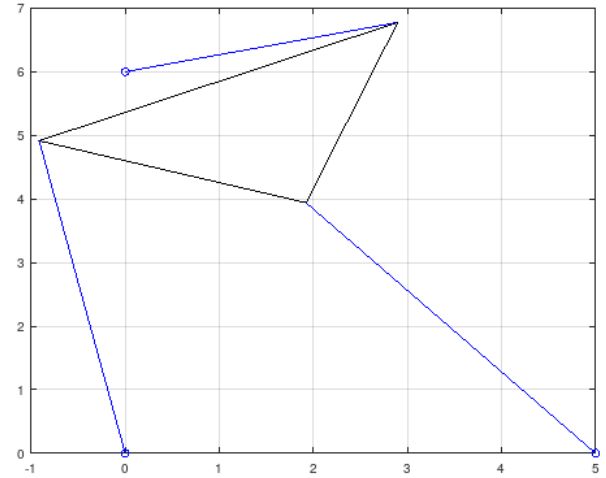
Terlihat bahwa grafik fungsi tersebut memiliki empat akar. Keempat akar tersebut akan kami cari nilainya dengan menggunakan *Bisection Method* (fungsi 4). Untuk mendapatkan akar, batas yang kami gunakan berturut - turut adalah $[-1, -0.5]$; $[-0.5, 0]$; $[1, 1.5]$; dan $[2, 2.5]$. Hasil dari pengakaran tersebut adalah :

- akar pertama : $x_1 = -0.72085$
- akar kedua : $x_2 = -0.33101$
- akar ketiga : $x_3 = 1.1437$
- akar keempat : $x_4 = 2.1159$

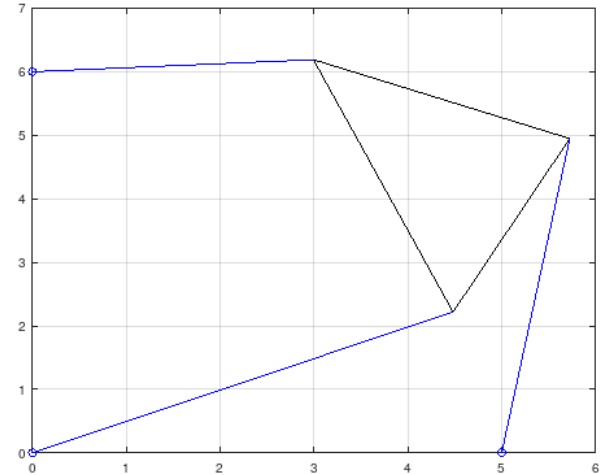
Kedua Selanjutnya kami mencari plot dari tiap θ tersebut menggunakan (fungsi 6) di kode kami.



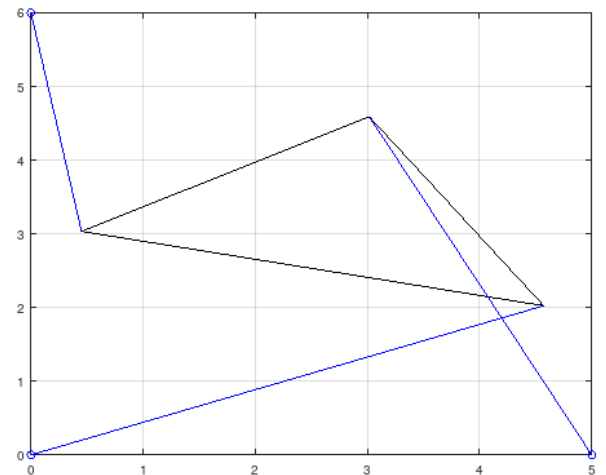
$\theta = -0.72085$ dan $(x, y) = (-1.3784, 4.8063)$



$\theta = -0.33101$ dan $(x, y) = (-0.91471, 4.9156)$



$\theta = 1.1437$ dan $(x, y) = (4.4818, 2.2167)$



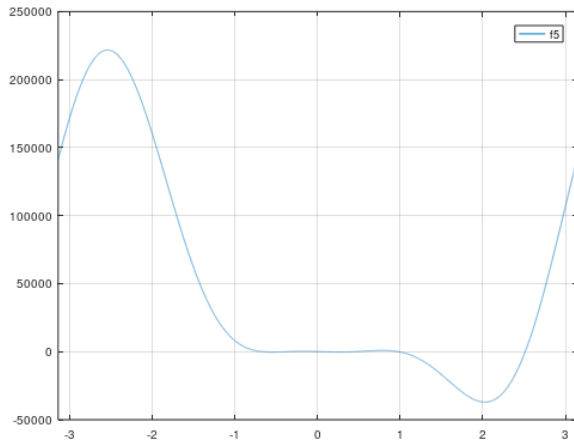
$\theta = 2.1159$ dan $(x, y) = (4.5718, 2.0244)$

Perlu diketahui bahwa kode plotting pose *Stewart Platform* yang kami buat di bagian 4 ini akan digunakan hingga bagian 6 dengan beberapa modi-

fikasi kecil.

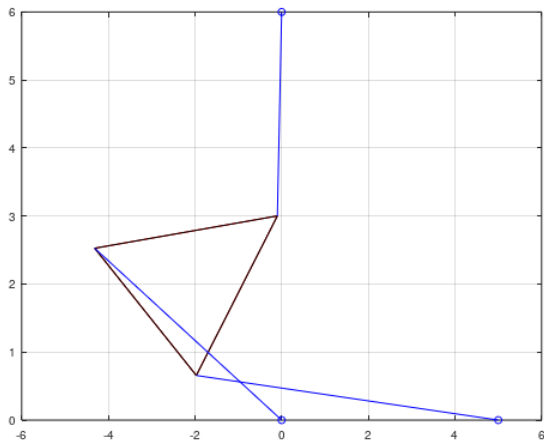
3.5 Part 5

Sama seperti di bagian 4, di bagian ini kami mencari grafik dari fungsi yang diimplementasikan ke kode kami. Ide kami adalah mencari terlebih dahulu plot dari fungsinya dan melihat kemungkinan nilai dari akar-akarnya.

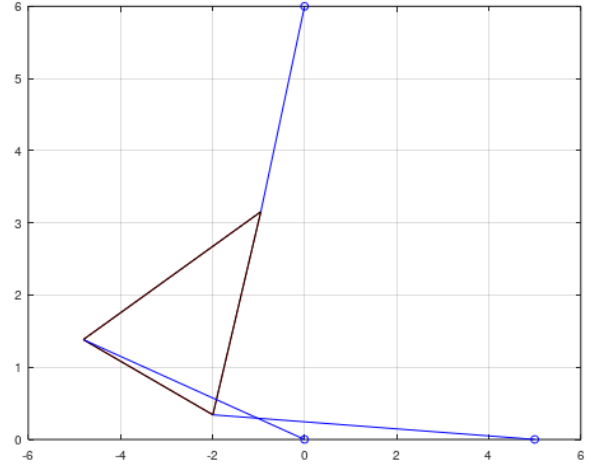


Plotting $f(\theta)$

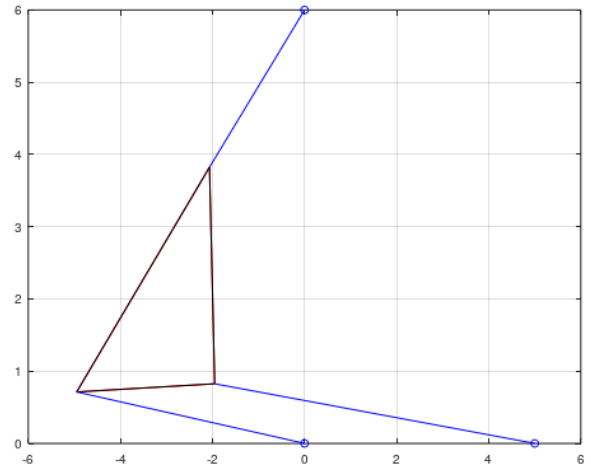
Lalu dengan metode *bisection* akan dicari nilai akar-akar dari fungsi tersebut yang selanjutnya akan kami gunakan sebagai parameter θ dari fungsi *Stewart Plot* yang telah kami buat sebelumnya.



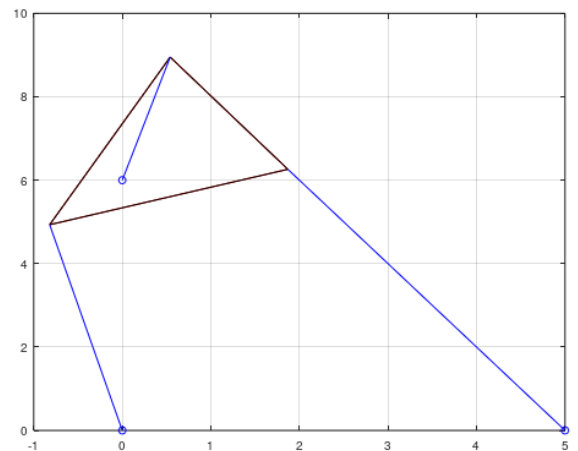
Batas $[-0.8, -0.6]$, $\theta = -0.67316$, $(x, y) = (-4.3148, 2.5264)$



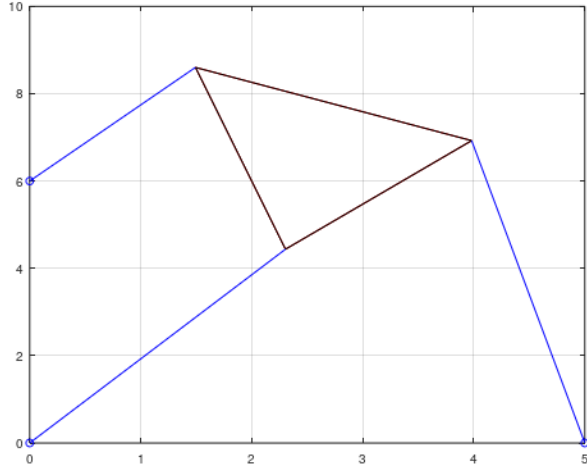
Batas $[-0.6, -0.2]$, $\theta = -0.35474$, $(x, y) = (-4.8049, 1.3831)$



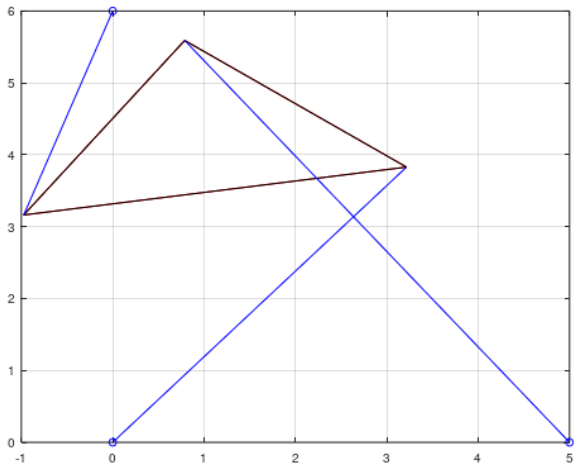
Batas $[-0.2, 0.2]$, $\theta = 0.037767$, $(x, y) = (-4.9490, 0.71215)$



Batas $[0.2, 0.6]$, $\theta = 0.45888$, $(x, y) = (-0.81980, 4.9323)$



Batas $[0.6, 1.2]$, $\theta = 0.97767$, $(x, y) = 2.3036, 4.4378$



Batas $[1.2, 2.6]$, $\theta = 2.5139$, $(x, y) = (3.2157, 3.8287)$

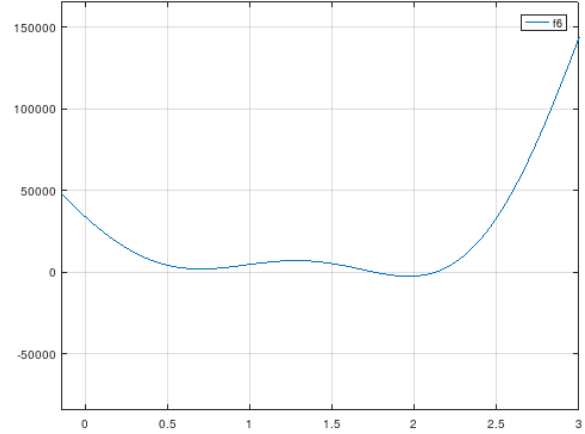
3.6 Part 6

Pada nomor 6, kami membuat algoritma untuk mencari nilai yang dapat merepresentasikan θ sehingga $f(\theta) = 0$ dan mampu menghasilkan tepat 2 buah pose *Stewart Platform* dengan memanfaatkan *Intermediate Value Theorem*.

Kami melakukan mengimplementasikan fungsi bersifat *brute force* (fungsi 10) terhadap nilai $p2$ untuk mencari berapa kali $f(\theta)$ bernilai 0 dengan menghitung berapa kali output dari algoritma kami berubah tanda (positif & negatif). Ketika fungsi tersebut berganti tanda sebanyak dua kali, maka fungsi $f(\theta)$ sudah bernilai 0 sebanyak dua kali. Dengan demikian, dua buah akar akan ditemukan dan

dapat dibuat tepat 2 buah pose.

Dari perhitungan fungsi tersebut, kami menemukan bahwa nilai $p2 = 9.2$ akan menghasilkan tepat 2 buah pose. Selanjutnya kami memasukkan nilai $p2 = 9.2$ sebagai $p2$ dari (fungsi 11) untuk membuat plotnya terhadap $[-\pi, \pi]$.



3.7 Part 7

Pengerjaan nomor 7 merupakan kelanjutan dari nomor 6, dimana kami mencoba mengimplementasikan metode *brute force* terhadap $p2$ untuk mendapatkan interval-interval yang memiliki jumlah akar yang genap. menggunakan (fungsi 13), kami mendapatkan hasil sebagai berikut :

1. interval $0 - 3.7150000000$: 0 akar
2. interval $3.7150000000 - 4.8650000000$: 2 akar
3. interval $4.8650000000 - 6.9700000000$: 4 akar
4. interval $6.9700000000 - 7.0250000000$: 6 akar
5. interval $7.0250000000 - 7.8500000000$: 4 akar
6. interval $7.8500000000 - 9.2650000000$: 2 akar
7. interval $9.2650000000 - \infty$: 0 akar

4 Conclusion

Dari aktivitas yang kami lakukan di atas, kami merasa senang karena telah berhasil memodelkan *Stewart Platform* dalam model dua dimensi. Kami menemukan banyak kesulitan sepanjang melakukan aktivitas ini, contohnya adalah dalam hal plotting dan mencari nilai-nilai θ dari suatu fungsi. Kami

menarik kesimpulan bahwa banyaknya pose dari su- akar-akar dari fungsi $f(\theta)$ tersebut.
atu *Stewart Platform* bergantung dari banyaknya

5 Reference

1. Bouck, Lucas, et al. (2016). *Project 1*. Retrieved from <http://mason.gmu.edu/~lbouck/project1.html>
2. Sauer, Timothy (2012). *Numerical Analysis*. Boston, MA: Pearson Education