ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika

ISSN(p): 2338-8323 | ISSN(e): 2459-9638 | Vol. 10 | No. 1 | Halaman 46 – 60
DOI : http://dx.doi.org/10.26760/elkomika.v10i1.46 Januari 2022

Rule-Based Learning untuk Robot Humanoid T-FLoW Belajar Berjalan

FAIZ ULURRASYADI[1], ALIRIDHO BARAKBAH[2], RADEN SANGGAR DEWANTO[3], DADET PRAMADIHANTO[4]

[1]Departemen Teknik Elektro, Politeknik Elektronika Negeri Surabaya
[2,4]Departemen Teknik Informasi dan Komputer, Politeknik Elektronika Negeri Surabaya
[3]Departemen Teknik Mekanika dan Energi, Politeknik Elektronika Negeri Surabaya
Email: faizulurrasyadi04@gmail.com

Received 12 Juli 2021 **|** *Revised* 2 Agustus 2021 **|** *Accepted* 9 Agutus 2021

ABSTRAK

Riset tentang penggunaan learning dalam motion robot humanoid telah banyak dilakukan di seluruh dunia. Salah satunya adalah melakukan learning gerakan berjalan pada robot. Penelitian ini akan menjelaskan suatu metode learning "Rule Based" yang simple dan cepat dalam menemukan solusi gerakan berjalan yang stabil pada robot humanoid T-FLoW . Robot diibaratkan seperti anak kecil yang belajar berjalan, dia tahu cara berjalan, akan tetapi tidak tahu seberapa besar dia harus menggerakkan sendi-sendi atau joint di kakinya agar dapat berjalan seimbang. Oleh karena itu sistem learning akan menemukan nilai point-point trayektori yang cocok untuk berjalan dengan stabil. Dengan menggunakan software simulasi CoppeliaSim, kami menerapkan metode tersebut. Hasilnya, robot humanoid T-FLoW dapat berjalan dengan stabil sejauh 170 langkah hanya dengan melakukan learning sebanyak 400 episode.

***Kata kunci**: Robot humanoid T-FLoW, Rule-Based Learning, Learning, CoppeliaSim, Trayektori.*

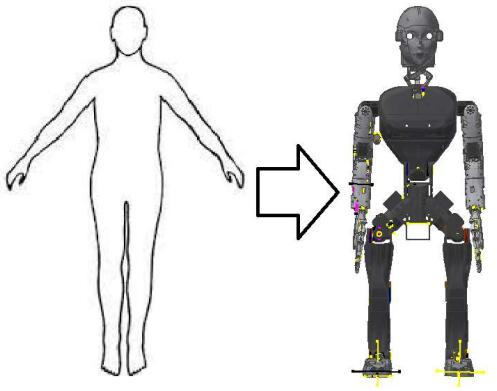
ABSTRACT

Research about the use of learning in motion of humanoid robot has been done in many countries. One of them was done by learning a stable walking gait in humanoid robot. This research will explain a fast and simple Rule Based learning method to find the solution of stable walking motion for T-FLoW humanoid robot. A robot was assumed like a child trying to walk, he knows how to walk, but doesn't know how much he has to move his legged joints to get a stable walking. So, our learning system will find those trajectory point values that is suitable to walk stably. By using CoppeliaSim software, we implement our method. The result is, T-FLoW humanoid robot was able to walk stably for about 170 steps with only 400 episodes of learning.

***Keywords**: T-FLoW humanoid robot, Rule-Based Learning, Learning, CoppeliaSim, Trajectory.*

1. PENDAHULUAN

Pandemi virus pada tahun 2020 tidak menghentikan laju dari para peneliti untuk terus melakukan kemajuan dalam berbagai bidang yang ditekuninya, termasuk bidang robotika. Jika selama ini penelitian dilakukan di laboratorium, maka pada masa kini, simulasi menjadi sebuah pilihan untuk terus melanjutkan eksperimen. Hal serupa juga dialami oleh *RoISC*, sebuah grup riset dari PENS yang terus berkarya walaupun dengan keterbatasan-keterbatasan yang telah ditetapkan untuk mengurangi dampak pandemi. Dari beberapa penelitian yang telah dilakukan, salah satunya adalah tentang robot humanoid. Robot humanoid merupakan robot yang dibuat agar memiliki bentuk dan gerakan yang menyerupai manusia seperti yang ditunjukkan pada Gambar 1. Gerakan dari robot humanoid sendiri terbagi menjadi dua jenis, yaitu gerakan statik dan dinamik. Gerakan secara statik ini hanya melibatkan analisis kinematik yang mengabaikan massa dan gravitasi, sedangkan untuk gerakan dinamik melibatkan analisis kinematik dan dinamik. Tidak semua hal dalam robot humanoid dapat dianalisis atau dimodelkan dengan mudah dengan persamaan-persamaan kinematik dan dinamik. Di sini, peran dari *Artificial Intelligence* atau *AI* dibutuhkan untuk mengkompensasi adanya *missing* parameter akibat dari pemodelan kinematik dan dinamik.



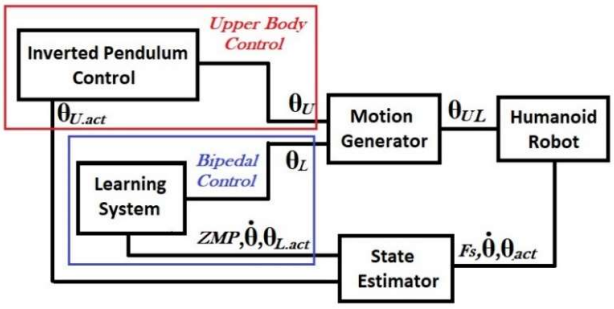
Gambar 1. Desain Robot Humanoid yang Diserupakan Manusia

Banyak sekali beberapa penerapan *AI* dalam robot humanoid mulai dari *low level control* hingga *high level control*. **(Missura, dkk, 2015)** telah berhasil melakukan *learning* untuk robot humanoid yang dapat menyeimbangkan kembali setelah diberi gangguan dengan besaran tertentu. Sistemnya dibedakan menjadi dua yaitu yang pertama adalah kontrol analisis yang di dalamnya terdapat bentuk *closed-form* dari persamaan *Linear Inverted Pendulum Model (LIPM)* serta kontrol untuk *Centre of Mass (CoM)* dan yang kedua adalah kontrol yang berbasis *learning*. **(Zhang, dkk, 2015)** telah berhasil melakukan *learning* dalam simulasi untuk robot humanoid yang berjalan di medan yang tidak teratur. Dengan menggunakan *reinforcement learning* yang digabungkan dengan *RBF Network* dapat meningkatkan performansi berjalan dari robot humanoid secara *realtime*. **(Silva, dkk, 2016)** menggabungkan kombinasi dua layer yaitu *gait generation* yang tradisional dan *reinforcement learning* untuk melakukan *learning* posisi sudut dari tubuh robot humanoid yang berjalan di bidang miring. **(Kormushev, dkk, 2011)** menggunakan pendekatan *learning* untuk meminimalisasi konsumsi energi listrik ketika berjalan dengan bipedal robot yang memiliki gerakan *joint* pasif. Pengurangan konsumsi energi dilakukan dengan *learning* berbagai macam ketinggian *centre of mass* dari robot dan dengan metode *reinforcement learning* yang secara dinamik dapat mengubah parameter-parameter kebijakan selama proses *learning* berlangsung. Hasilnya, metode tersebut dapat mengurangi konsumsi energi sebanyak 18% dalam bidang sagital ketika robot berjalan.

Walaupun telah banyak dilakukan, kebanyakan dari robot humanoid yang digunakan memiliki struktur mekanik atau kinematik yang mudah untuk seimbang. Pada penelitian ini, robot humanoid yang digunakan memiliki bentuk struktur yang lebih sulit untuk diseimbangkan, dikarenakan jarak antara kaki kanan dan kiri lebar sehingga usaha untuk memasukkan *centre of mass* ke *support polygon* pada satu kaki yang menapak di tanah cukup sulit dan juga bentuk telapak kaki yang kecil daripada kebanyakan robot humanoid lainnya yang menyebabkan robot mudah untuk jatuh dengan gangguan kecil saja. Sistem *learning* akan berusaha untuk melakukan pemetaan terhadap *action* yang sesuai untuk setiap *state* sehingga robot humanoid dapat berjalan dengan tidak jatuh. Pada bagian selanjutnya akan dijelaskan metode atau sistem yang digunakan pada penelitian ini.

2. METODE PENELITIAN

Tujuan dari penelitian ini adalah untuk mendapatkan trayektori berjalan robot humanoid T-FLoW yang stabil dengan cara melakukan *learning*. Tidak semua *joint* pada robot akan dikontrol dari sistem *learning*, akan tetapi dibagi menjadi dua bagian, yaitu kontrol tubuh bagian atas dan kontrol tubuh bagian bawah. Untuk kontrol bagian atas memegang kendali pada tubuh robot dengan metode 3D *Inverted* Pendulum **(Elhasairi, dkk, 2015)**, sedangkan pada bagian bawah memegang kendali atas *bipedal robot* (mulai dari *hip* hingga telapak kaki) dengan diserahkan tambahan adanya sistem *learning*. Blok diagram sistem secara keseluruhan dapat dilihat pada Gambar 2.



Gambar 2. Blok Diagram Sistem

Jadi secara garis besarnya, dua sistem kontrol (berkotak merah dan biru) mengeluarkan sinyal berupa posisi, di mana:

$$\theta_U = \{\theta_{Tx} \quad \theta_{Ty} \quad \theta_{Tz}\} \tag{1}$$

$$\theta_L = \{\theta_{Sx} \quad \theta_{Sy} \quad \theta_{Sz}\} \tag{2}$$

θ_{Tx}, θ_{Ty}, θ_{Tz} adalah masing-masing posisi dalam sumbu x *(roll)*, y *(pitch)* dan z *(yaw)* yang dibutuhkan untuk menggerakkan tubuh agar robot tetap dalam keadaan stabil dengan nilai gangguan yang masih dapat diredam oleh kontrol 3D *inverted* pendulum. Sedangkan untuk θ_{Sx}, θ_{Sy}, θ_{Sz} adalah posisi dari *end of effector* kaki robot (x, y dan z) yang akan melakukan gerakan melangkah *(swing)*. Semua posisi tersebut menjadi masukan untuk blok *motion generator*, di mana di dalamnya terdapat persamaan kinematika terbalik untuk robot humanoid T-FLoW. Sinyal yang keluar dari *motion generator* juga masih sama dengan sebelumnya, yaitu posisi. Hanya saja di sini posisinya bukan dalam bentuk sumbu x, y dan z, akan tetapi dalam bentuk sudut-sudut *joint* yang diperlukan untuk bergerak. Selanjutnya pada blok *state*

estimator yang isinya adalah *hardware* berupa sensor-sensor, mengambil data berupa *Fs* (*Force* Sensor pada kaki), $\dot{\theta}$ (percepatan) dan $\theta_{L.act}$ (posisi aktual) pada robot untuk kemudian diolah dan dijadikan *feedback* untuk sistem *learning* dan juga *inverted* pendulumnya.

2.1 Kinematika Robot

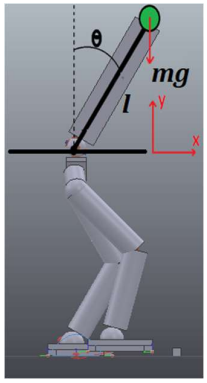
Robot humanoid T-FLoW memiliki sejumlah *Degree of Freedom* sebanyak 32 yang semuanya itu diaktuasikan oleh motor servo, dengan rinciannya adalah 6 *Degrees of Freedom* terdapat di masing-masing kaki kanan dan kiri, 3 di bagian *hip*, 3 di bagian leher dan sisanya ada pada kedua tangan. Robot humanoid T-FLoW memiliki tinggi 115 cm dan berat sebesar 9 kg.

Gambar 3. Desain Robot Humanoid T-FLoW

Gambar 3 menunjukkan desain mekanik dari robot humanoid T-FLoW yang pada saat ini terealisasi dengan menggunakan bahan EPACF filamen karbon, hasil dari 3D *printing*. Terdapat beberapa sensor yang terpasang di antaranya, *accelerometer* yang ada di bagian *hip* robot, *force* sensor (sensor tekanan) sebanyak 4 buah yang terpasang di masing-masing telapak kaki dan juga di setiap motor servo yang sudah *built-in* dengan berbagai macam program, sehingga dapat diambil beberapa informasi selayaknya sensor, salah satunya posisi dari servo, torsi dan percepatannya.

2.2 Kontrol *Inverted* Pendulum

Untuk menyederhanakan pemodelan robot humanoid, para peneliti biasanya menggunakan konsep dari *inverted* pendulum. *Inverted* pendulum adalah pendulum yang memiliki pusat massa di atas titik pivotnya seperti pada Gambar 4. Sedangkan pada pendulum yang normal selalu stabil saat digantung, *inverted* pendulum secara inheren tidak stabil dan membutuhkan kontrol untuk membuatnya stabil, yaitu dengan menerapkan torsi pada titik pivot.



Gambar 4. *Inverted* Pendulum dalam Robot Humanoid

Persamaan gerak dari *inverted* pendulum ada banyak sekali bergantung pada batasan-batasan yang ada pada sistem atau *plant*. Dalam kasus robot humanoid ini, titik pivot berada di bagian *hip*, karena *inverted* pendulum ini hanya mengendalikan tubuh bagian atas robot. Berdasarkan ilustrasi pada Gambar 4, persamaan *inverted* pendulum dapat dituliskan dalam bentuk torsi.

$$\tau = mglsin\theta \tag{3}$$

Di mana *m* adalah massa, *g* adalah kecepatan gravitasi, *l*panjang dari titik pivot ke titik massa dan θ adalah sudut yang terbentuk. Karena torsi menyebabkan suatu benda yang bermassa bergerak melingkar yang disebabkan karena adanya inersia benda dengan percepatan tertentu, sehingga Persamaan (3) juga dapat ditulis sebagai berikut.

$$\tau = I\ddot{\theta} \tag{4}$$

Sehingga

$$I\ddot{\theta} = mglsin\theta \tag{5}$$

Iyang merupakan inersia memiliki bermacam-macam rumus bergantung dari titik putarannya. Untuk robot humanoid yang memiliki sistem dengan titik putarannya di ujung (seperti lengan), sehingga persamaan dari inersia dituliskan sebagai berikut.

$$I = \frac{1}{3}ml^2 \tag{6}$$

Dari Persamaan (6), dapat ditulis kembali untuk Persamaan (5) menjadi sebagai berikut.

$$\frac{1}{3}ml^2\ddot{\theta} = mglsin\theta \tag{7}$$

$$\ddot{\theta} = \frac{3gsin\theta}{l} \tag{8}$$

Dari Persamaan (8) didapatkan percepatan yang dibutuhkan untuk melakukan perlawanan terhadap percepatan yang menyebabkan robot jatuh. Akan tetapi dari Persamaan (8) tersebut, hanya akan menyebabkan robot mempertahankan posisinya di sudut tertentu. Percepatan yang dihasilkan dari Persamaan (8) tidak akan membuat tubuh dari robot humanoid kembali tegak lurus, sehingga dibutuhkan percepatan yang lebih besar untuk melakukan hal tersebut. Sehingga berlaku persamaan berikut.

$$\ddot{\theta} > \frac{3gsin\theta}{l} \tag{9}$$

Dari percepatan yang dihasilkan tersebut, kemudian dijadikan input untuk kontroler *PID*. Di mana nantinya referensi posisinya adalah tegak lurus atau 0^0.

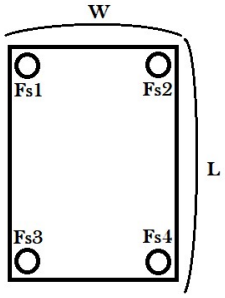
2.3 Sistem *Learning*

Sistem *learning* di sini akan bertanggung jawab dalam membentuk trayektori berjalan robot T-FLoW yang stabil. Sistem *learning* hanya akan memberitahu lokasi koordinat di mana *end of effector* kaki harus bergerak, sedangkan untuk kontrol gerakan kakinya atau *low level* kontrolnya diserahkan pada *software* simulasi *CoppeliaSim*, seperti persamaan kinematika terbalik yang mengubah posisi menjadi sudut telah terkalkulasi di awal beserta kontrol *PID*

untuk mempertahankan *link* dalam posisi sudut tertentu. Dua metode *learning* digunakan dalam penelitian ini, yaitu *Rule Based Learning* **(Darsih, 2015)** dan *Reinforcement Learning (Q-Learning*) **(Silva, dkk, 2017)***.* Penggunaan dari *Reinforcement Learning* di sini adalah sebagai pembanding dalam hal metode yang diajukan (*Rule-Based Learning)* dan juga hasilnya.

2.3.1 *State*

Dalam pembuatan sistem *learning*, *state* merupakan salah satu komponen yang paling penting dalam menentukan berhasil atau tidaknya suatu sistem *learning. State* merupakan kondisi yang dapat menjelaskan keadaan suatu *agent learning* atau dalam hal ini adalah robot humanoid T-FLoW. Banyak *paper* dan jurnal selalu menghindari pembentukan *state* yang banyak, dan dampak dari hal ini adalah *agent* menjadi tidak dapat memetakan setiap *state* dengan baik. **(Hwang, dkk, 2015)** melakukan pengelompokan *state* berupa *ZMP (Zero Moment Point)* ke dalam beberapa wilayah di kaki. Hal ini dapat menjadikan dalam satu wilayah tertentu terdapat beberapa point *ZMP* yang tentu saja dapat memiliki parameter keseimbangan yang berbeda-beda, sehingga akan menghasilkan *reward* yang berbeda pula untuk sistem *learning.*



Gambar 5. Posisi Sensor Tekanan untuk *ZMP*

Terinspirasi dari **(Hwang, dkk, 2016)**, representasi *state* dalam penelitian ini akan diwakilkan oleh lokasi dari *ZMP* robot humanoid. Dengan cara memasang sensor tekanan di ujung-ujung telapak kaki robot humanoid seperti pada Gambar 5 lokasi *ZMP* dapat diketahui dengan Persamaan (10) dan Persamaan (11).

$$x = \frac{W(fs_2 + fs_4) - (fs_1 + fs_3)}{2(\sum_1^4 fs_i)} \tag{10}$$

$$y = \frac{L(fs_1 + fs_2) - (fs_3 + fs_4)}{2(\sum_1^4 fs_i)} \tag{11}$$

Nilai murni x dan y hasil dari persamaan di atas akan menjadi komponen *state* dalam *learning*. Salah satu alasan memakai *ZMP* adalah karena *ZMP* merupakan salah satu kriteria standar yang digunakan untuk mengukur gaya berjalan robot humanoid **(Li, dkk, 2012)**.

2.3.2 *Action*

Representasi dari *action* juga tidak kalah pentingnya dari *state*, berhubung transisi di setiap pergantian *state* selalu diawali terlebih dahulu dengan pengambilan keputusan terkait *action*. Tentang pengambilan keputusan ini, terdapat dua gerakan pada umumnya, yaitu eksplorasi dan eksploitasi. Eksplorasi bersifat acak, artinya *agent* atau robot humanoid akan mengambil *action* terhadap *state* tertentu tidak berdasarkan apa-apa, murni secara acak. Hal ini dilakukan

untuk mencari pasangan *state-action* yang mungkin saja mempunyai solusi lebih baik. Sedangkan untuk eksploitasi adalah gerakan memilih *action* dengan melihat nilai pasangan *state-action* yang paling besar di antara pasangan *state-action* yang tersedia.

Proses *learning* akan menjadi lebih efisien jika representasi dari *action* tidak terlalu banyak dan juga tepat. Untuk menyederhanakan operasi dari 12 motor pada bagian bipedal, *action-spaces* direpresentasikan dengan posisi dari *end of effector* pada dua kaki robot dalam x, y dan z. Untuk *range* nilai dari masing x, y dan z berbeda-beda.

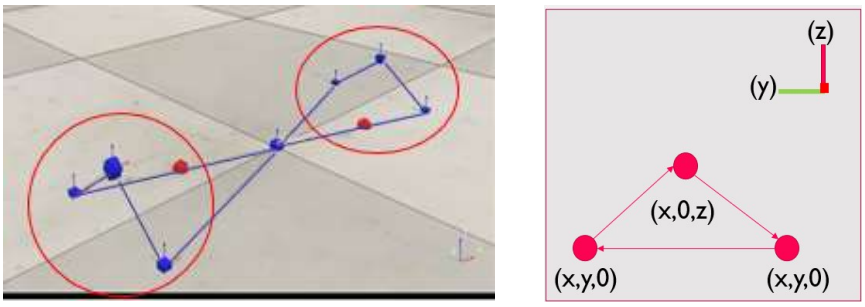
Tabel 1. Range Komponen *Action*

Sumbu	Nilai (m)
X	0.2 - 0.35
Y	0.05 - 0.2
Z	0.03 - 0.1

Nilai-nilai pada Tabel 1 didapatkan dari beberapa pengujian yang menjelaskan keterbatasan robot dalam melakukan gerakan. Jika ditotal maka terdapat 448 variasi *action* dalam setiap *state*.

2.3.3 Metode *Rule-Based Learning*

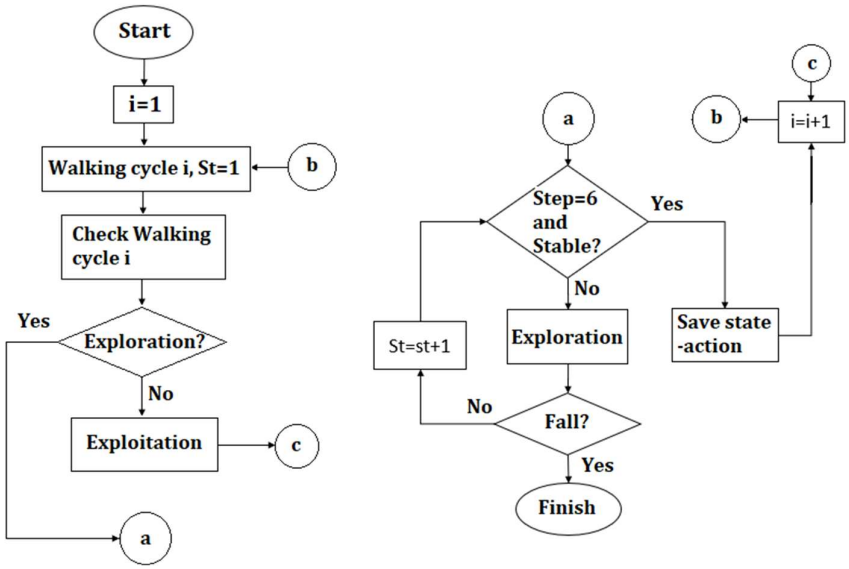
Ide yang mendasari dalam pembuatan sistem *learning* pada penelitian ini didapat dari bagaimana seorang anak kecil yang hendak belajar berjalan. *Agent* atau robot humanoid tidak mempelajari *gait* berjalan dari nol, melainkan telah diberi arahan tentang bagaimana cara berjalan, contohnya seperti apabila berjalan maka robot perlu mengangkat kaki untuk mengayun, sebelum mengangkat kaki maka terlebih dahulu badan dimiringkan sehingga *centre of mass* robot masuk ke bagian *support polygon* kaki yang menapak di tanah. Walaupun telah diberi arahan tentang bagaimana cara berjalan, akan tetapi *agent* atau robot tidak tahu seberapa jauh dia harus melangkahkan kakinya agar seimbang, seberapa tinggi dia harus mengangkat kaki dan seberapa miringnya dia sehingga tidak jatuh pada saat kaki akan mengayun. Hal-hal itulah yang robot humanoid coba temukan lewat *learning*, yaitu *policy* untuk bagaimana supaya tidak jatuh pada saat berjalan. Pemberian pengetahuan tentang bagaimana cara berjalan dilakukan dengan membuatkan trayektori dengan beberapa poin-poin yang nantinya akan dilakukan *learning* seperti pada Gambar 6



Gambar 6. Trayektori Berjalan (a) dan Tampak Sampingnya (b)

Total terdapat 6 poin trayektori yang perlu dilakukan *learning* (masing-masing kanan dan kiri ada 3 buah), sehingga dari sini dapat dikatakan satu siklus berjalan robot berjumlah enam langkah atau *step.* Penentuan jumlah *step* atau langkah ini bersifat terserah, apabila semakin banyak maka akan semakin halus juga gerakan dari robot, akan tetapi akan membebani dimensi dari *state-action spaces*. Menurut **(Noreils, 2019)** terdapat 8 *step* untuk manusia berjalan normal.

Metode *rule-based learning* ini menggunakan parameter yang sama dalam metode *learning* pada umumnya, yaitu adanya komponen *state* dan *action.* Dua komponen tersebut digunakan untuk memetakan *action-spaces* yang diambil terhadap *state* pada saat itu. Algoritma *rule-based learning* dalam bentuk *flowchart* ditunjukkan pada Gambar 7



Gambar 7. *Flowchart* Metode *Rule-Based Learning*

Terdapat perubahan gerakan dari eksplorasi ke eksploitasi pada suatu siklus yang dilakukan secara langsung apabila robot telah berhasil melewati satu siklus berjalan yang stabil atau tidak jatuh. Penentuan jatuh atau tidaknya robot dapat terdeteksi dengan bantuan kecepatan sudut **(Arfaq, dkk, 2019)** yang didapat dari pembacaan sensor *IMU* yang berada di *hip.* Dikatakan jatuh apabila kecepatan sudut robot humanoid melebihi 1 rad/s atau kurang dari -1 rad/s. Nilai dari kecepatan sudut tersebut didapatkan dari beberapa kali percobaan yang menjelaskan batas maksimum dan minimum robot humanoid T-FLoW dalam menerima gangguan.

Untuk lebih mudah dalam memahami *flowchart* pada Gambar 7 di atas, sebagai contoh, apabila pada *epoch* pertama, robot humanoid dengan gerakan eksplorasinya dapat berjalan sejauh 15 langkah atau sama dengan 2 siklus berjalan lebih dan jatuh di siklus ketiga, maka nantinya pada *epoch* kedua, gerakan eksplorasi pada siklus pertama dan kedua berubah menjadi eksploitasi secara langsung, sehingga robot memulai *learning* dengan gerakan eksplorasi pada siklus ketiga.

Terdapat sebuah variabel yang menyimpan semua *action* untuk tiap *state* dalam setiap siklusnya apabila robot humanoid berhasil menemukan *gait* berjalan yang stabil. Variabel ini akan terus berkembang sebanyak siklus berjalan yang berhasil robot humanoid temukan dalam keadaan stabil.

$$RS = \{Sc_1, Sc_2, Sc_3, \ldots . Sc_n\} \tag{13}$$

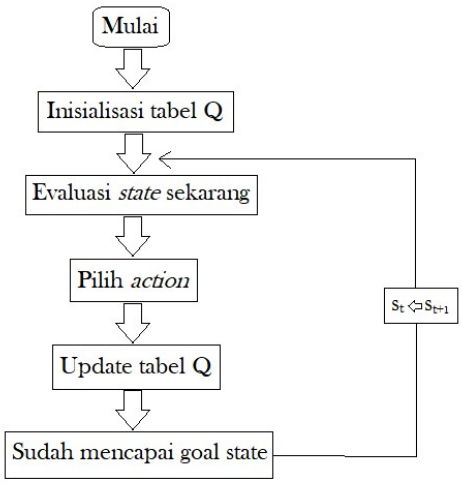
$$Sc_i = \{Sa_1, Sa_2, Sa_3, Sa_4, Sa_5, Sa_6\} \tag{14}$$

Sebagai pembanding, selanjutnya terdapat penjelasan tentang metode *reinforcement learning* yang diterapkan dalam robot humanoid T-FLoW. Dalam penjelasan tersebut, terdapat

perbedaan yang digunakan sebagai alasan mengapa *rule-based learning* mengungguli *reinforcement learning* dalam penelitian ini.

2.3.4 Metode *Reinforcement Learning*

Algoritma *reinforcement learning* yang dipakai adalah jenis model-*free* yaitu *Q-Learning*, sehingga bisa dikatakan cukup sederhana daripada model-*based Q-Learning* **(Le, dkk, 2017)**. Secara umum, di dalam *Q-Learning, agent* atau robot humanoid mengambil *action* a_t saat berada di *state* s_t saat itu, setelah mengambil *action* a_t, maka *agent* akan berada di *state* yang baru s_{t+1}. Di sini dilakukan update terhadap suatu tabel yaitu tabel Q, yang memetakan pasangan *state-action.* Untuk lebih jelasnya dapat dilihat pada Gambar 8



Gambar 8. Algoritma *Q-Learning*

Pada saat robot humanoid akan melakukan update nilai tabel Q untuk tiap $Q_{(s,a)}$, terdapat beberapa ketentuan, yaitu tentang pemberian *reward*. Di mana untuk *reward* memiliki ketentuan sebagai berikut, yaitu akan bernilai 0 apabila robot terdeteksi jatuh (kecepatan sudut robot humanoid melebihi 1 rad/s atau kurang dari -1 rad/s), lalu bernilai 0 apabila robot tidak jatuh akan tetapi belum mencapai satu siklus berjalan dan akan bernilai 1 apabila robot telah berhasil berjalan dengan tidak jatuh dan tiba di siklus akhir dari gerakan berjalan. Dari ketentuan pemberian *reward*, robot hanya akan bernilai 1 pada pasangan *state-action* di langkah atau *step* siklus terakhir. Rumus untuk *update* nilai tiap $Q_{(s,a)}$ adalah sebagai berikut.

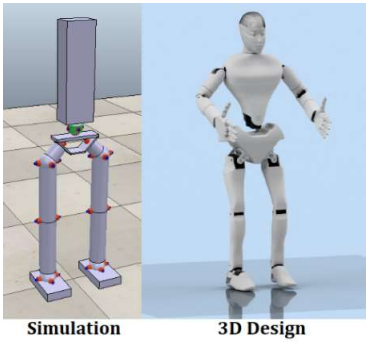
$$Q_{(s,a)} = Q_{(s,a)} + \alpha(r + (y\ max Q_{(s+1,\ a+1)}) - Q_{(s,a)}) \qquad (12)$$

Untuk mempercepat proses *learning*, terdapat tambahan pada bagian sistemnya. Yaitu pada setiap kali 25 *epoch* eksplorasi, akan terdapat *testing* atau gerakan eksploitasi, yang mana pada saat *testing* ini, robot humanoid hanya melihat nilai $Q_{(s,a)}$ yang paling besar ketika akan memilih *action* a_t di *state* s_t. Lalu apabila pada saat *testing* tersebut ternyata robot humanoid mampu berjalan hingga satu siklus, maka pada siklus tersebut nantinya tidak akan terjadi gerakan eksplorasi lagi, akan tetapi menjadi eksploitasi. Dengan begini robot seakan-akan memulai *learning* pada siklus kedua, sehingga dapat mempercepat proses *learning*-nya.

Dua metode di atas yaitu *rule-based learning* dan *reinforcement learning* bisa dibilang hampir sama. Hanya saja pada *rule-based learning*, tidak terjadi *update* dari nilai Q table dan pergantian dari gerakan eksplorasi ke eksploitasi terjadi secara langsung.

3. HASIL DAN PEMBAHASAN

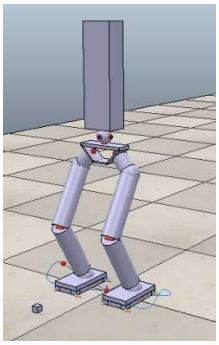
Rencana setiap sistem model di atas disimulasikan menggunakan *software CoppeliaSim* versi *Education* (gratis), di mana model dari robot humanoid T-FLoW menjadi lebih disimpelkan daripada bentuk aslinya seperti pada Gambar 9. Disimpelkan di sini maksudnya adalah menggunakan bentuk primitif (balok, silinder, bola) dalam membentuk pemodelan robot humanoid T-FLoW. Walaupun dengan bentuk yang lebih simpel, akan tetapi masih menyimpan semua parameter kinematik dan dinamik yang dibutuhkan agar hasil yang didapat dari simulasi mendekati seperti robot aslinya, contohnya seperti panjang dan berat dari tiap *link* serta torsi maksimum motor yang dibuat sama dengan robot aslinya. Salah satu tujuan model robot humanoidnya dibuat lebih simpel adalah untuk mengurangi komputasi yang berat pada saat *running. Dynamic Engine* yang digunakan dalam simulasi menggunakan *Newton*, di mana dalam *Dynamic Engine* ini tidak terlalu memakan komputasi yang besar daripada *Dynamic Engine* lain seperti *ODE, Vortex* dan *Bullet*. Untuk *Simulation Time* diatur dalam keadaan normal yaitu 50 *ms*.



Gambar 9. Perbandingan Model T-FLoW untuk Simulasi dan Desain 3D

3.1 Hasil Menggunakan *Reinforcement Learning*

Percobaan pertama, robot humanoid T-FLoW menggunakan *reinforcement learning* untuk menemukan *gait* berjalan yang stabil. Posisi awal robot untuk melakukan *learning* adalah dalam posisi berdiri dengan kaki ditekuk seperti pada Gambar 10.



Gambar 10. Posisi Awal Robot Humanoid T-FLoW

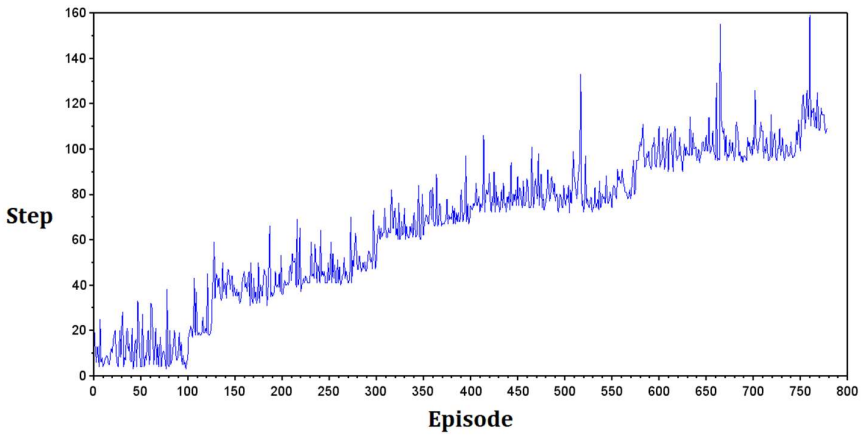
Untuk *action* yang digunakan hanya berjumlah 4 *action*. Hal ini dilakukan untuk mengetahui apakah dengan representasi *state* dan *action* serta pengaturan dalam pemberian *reward* seperti yang direncanakan dalam sistem dapat dikatakan berhasil atau tidak. Dengan jumlah *action* yang sedikit, tentu saja akan berpengaruh dalam proses *learning*, yaitu akan semakin cepat terbentuknya sebuah *policy gait* yang stabil.

Ada beberapa parameter dalam *reinforcement learning* yang perlu dideklarasikan di awal dan tidak akan berubah selama proses *learning*. Tabel 2 menunjukkan beberapa parameter dalam pengujian menggunakan metode *reinforcement learning*.

Tabel 2. Parameter *Reinforcement Learning*

Axis	Nilai (m)
Learning rate α	0.2
Discount rate γ	0.8
Episode	800
Exploration rate	0.96
Exploitation rate	0.04

Setelah melakukan *training* selama 800 episode, robot humanoid T-FLoW mampu berjalan sejauh 120 langkah.



Gambar 11. Grafik Belajar Berjalan dengan *Reinforcement Learning*

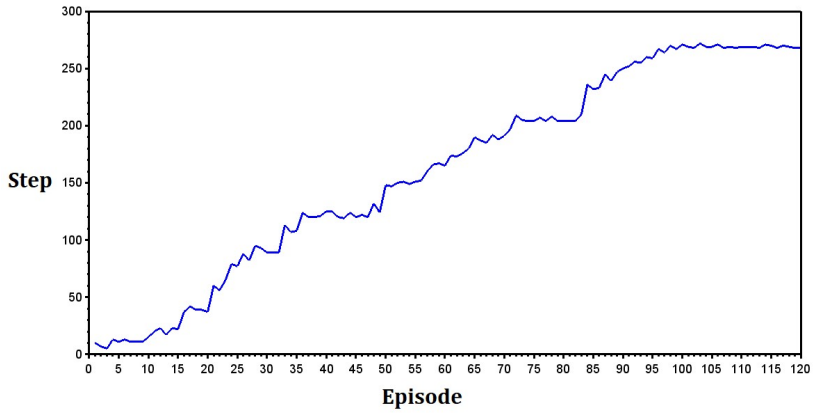
Dari grafik pada Gambar 11 dapat diketahui robot humanoid T-FLoW mengalami kesulitan di awal episode karena hanya bisa melangkah dengan rata-rata di bawah 20 *step*. Hingga akhirnya pada awal episode ke-100, robot mampu untuk meningkatkan jumlah langkahnya untuk berjalan dengan stabil secara signifikan hingga akhirnya di akhir episode 800, mampu untuk berjalan dengan tidak jatuh sejauh 120 langkah.

Metode *reinforcement learning* ini sangat bergantung dari gerakan eksplorasinya, yang mana gerakan eksplorasi ini bersifat acak. Jadi sangat bergantung dari sering-sering terpilihnya suatu pasangan *state-action*. Apabila terjadi suatu kasus yang mana robot sudah mampu untuk berjalan jauh, semisal 50 *step*, akan tetapi pasangan *state-action* tersebut tidak sering terpilih, maka nilai dari Q untuk *state-action* yang mencapai 50 *step* tidak akan terupdate, sehingga robot humanoid tidak akan bisa berjalan sejauh 50 langkah tadi.

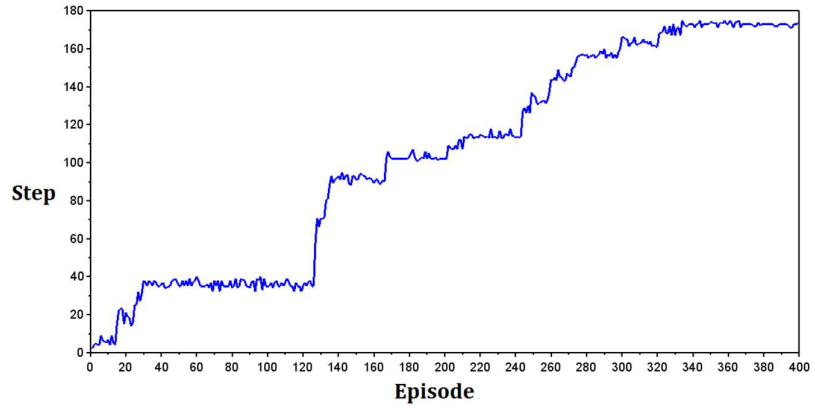
3.2 Hasil Menggunakan *Rule Based Learning*

Untuk percobaan dengan *rule-based learning*, terdapat dua jenis eksperimen berbeda. Yaitu yang pertama dengan menggunakan jumlah *action* yang sedikit yaitu hanya 4 buah *action*, dan yang kedua dengan *action* sebanyak 448. Untuk yang hanya menggunakan 4 buah *action* bertujuan untuk mengetahui bagaimana performa antara metode dengan *Q-Learning* dengan *rule-based learning*. Sedangkan untuk yang 448 *action* adalah pengujian yang sesungguhnya.

Grafik pada Gambar 12 dan Gambar 13 menunjukkan *rule-based learning* dapat membantu robot humanoid menemukan *gait* berjalan yang stabil.



Gambar 12. Grafik Belajar Berjalan dengan *Rule-Based Learning* 4 *Action*

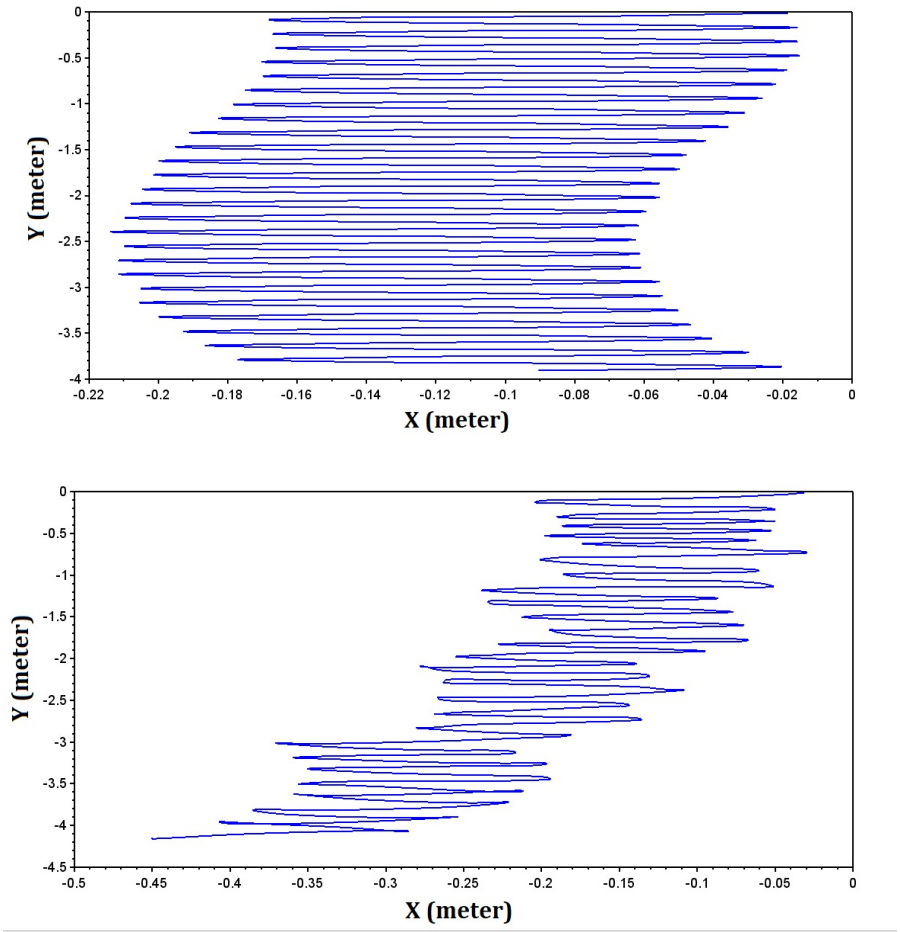


Gambar 13. Grafik Belajar Berjalan dengan *Rule-Based Learning* 448 *Action*

Dengan *training* yang tidak sebanyak seperti dalam metode *reinforcement learning*, dengan metode *rule-based learning* robot humanoid mampu melangkah lebih jauh, yaitu sejauh 270 langkah untuk *training* sebanyak 100 episode saja dengan 4 *action* dan sejauh hampir 180 langkah dengan 448 *action* sebanyak 400 episode. Masalah pada *reinforcement learning* sebelumnya dapat teratasi dengan metode *rule-based learning* ini, karena setiap kali *agent* atau robot humanoid berhasil melangkah 50 *step*, maka *action-action* tersebut akan langsung tersimpan, tidak seperti metode *reinforcement learning* yang harus menunggu untuk kembali terkunjunginya pasangan *state-action* dengan 50 langkah tadi. Karena tujuan dari penelitian ini adalah untuk menemukan *gait* berjalan yang stabil tanpa memperhatikan performa apakah berjalannya bagus atau tidak, maka *rule-based learning* dapat dikatakan lebih cepat dalam menemukan *gait* yang stabil tersebut.

3.3 Perbandingan Berjalan Dengan *Pattern Generation*

Metode yang sering digunakan untuk membuat *gait* berjalan robot humanoid adalah dengan *Pattern Generation* **(Agravante, dkk, 2016)** dan **(Khomariah, dkk, 2015)**. Robot humanoid T-FLoW telah memiliki *gait* berjalan yang stabil sebelumnya, yaitu lewat metode *Pattern Generation* ini. Untuk melihat seberapa bagusnya *gait* yang dihasilkan dari proses *learning*, dapat dilihat dari pergerakan *centre of mass* robot humanoid ketika sedang berjalan.



Gambar 14. Pergerakan *COM Pattern Generation* (Atas) dan *Rule-Based Learning* (Bawah)

Dari Gambar 14, dengan metode *rule-based learning* yang menggunakan 448 *action*, robot humanoid berjalan menyimpang ke kanan sejauh sekitar 40 cm dari posisi awal, sedangkan dengan *Pattern Generation* hanya sekitar 20 cm dari posisi awal. Pergerakan *centre of mass* dari *gait* berjalan yang dihasilkan dengan metode *Pattern Generation* lebih rapi dan bagus daripada *gait* berjalan yang dihasilkan dari proses *rule-based learning*. Hal ini terjadi karena di dalam sistem *rule-based learning* sendiri, belum memasukkan parameter berupa bagus atau tidaknya gerakan berjalan tersebut atau bisa dikatakan sistem *learning* tidak melakukan penilaian terhadap berjalan lurus atau berjalan berbelok-belok. Apabila memasukkan unsur seperti itu, maka metode seperti *reinforcement learning* sangat cocok untuk hal tersebut, karena dalam rumus update nilai Q terdapat variabel *reward* yang dapat digunakan untuk penentuan bagus atau tidaknya *gait* tersebut.

4. KESIMPULAN

Berdasarkan pada pengujian yang telah dilakukan, terdapat beberapa kesimpulan yang dapat dibuat, yaitu yang pertama representasi *state* berupa *Zero Moment Point (ZMP)* dan *action* berupa *end of effector* kaki robot dapat digunakan untuk proses *learning gait* berjalan robot humanoid, kedua yaitu dengan metode *reinforcement learning*, robot humanoid T-FLoW mampu berjalan sejauh 120 langkah dengan melakukan *training* terlebih dahulu sebanyak 800

episode dan dengan *rule-based learning* robot humanoid mampu melangkah lebih jauh yaitu sebanyak 270 langkah dengan hanya *action* sebanyak 4 buah setelah melakukan *training* selama 100 episode saja dan sebanyak 180 langkah dengan *action* sebanyak 448 dan *training* selama 400 episode dan yang ketiga yaitu metode *rule-based learning* dapat menemukan *gait* berjalan yang stabil lebih cepat daripada metode *reinforcement learning* dengan catatan tidak adanya penilaian terhadap bagus atau tidaknya gerakan berjalan yang dihasilkan.

Penelitian ini masih bisa dikembangkan lebih lanjut lagi, yaitu dengan menambahkan adanya *reward* yang akan menjadikan *gait* berjalan robot menjadi lebih rapi dan lebih bagus. Selain itu juga bisa ditambahkan *reward* untuk konsumsi energi yang dipakai, sehingga *gait* berjalan robot bisa dibilang *gait* yang efisien karena konsumsi baterai yang minimal.

UCAPAN TERIMA KASIH

Ucapan terimakasih kepada Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia beserta Laboratorium *Robotic and Intelligent Systems Centre (RoISC)* dan Politeknik Elektronika Negeri Surabaya (PENS) atas dukungan berupa finansial dan non-finansial yang telah diberikan sehingga penelitian ini dapat terlaksana dengan baik.

DAFTAR RUJUKAN

Agravante, D. J., Sherikov, A., Wieber, P. B., Cherubini, A., & Kheddar, A. (2016). Walking Pattern Generators Designed For Physical Collaboration. *IEEE International Conference on Robotics and Automation*, (pp. 1573-1578).

Arfaq, M., Dewanto, R. S., & Pramadihanto, D. (2019). Fall Detection in T-FLoW Humanoid Robot: V-REP Simulation. *International Electronics Symposium on Engineering Technology and Applications,* (pp. 224–228).

Darsih, D. D. (2015). Metode Rule-Base Untuk Analisis Mutu Pembelajaran E-Learning Pada Perguruan Tinggi. *Jurnal Sistem Informasi Bisnis*, *5*(1), 72–78.

Elhasairi, A., & Pechev, A. (2015). Humanoid Robot Balance Control Using The Spherical Inverted Pendulum Mode. *Frontiers Robotics AI*, *2*(21), 1–13.

Hwang, K. S., Lin, J. L., & Yeh, K. H. (2015). Learning to Adjust and Refine Gait Patterns for a Biped Robot. *IEEE Transactions on Systems, Man and Cybernetics Systems*, *45*(12), 1481–1490.

Hwang, K. S., Lin, J. L., & Li, J. S. (2016). Biped Balance Control By Reinforcement Learning. *Journal of Information Science and Engineering*, *32*(4), 1041–1060.

Khomariah, N. E., Pramadihanto, D., & Dewanto, R. S. (2015). Flow Bipedal Robot : Walking Pattern Generation. *International Electronics Symposium*, (pp. 73-78).

Kormushev, P., Ugurlu, B., Calinon, S., Tsagarakis, N. G., & Caldwell, D. G. (2011). Bipedal Walking Energy Minimization By Reinforcement Learning With Evolving Policy

Parameterization. *IEEE International Conference on Intelligent Robots and Systems*, (pp. 318–324).

Le, T. D., Le, A. T., & Nguyen, D. T. (2017). Model-Based Q-Learning for Humanoid Robots. *International Conference on Advanced Robotics,* (pp. 608–613).

Li, T. H. S., Su, Y. Te, Liu, S. H., Hu, J. J., & Chen, C. C. (2012). Dynamic Balance Control For Biped Robot Walking Using Sensor Fusion, Kalman Filter And Fuzzy Logic. *IEEE Transactions on Industrial Electronics*, *59*(11), 4394–4408.

Missura, M., & Behnke, S. (2015). Online Learning of Bipedal Walking Stabilization. *KI - Kunstliche Intelligenz*, *29*(4), 401–405.

Noreils, F. (2019). *Human gait and mass distribution analysis TR 2017-01*. 10.13140/RG.2.2.30605.49126.

Silva, I. J., Perico, D. H., Costa, A. H., & Bianchi, R. A. (2017). Using Reinforcement Learning to Optimize Gait Generation. *Simposio Brasileiro de Automacao Inteligente*, (pp. 288–294).

Silva, I. J., Perico, D. H., Homem, T. P. D., Vilao, C. O., Tonidandel, F., & Bianchi, R. A. (2016). Using Reinforcement Learning to Improve the Stability of a Humanoid Robot: Walking on Sloped Terrain. *12th LARS Latin American Robotics Symposium and 3rd SBR Brazilian Robotics Symposium*, (pp. 210–215).

Zhang, Y., Huang, Q., Bi, S., Min, H., Zheng, Q., & Luo, Y. (2015). Biped Walking On Rough Terfrain Using Reinforcement Learning. *IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems,* (pp. 2061–2066).