

Pembuatan Aplikasi Simulator Robot Manipulator dengan Menggunakan Forward dan Inverse Kinematic di Unity

Kevin Nathanael Sugiharto

Program Studi Teknik Informatika

Fakultas Teknologi Industri, UK. Petra

Jl. Siwalankerto 121-131, Surabaya

Telp(031)2983455 Fax.(031)8417658

pencarisenja@gmail.com

Iwan Njoto Sandjaja

Program Studi Teknik Informatika

Fakultas Teknologi Industri, UK. Petra

Jl. Siwalankerto 121-131, Surabaya

Telp(031)2983455 Fax.(031)8417658

iwans@petra.ac.id

Handry Khoswanto

Program Studi Teknik Elektro

Fakultas Teknologi Industri, UK. Petra

Jl. Siwalankerto 121-131, Surabaya

Telp(031)2983455 Fax.(031)8417658

handry@petra.ac.id

ABSTRAK

Robot manipulator merupakan teknologi yang banyak digunakan di semua aplikasi kehidupan. Karena meluasnya penggunaan robot manipulator, maka diberikan pembelajaran robotika bagi mahasiswa peminatan robotik di program studi Teknik Elektro. Harga robot yang relatif mahal serta adanya perhitungan matematis menjadi hambatan.

Penelitian ini dibuat untuk menjawab masalah diatas. Aplikasi ini dibuat dengan *engine* Unity dengan bahasa pemrograman C#. Aplikasi ini mensimulasikan pergerakan robot baik secara *forward* maupun *inverse* kinematik dengan batasan DOF 2 dan 3. Aplikasi sejenis yang menjadi pembanding adalah RoboAnalyzer.

Berdasarkan hasil pengujian, program 'FIK' sudah menggambarkan *forward* maupun *inverse* kinematik, perhitungan rumus matematis, pergerakan simulator, serta *trace* yang dihasilkan. Untuk hasil *inverse* kinematik hanya menampilkan satu hasil, sedangkan pada program pembanding menampilkan kemungkinan 2-4 jawaban.

Kata Kunci: Simulator, Windows, Android, C#, Robot, DOF, *forward*, *inverse*, kinematik.

ABSTRACT

Robot manipulator is a technology that is widely used in all life applications. Because of the widespread use of robot manipulators, it is given robotics learning for students interested in robotics in Electrical Engineering courses. The price of the robot is relatively expensive and the existence of mathematical calculations become obstacles.

This thesis is made to answer the above problem. This application is made with Unity engine with C # programming language. This application simulates the movement of robots either forward or inverse kinematic with the limits of DOF 2 and 3. A similar application that is a comparison is RoboAnalyzer.

Based on the test results, the program 'FIK' has described the forward and inverse kinematik, mathematical formula calculation, simulator movement, and the resulting trace. For kinematic inverse results only show one result, whereas in the comparison program display the possibility of 2-4 answers.

Keywords: Simulator, Windows, C#, Robot, DOF, *forward*, *inverse*, kinematik.

1. PENDAHULUAN

Teknik Elektro merupakan salah satu program studi di Fakultas Teknologi Industri di Universitas Kristen Petra. Salah satu mata kuliah di program studi Teknik Elektro adalah pembelajaran

elektronik seperti *robot manipulator*. Pada pembelajaran *robot manipulator*, mahasiswa Teknik Elektro wajib mempelajari dasar dari pembelajaran tersebut, yakni materi *forward* dan *inverse* kinematik.

Pembelajaran *forward* dan *inverse* kinematik pada Teknik Elektro adalah pembelajaran dengan melakukan perhitungan. Visualisasi *robot* digunakan untuk membantu pembelajaran. Namun, pada kenyataan bahwa perhitungan terhadap pergerakan lengan *robot* yang cukup sulit dijelaskan karena pergerakan *robot* yang dihitung berdasarkan banyaknya persamaan matematik trigonometri serta tidak adanya visualisasi secara langsung respon gerakan lengan *robot* sehingga dalam proses belajar mengajar akan mengalami hambatan. Dari semua permasalahan terhadap visualisasi robot serta perhitungan matematis, maka diperlukan sebuah aplikasi simulasi *robot* yang dapat menunjukkan pergerakan *robot* saat melakukan pengajaran di kelas dengan mudah.

Program ini menunjukkan simulasi berupa pergerakan *robot* dari sumbu awal ke sumbu tertentu dengan ada animasi pergerakan lengan *robot*. Program ini dibuat dalam bentuk 3D untuk mempermudah visualisasi dalam kegiatan belajar mengajar.

Aplikasi sejenis yang sudah ada, yakni RoboAnalyzer. RoboAnalyzer merupakan aplikasi tiga dimensi yang mensimulasikan banyak jenis *robot* dan dikembangkan oleh Indian Insitute of Technology Dehli, India. Aplikasi ini bisa mensimulasikan DOF (*Degree of Freedom*) satu hingga tujuh. Untuk jenis lengan yang disediakan memiliki pilihan yakni secara *rotate* ataupun *planar*. [1]

Pada penelitian ini, akan dibuat simulator *robot* yang dinamakan "FIK" dengan menggunakan dua dan tiga lengan serta menggunakan jenis lengan *rotate* saja. Pengguna dapat melihat hasil pergerakan *robot* dengan hasil perhitungan matematis yang ditampilkan di program sehingga memudahkan pengguna dalam mempelajari *robot manipulator*.

2. TINJAUAN PUSTAKA

2.1 Unity

Unity merupakan aplikasi yang mengembangkan berbagai game dan aplikasi yang berbasis 2D atau 3D. Aplikasi unity dilengkapi dengan adanya *asset store* yang memungkinkan pembuat aplikasi dapat memiliki pilihan antara membuat atau tidak komponen *user interface* dari aplikasi yang akan dibuat. Unity dilengkapi dengan teknologi yang bagus sehingga memungkinkan programmer untuk membuat aplikasi dengan kualitas tinggi. Aplikasi Unity dapat mendukung teknologi terbaru seperti Windows, Mac, iOS, Android, VR, dll. Bahasa pemrograman yang digunakan di dalam Unity adalah C# dan javascript. [2]

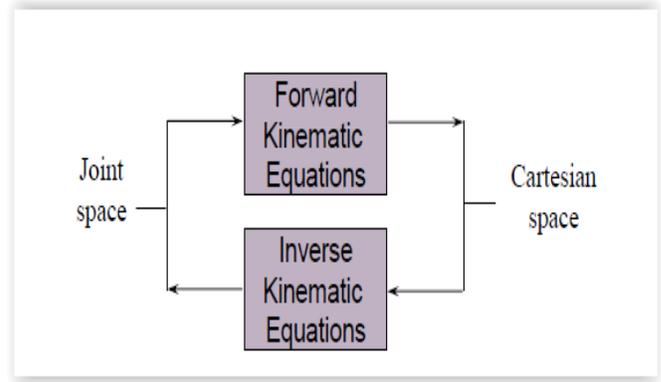
2.2 Robot Manipulator

Robot manipulator dapat diartikan sebagai sistem mekanik yang memiliki susunan *link* dan *joint* yang mampu menghasilkan gerakan terkontrol untuk memindah, mengangkat, dan memanipulasi benda kerja. Salah satu aplikasi terbesar dari manipulator robot yang sering digunakan adalah *welding arm robot* [3]. Robot ini dibagi menjadi dua jenis:

- Lengan dan Tubuh - lengan dan tubuh robot digunakan untuk memindahkan bagian-bagian posisi dan atau alat dalam amplop kerja. Mereka terbentuk dari tiga sendi dihubungkan dengan *link* besar.
- Wrist* - pergelangan tangan ini digunakan untuk mengarahkan bagian bagian atau peralatan di lokasi kerja. Ini terdiri dari dua atau tiga kompak sendi. Manipulator robot dibuat dari urutan kombinasi *link* dan sendi. *Link* yang menghubungkan para anggota kaku sendi, atau kapak. Sumbu adalah komponen bergerak dari robot yang menyebabkan gerakan relatif antara *link* yang berdekatan. Sendi mekanis digunakan untuk membangun manipulator terdiri dari lima jenis utama. Dua dari sendi yang *linear*, di mana gerakan relatif antara *link* yang berdekatan adalah non-rotasi, dan tiga jenis rotasi, di mana gerakan relatif melibatkan rotasi antara *link*. Bagian lengan-dan-tubuh manipulator didasarkan pada salah satu dari empat konfigurasi. Masing-masing anatomi memberikan amplop kerja yang berbeda dan cocok untuk aplikasi yang berbeda yaitu :
 - Gantry* - robot ini memiliki sendi linier dan dipasang *overhead*. Mereka juga disebut robot *Cartesian* dan bujursangkar.
 - Cylindrical* – Dinamakan untuk bentuk tempat kerjanya, robot anatomi silinder yang dibuat dari sendi *linear* yang terhubung ke basis bersama *rotary*.
 - Polar*- Dasar bersama robot kutub memungkinkan untuk memutar dan sendi adalah kombinasi jenis putar dan linier. Ruang kerja yang diciptakan oleh konfigurasi ini bulat
 - Joined-Arm* - Ini adalah konfigurasi yang paling populer robot industri. Lengan menghubungkan dengan bersama memutar, dan *link* di dalamnya dihubungkan dengan sendi putar. Hal ini juga disebut robot diartikulasikan. [4]

2.3 Forward dan Inverse Kinematic

Kinematika adalah ilmu tentang gerak tanpa mengindahkan penyebabnya. Di dalam kinematika, terdiri atas 2 cabang, yakni *forward* & *inverse*. Dua cabang tersebut sangat berhubungan karena adanya kalkulasi yang saling berelasi diantara posisi sendi serta nilai mutlak *frame* koordinat *Cartesian*. [5] Hubungan antara keduanya dapat digambarkan di Gambar 1.



Gambar 1. Hubungan *forward* dan *inverse* kinematik

2.3.1 Forward kinematic

Menginputkan suatu konfigurasi sendi (pose) yang menghasilkan gerakan dari derajat kebebasan. Berikut persamaan matematik *forward* kinematik 2 DOF di persamaan 2.1, 2.2, dan 2.3.

$$P_x = \text{Cos}(\theta_1) * (L_2 * \text{Cos}(\theta_2)) \quad (2.1)$$

$$P_y = \text{Sin}(\theta_1) * (L_2 * \text{Cos}(\theta_2)) \quad (2.2)$$

$$P_z = L_1 + L_2 * \text{Sin}(\theta_2) \quad (2.3)$$

(P_x, P_y, P_z) => Posisi xyz
 (θ_1, θ_2) => Sudut 1,2
 (L_1, L_2) => Panjang lengan 1,2

Berikut persamaan matematik *forward* kinematik 3 DOF di persamaan 2.4, 2.5, dan 2.6.

$$P_x = \text{Cos}(\theta_1) * (L_2 * \text{Cos}(\theta_2) + L_3 * \text{Cos}(\theta_2 + \theta_3)) \quad (2.4)$$

$$P_y = \text{Sin}(\theta_1) * (L_2 * \text{Cos}(\theta_2) + L_3 * \text{Cos}(\theta_2 + \theta_3)) \quad (2.5)$$

$$P_z = L_1 + L_2 * \text{Sin}(\theta_2) + L_3 * \text{Sin}(\theta_2 + \theta_3) \quad (2.6)$$

(P_x, P_y, P_z) => Posisi xyz
 ($\theta_1, \theta_2, \theta_3$) => Sudut 1,2,3
 (L_1, L_2, L_3) => Panjang lengan 1,2,3

2.3.2 Inverse kinematic

Menginputkan titik tujuan dan dapat menghasilkan sudut untuk mencapai tujuan tersebut. Berikut persamaan matematik *inverse* kinematik 2 DOF di persamaan 2.7 dan 2.8.

$$\theta_1 = \text{Atan2}(P_y, P_x) \quad (2.7)$$

$$\theta_2 = \text{Asin} \left(\frac{P_z - L_1}{L_2} \right) \quad (2.8)$$

(P_x, P_y, P_z) => Posisi xyz
 (θ_1, θ_2) => Sudut 1,2
 (L_1, L_2) => Panjang lengan 1,2

Berikut persamaan matematik *inverse* kinematik 3 DOF di persamaan 2.9, 2.10, 2.11, 2.12, dan 2.13.

$$r = \pm \sqrt{P_x^2 + P_y^2} \quad (2.9)$$

$$D = \pm \sqrt{(P_z - l_1)^2 - r^2} \quad (2.10)$$

$$\theta_1 = \text{Atan2}(P_y, P_x) \quad (2.11)$$

$$\theta_3 = \text{Acos} \left(\frac{D^2 - L_2^2 - L_3^2}{2L_2L_3} \right) \quad (2.12)$$

$$\theta_2 = \text{Atan2}(r, P_z - L_1) - \text{Atan2}(L_2 + L_3 * \text{Cos}\theta_3, L_3 * \text{Sin}\theta_3) \quad (2.13)$$

(P_x, P_y, P_z) => Posisi xyz
 $(\theta_1, \theta_2, \theta_3)$ => Sudut 1,2,3
 (L_1, L_2, L_3) => Panjang lengan 1,2,3

2.3.3 Derajat kebebasan

Derajat kebebasan atau yang dikenal dengan *degree of freedom (DOF)* merupakan bagian dari mesin serta memiliki sendi mekanisme yang berfungsi mengatur kerja mesin. Setiap anggota mekanisme dapat bergerak ke arah tertentu atau memutar di sekitar sumbu tertentu dan tidak diperbolehkan bergerak atau berputar ke arah lain. Derajat kebebasan menentukan kemungkinan pergerakan mekanismenya.[6]

2.4 C#

C# (dibaca C *sharp*) adalah bahasa pemrograman yang didesain untuk berbagai macam aplikasi yang dijalankan pada .NET Framework. C# merupakan bahasa yang sederhana, luar biasa, aman, dan berorientasi pada obyek. Macam-macam inovasi dalam C# adalah memungkinkan pengembangan aplikasi yang cepat sementara tetap mempertahankan ekspresi dan keanggunan gaya bahasa C.[7]

2.5 RoboAnalyzer

RoboAnalyzer merupakan model tiga dimensi berbasis aplikasi yang dapat digunakan untuk kegiatan mengajar dan belajar konsep robotik. Itu adalah produk berkembang yang dikembangkan di Mechatronics Lab, Departement of Mechanical Engineering di IIT Delhi, India.

Virtual Robot Module, bagian dari RoboAnalyzer sudah mengembangkan aplikasi dimana memiliki *joint* dan gerakan kartesian. Itu dibuat dari *COM server*, menggunakan salah satu yang mengintegrasikan VRM dan Matlab, Ms. Excel, dan aplikasi lainnya yang memiliki tampilan *COM*. [1] Tampilan simulator RoboAnalyzer bisa dilihat di Gambar 2.

Gambar 2. Tampilan awal simulator RoboAnalyzer

2.6 Simulator

Simulator adalah program yang berfungsi untuk mensimulasikan suatu peralatan namun kerjanya agak lambat daripada keadaan yang sebenarnya. [8]

3. ANALISIS DAN DESAIN

Program simulator memiliki 3 pilihan DOF. Secara garis besar tahap dalam menggunakan program dibagi dalam 5 Tahap.

Tahap pertama adalah memilih DOF yang diinginkan. Untuk DOF yang disediakan hanya antara 2 atau 3 DOF.

Tahap kedua adalah *input* panjang lengan sesuai DOF yang diinginkan. Panjang lengan dalam Simulator menggunakan satuan *centimeter*.

Tahap ketiga akan diminta untuk *input* besar derajat tiap DOF serta berapakah hasil *input* berjalan (*counter of steps*).

Tahap keempat program akan melakukan penghitungan terhadap derajat yang *input* serta melakukan pergerakan terhadap robot serta menghasilkan lintasan (*trajectory*), dan menampilkan hasil perhitungan.

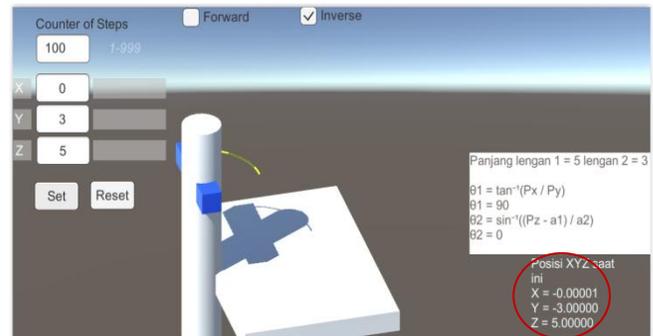
Tahap kelima pergerakan simulator berhenti setelah robot mencapai sudut tujuannya.

4. HASIL DAN PEMBAHASAN

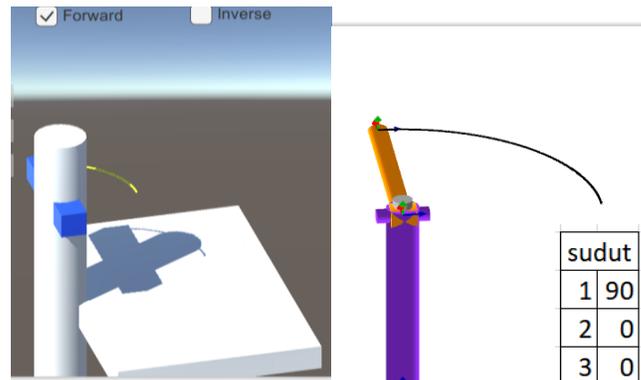
Program yang dibuat dapat menggambarkan *forward* dan *inverse* kinematik. Berikut merupakan contoh hasil pengujian yang telah dibuat beserta tabel perhitungan untuk *forward* kinematik 2 DOF pada Tabel 1 dengan excel dengan Gambar 3 pada bagian lingkaran merah.

Tabel 1. Hasil *forward* kinematik 2 DOF

derajat input		X	Y	Z
1	2			
90	0	3.00	0.00	5.00



Gambar 3. Hasil *forward* 2 DOF di FIK



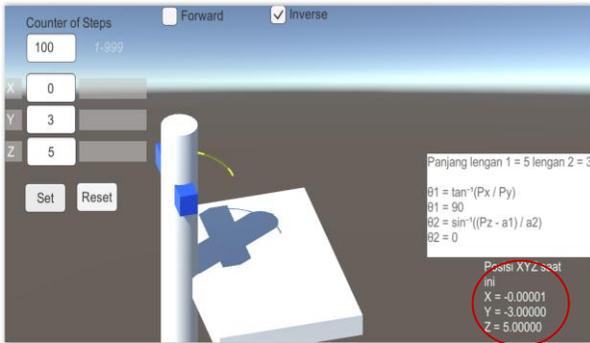
Gambar 4. Hasil pengujian terhadap gerakan pada program FIK dan RoboAnalyzer 2 DOF

Pada Gambar 4 menunjukkan bahwa hasil pergerakan serta *trace* (garis) yang dihasilkan sesuai.

Berikut merupakan contoh hasil pengujian yang telah dibuat beserta Tabel perhitungan untuk *inverse* kinematik 2 DOF pada Tabel 2 dengan excel dengan Gambar 5 pada bagian lingkaran merah.

Tabel 2. Hasil *inverse* kinematik 2 DOF

derajat input		X	Y	Z
1	2			
90	0	3.00	0.00	5.00

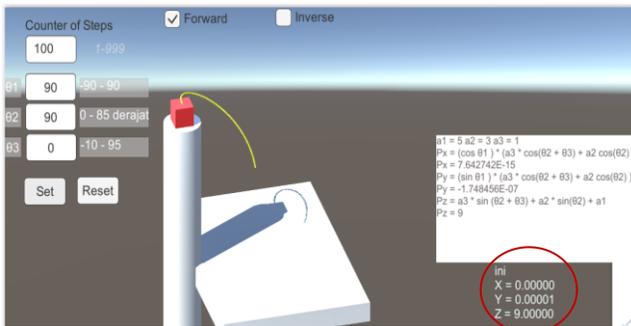


Gambar 5. Hasil *inverse* 2 DOF

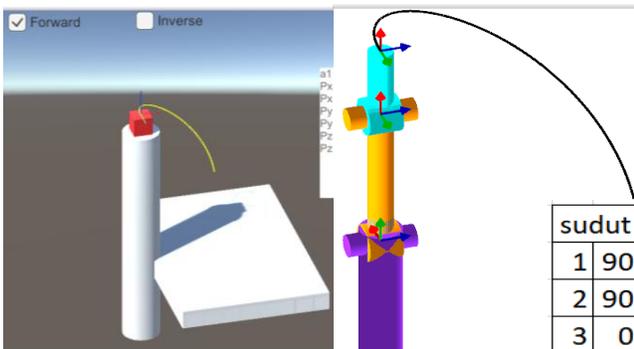
Berikut merupakan contoh hasil pengujian yang telah dibuat beserta tabel perhitungan untuk *forward* kinematik 3 DOF pada Tabel 2 dengan excel dengan Gambar 6 pada bagian lingkaran merah.

Tabel 4.3 hasil *forward* kinematik 3 DOF

derajat input		X	Y	Z
1	2			
90	90	0.00	0.00	9.00



Gambar 6. Hasil *forward* 3 DOF



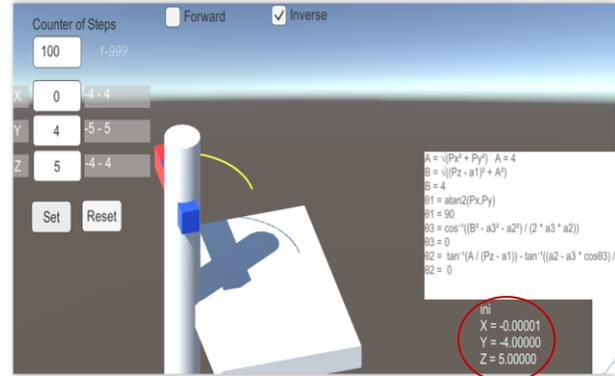
Gambar 7. Hasil pengujian terhadap gerakan pada program FIK dan RoboAnalyzer 3 DOF

Pada Gambar 7 menunjukkan bahwa hasil pergerakan serta *trace* (garis) yang dihasilkan sesuai.

Berikut merupakan contoh hasil pengujian yang telah dibuat beserta tabel perhitungan untuk *inverse* kinematik 3 DOF pada Tabel 4 dengan excel dengan Gambar 8 pada bagian lingkaran merah.

Tabel 4. hasil *inverse* kinematik 3 DOF

derajat input		X	Y	Z
1	2			
90	0	3.00	0.000	5.000



Gambar 8. Hasil *inverse* 3 DOF

5. SECTIONS

5.1 Kesimpulan

1. Berdasarkan hasil pengujian, program 'FIK' sudah menggambarkan *forward* maupun *inverse* kinematik, perhitungan rumus matematis, pergerakan simulator, serta *trace* yang dihasilkan.
2. Semua hasil *inverse* kinematik, baik 2 DOF maupun 3 DOF hanya mampu menampilkan satu jawaban, namun pada RoboAnalyzer dapat menghasilkan dua hingga empat kemungkinan hasil perhitungan.

5.2 Saran

1. Pengembangan untuk dijalankan dengan Robot DOBOT MAGICIAN.
 2. Pengembangan *inverse* kinematik yang dapat menemukan beberapa kesimpulan atau hasil yang berbeda.
 3. Pengembangan terhadap tampilan robot seperti dengan bentuk robot yang sebenarnya.
- Pengembangan *rotate* sudut tampilan simulator, baik di aplikasi Windows maupun Android.

6. REFERENCES

- [1] Roboanalyzer. n.d. *RoboAnalyzer*. Retrieved March 12, 2017 from <http://www.rob analyzer.com>
- [2] Unity. 2016. Unity. Retrieved November 10, 2016 from <https://www.unity3d.com>
- [3] Cucu. 2012. Robot manipulator. Retrieved November 12, 2016 from http://www.robotic-explorer-bandung.com/artman2/publish/in-house-training/Robot_Manipulator.shtml
- [4] RobotWorx. 2016. What is robot manipulator. Retrieved November 10, 2016 from

<https://www.robots.com/faq/show/what-is-a-robot-manipulator>

[5] Kamus Besar Bahasa Indonesia. n.d. Retrieved Juli 26, 2017 from <http://kbbi.web.id/kinematika>

[6] Lafmenjani, Hossein Sadegh., Zarabadipour, Hassan. 2014. *Modelling, simulation and position control of 3dof articulated manipulator*. Retrieved June 28, 2017 from https://www.researchgate.net/publication/281237691_Model

[ing_Simulation_and_Position_Control_of_3DOF_Articulated_Manipulator](#)

[7] Naveenagrawal. 2009. *Degree of Freedoms*. Retrieved Februari 26, 2017 from <http://www.brighthubengineering.com/machine-design/6634-kinematics-design-of-mechanisms-degrees-of-freedom/>

[8] KBBI. n.d. Retrieved Juli 26, 2017 from <http://kbbi.web.id/simulator>