

Pengembangan Lingkungan Simulasi untuk Pengujian *Socially Assistive Robots* Menggunakan ROS 2 dan Gazebo

Muhammad Alfie Maulana Fikri
Departemen Teknik Komputer
Fakultas Teknologi Elektro
dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
maulana.17072@mhs.its.ac.id

Mauridhi Hery Purnomo
Departemen Teknik Komputer
Fakultas Teknologi Elektro
dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
hery@ee.its.ac.id

I Ketut Eddy Purnama
Departemen Teknik Komputer
Fakultas Teknologi Elektro
dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
ketut@te.its.ac.id

Abstrak—Pada penelitian ini kami mengajukan lingkungan simulasi yang dikembangkan menggunakan ROS 2 dan Gazebo untuk pengujian *socially assistive robots* (SARs). Di dalam lingkungan simulasi ini, model robot yang digunakan akan diujikan dengan model pengguna serta model-model objek lain secara virtual. Untuk mempermudah pemindahan program dari simulasi ke robot fisik, kontroler robot akan dikembangkan secara terpisah dari lingkungan simulasi yang mana ketika pengujian, keduanya akan saling terhubung menggunakan sistem komunikasi antar proses yang ada di ROS 2. Diharapkan, lingkungan simulasi yang dibuat dapat membantu pengujian SARs dengan meminimalisir resiko, mengurangi biaya, dan menghemat waktu jika dibandingkan dengan melakukan pengujian secara langsung menggunakan robot fisik.

Kata kunci—Simulasi, Assistive Robotics, ROS2, Gazebo.

I. PENDAHULUAN

Selama beberapa tahun terakhir, robot telah mengalami perkembangan yang signifikan dari robot beroda untuk edukasi [1] hingga robot *manipulator* untuk skala industri [2]. Salah satu bentuk perkembangan lain dari robot tersebut adalah *socially assistive robots* (SARs). SARs merupakan jenis robot dalam bidang *socially assistive robotics* yang menggabungkan aspek yang ada pada *assistive robotics* dan *socially interactive robotics* sehingga menjadikan SARs sebagai robot yang mampu memberikan bantuan kepada pengguna dalam bentuk interaksi sosial [3].

Namun, karena sifat dari SARs yang melibatkan interaksi langsung dengan pengguna, maka pengujian dari robot akan menjadi sulit dan beresiko bagi pengguna yang ikut terlibat dalam pengujian tersebut [4]. Salah satu solusi untuk mengatasi masalah tersebut adalah dengan melakukan pengujian secara virtual melalui simulasi robot. Selain bisa meminimalisir resiko, penggunaan simulasi robot sebagai media pengujian robot juga bisa mengurangi biaya yang dibutuhkan dan menghemat waktu pengujian selama pengembangan robot tersebut [5].

Hingga saat ini sudah ada beberapa simulator yang bisa digunakan untuk menjalankan simulasi robot seperti Webots [6],

Gazebo [7], V-REP [8], OpenAI Gym [9], dan lain sebagainya. Namun, simulator-simulator tersebut hanyalah *platform* yang secara umum digunakan untuk membantu pengembangan robot melalui simulasi virtual. Sedangkan pengembangan dari lingkungan simulasi dan kontroler robot untuk simulasi tersebut harus dibuat sendiri oleh pengembang robot.

Untuk itu, pada penelitian ini kami mengajukan penelitian terkait pengembangan lingkungan simulasi untuk pengujian SARs menggunakan ROS 2 dan Gazebo. ROS 2 dan Gazebo sendiri dipilih karena tersedianya banyak *library* yang dapat membantu pengembangan maupun pengujian robot, terutama untuk simulasi. Selain itu, dengan adanya ROS 2, kontroler robot yang diuji melalui simulasi bisa dengan mudah dipindahkan ke robot fisik untuk diuji secara langsung pada pengguna [5].

II. PENELITIAN TERKAIT

Beberapa penelitian sebelumnya telah berhasil dalam mengembangkan lingkungan simulasi untuk robot menggunakan ROS (Pendahu ROS 2) dan Gazebo. Seperti yang dilakukan Qian et al. [10] yang mengembangkan simulasi untuk robot *manipulator*, Zhang et al. [11] yang mengembangkan simulasi untuk robot *quadrotor UAV*, dan Takaya et al. [5] yang mengembangkan lingkungan simulasi untuk pengujian terhadap *mobile robot*. Namun, berbeda dengan penelitian yang telah dilakukan sebelumnya, penelitian yang akan kami lakukan memilih menggunakan ROS 2 agar kontroler robot yang dibuat untuk simulasi memiliki performa yang lebih baik serta dapat bekerja secara *real-time* [12].

Selain itu, penelitian lain juga telah dilakukan oleh Erickson et al. [4] yang mengembangkan Assistive Gym, sebuah *framework* simulasi untuk *assistive robotics* berbasis OpenAI Gym. *Framework* simulasi tersebut kemudian digunakan oleh Clegg et al. [13] untuk mengembangkan metode *learning* melalui simulasi pada kolaborasi antara robot dengan manusia dalam membantu pemakaian baju pada manusia. Namun, karena tidak menggunakan ROS, kontroler robot yang dibuat untuk

simulasi yang menggunakan *framework* tersebut perlu dibuat ulang ketika akan diujikan secara langsung pada pengguna menggunakan robot fisik. Walaupun begitu, penelitian yang dilakukan oleh Zamora et al. [14] menunjukkan bahwa simulasi yang ada pada OpenAI Gym juga bisa diintegrasikan pada ROS dan Gazebo, sehingga tidak menutup kemungkinan bahwa Assistive Gym juga bisa digunakan bersamaan dengan ROS 2 dan Gazebo.

III. ROS 2 DAN GAZEBO

A. Robot Operating System 2 (ROS 2)

Robot Operating System (ROS) [15] merupakan kumpulan dari *libraries*, *drivers*, dan *tools* yang mempermudah pengembangan sistem pada robot. ROS memiliki *command tool* seperti Linux, sistem komunikasi antar proses, dan berbagai macam *packages* yang berhubungan dengan pengembangan sistem pada robot. Proses yang dieksekusi pada ROS disebut sebagai *node*, komunikasi antar proses yang dimiliki menggunakan model *publish/subscribe*, dan data komunikasi yang dikirimkan disebut sebagai *topic*. Suatu proses *publisher* mampu mengirimkan satu maupun lebih *topic*, kemudian proses-proses lain yang melakukan *subscribe* pada suatu *topic* bisa memperoleh isi dari *topic* tersebut. Selain itu ada juga *service* yang memiliki fungsi seperti *topic*, hanya saja dilakukan secara dua arah. *Service* ini bekerja menggunakan model *client/server* dimana *service client* akan mengirimkan data permintaan dalam bentuk *request* dan kemudian *service server* akan mengirimkan data balasan dalam bentuk *response*.

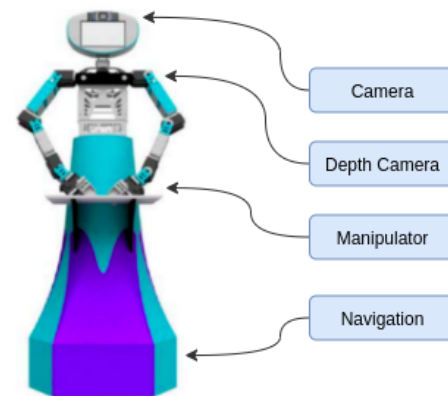
Generasi kedua dari Robot Operating System, ROS 2, merupakan kelanjutan dari ROS yang mengusung reliabilitas dan performa untuk penggunaan *real-time* sembari masih mendukung keunggulan yang dimiliki oleh ROS sebelumnya [12]. Untuk memenuhi kebutuhan reliabilitas dan performa untuk penggunaan *real-time* tersebut, ROS 2 menggunakan *Data Distribution Service* (DDS) [16] [17], standar industri untuk sistem komunikasi *real-time* dan *end-to-end middleware*, yang menggantikan sistem komunikasi antar proses yang dimiliki ROS sebelumnya.

B. Gazebo

Gazebo [7] merupakan bagian dari Player Project [18] yang memungkinkan sebuah simulasi robot dan aplikasi sensor bekerja di lingkungan simulasi indoor maupun outdoor tiga dimensi. Gazebo memiliki arsitektur *client/server* dan model *publish/subscribe* untuk sistem komunikasi antar prosesnya. Setiap objek simulasi di Gazebo dapat diasosiasikan dalam satu maupun lebih kontroler yang akan memproses perintah untuk mengatur dan menentukan keadaan dari suatu objek. Data yang dihasilkan oleh suatu kontroler akan dikirim ke *shared memory* menggunakan *Gazebo interfaces (ifaces)*. Nantinya *ifaces* dari proses-proses lain dapat membaca data tersebut pada *shared memory*, sehingga memungkinkan komunikasi antar proses antara program yang mengontrol robot dan Gazebo, terlepas dari bahasa pemrograman yang digunakan.

IV. DESAIN SISTEM

A. Desain Robot yang Digunakan



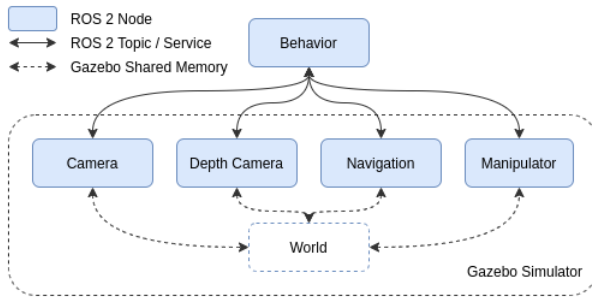
Gambar 1. Diagram desain robot Dienen.

Robot yang akan digunakan pada penelitian ini adalah robot *Dienen* yang merupakan kelanjutan dari robot *IRIS* [19][20] dengan penambahan desain dari robot *ICHIRO* [21] di bagian atas robot. Desain seperti ini secara umum dikenal sebagai desain *mobile humanoid robot* [22], yang merupakan desain gabungan antara robot *mobile* dan robot *humanoid*. Seperti yang terlihat pada Gambar 1, bagian bawah robot menyerupai robot *mobile* dengan penggerak *omnidirectional wheels* yang memungkinkan pergerakan robot secara *holonomic* ke segala arah[23], sedangkan bagian atas robot menyerupai robot *humanoid* yang terdiri atas badan, kepala, dan lengan. Dengan desain *mobile humanoid robot* ini, diharapkan pengguna bisa merasakan interaksi sosial yang lebih baik dengan robot karena memiliki bentuk mendekati manusia [24] sekaligus mempermudah navigasi dari robot ke berbagai tempat.

Robot *Dienen* dilengkapi dengan beberapa sensor seperti IMU (*inertial measurement unit*) untuk mengetahui orientasi dari robot, *rotary encoder* untuk melakukan perhitungan odometri dari robot, *distance sensor* untuk mendeteksi adanya objek lain di sekitar robot, sensor kamera di kepala untuk menangkap citra, dan sensor *depth camera* yang nantinya bisa digunakan untuk melakukan pemetaan dari ruangan. Selain itu robot ini juga dilengkapi dengan dua lengan seperti robot *manipulator* yang bisa diatur pada berbagai posisi dan orientasi [25]. Dengan adanya sensor dan aktuator ini diharapkan robot mampu melakukan tindakan *assistive* secara sosial sesuai dengan data yang didapatkan dari sensor yang ada.

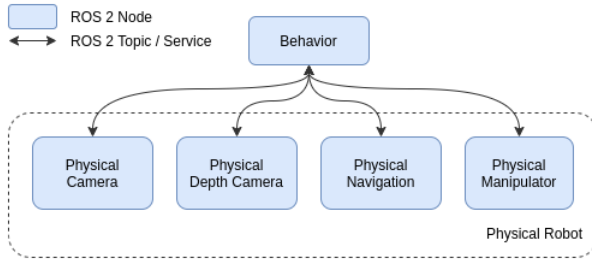
B. Desain Kontroler Robot

Kontroler robot yang digunakan untuk simulasi ini akan dikembangkan menggunakan ROS 2. Kontroler tersebut akan dipisah menjadi beberapa bagian dalam bentuk *ROS 2 node* seperti yang terlihat pada Gambar 2. Setiap *node* yang ada akan terhubung satu sama lain menggunakan sistem komunikasi antar proses ROS 2 yang berupa *topics* dan *services*.



Gambar 2. Diagram sistem kontroler robot di simulasi.

Bagian utama dari kontroler robot tersebut adalah *node behavior* yang berisi program yang mengatur segala tindakan robot berdasarkan data yang didapat dari sensor yang ada di simulasi. Kemudian *node behavior* tersebut akan terhubung dengan empat *node* lain yang merepresentasikan sensor dan aktuator yang ada pada robot. Keempat *node* tersebut akan terpasang di dalam cakupan simulator Gazebo sebagai *Gazebo plugins*, sehingga mampu digunakan untuk mengakses dan memanipulasi data yang ada di simulasi menggunakan sistem *shared memory* pada Gazebo [26].



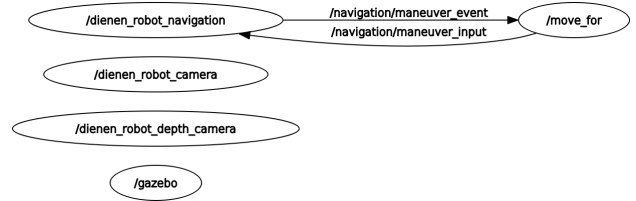
Gambar 3. Diagram sistem kontroler robot untuk robot fisik.

Sistem ini dirancang secara terpisah agar *node behavior* yang diujikan di lingkungan simulasi bisa langsung bekerja pada robot fisik. Seperti yang terlihat pada Gambar 3, transfer kontroler ke robot fisik dapat dilakukan dengan mengubah keseluruhan cakupan yang ada di simulator Gazebo, yang berupa empat *node* yang telah disebutkan sebelumnya, menjadi *node* yang memproses sensor dan aktuator yang ada pada robot fisik. Dengan ini pengujian yang dilakukan di simulasi bisa langsung diterapkan ketika diujikan pada robot fisik karena tidak perlunya pembuatan ulang kontroler yang menyesuaikan sistem yang ada pada robot.

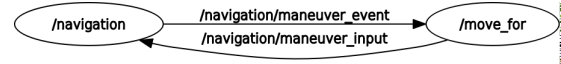
V. HASIL PENGUJIAN

A. Pengujian Gerakan

Pengujian gerakan dilakukan dengan menjalankan *node move_for* sebagai *node behavior* yang akan memerintahkan robot untuk bergerak dengan kecepatan tertentu selama kurun waktu tertentu. Seperti yang terlihat pada Gambar 4, di simulasi, *node move_for* akan terhubung dengan *node dienen_robot_navigation* untuk mengatur kecepatan



Gambar 4. Skema *node* pengujian gerakan di simulasi.



Gambar 5. Skema *node* pengujian gerakan pada robot fisik.

an dari robot yang ada di simulasi menggunakan *topic* */navigation/maneuver_input*. Sedangkan untuk pengujian di dunia nyata, seperti yang terlihat pada Gambar 5, peran dari *node diene_robot_navigation* yang mengatur navigasi pada robot virtual akan digantikan oleh *node navigation* yang mengatur navigasi yang ada pada robot fisik.

Pengujian gerakan terbagi menjadi dua bagian, pengujian gerakan linier dan pengujian gerakan sudut, masing-masing diujikan selama 10 detik pada robot virtual di lingkungan simulasi dan pada robot fisik di dunia nyata. Seperti yang terlihat pada Tabel I dan Tabel II, posisi dan orientasi odometri yang diterima robot memiliki nilai yang sama dengan posisi dan orientasi model robot di simulasi. Hal ini terjadi karena nilai odometri yang dikirimkan oleh *node diene_robot_navigation* adalah nilai yang sama dengan transformasi model robot di simulasi.

Berbeda dengan pengujian di lingkungan simulasi, pengujian di dunia nyata memiliki hasil yang berbeda antara pengukuran langsung dengan nilai odometri yang diterima

Tabel I
HASIL UJI GERAKAN LINIER DI SIMULASI SELAMA 10 DETIK.

Kecepatan		Posisi di Simulasi		Posisi Odometri	
x (m/min)	y (m/min)	x (m)	y (m)	x(m)	y(m)
40	0	5.2	0.0	5.2	0.0
60	0	7.9	0.0	7.9	0.0
-40	0	-5.2	0.0	-5.2	0.0
0	40	0.0	5.4	0.0	5.4
0	-40	0.0	-5.3	0.0	-5.3
40	20	5.0	3.0	5.0	3.0
-20	40	-2.9	5.5	-2.9	5.5

Tabel II
HASIL UJI GERAKAN SUDUT DI SIMULASI SELAMA 10 DETIK.

Kecepatan	Orientasi di Simulasi	Orientasi Odometri
z (rad/min)	z (deg)	z (deg)
40	39.0	39.0
120	118.7	118.7
-40	-39.2	-39.2
-120	-117.1	-117.1

Tabel III
HASIL UJI GERAKAN LINIER PADA ROBOT FISIK SELAMA 10 DETIK.

Kecepatan		Posisi di Dunia Nyata		Posisi Odometri	
x (m/min)	y (m/min)	x (m)	y (m)	x(m)	y(m)
40	0	5.1	-0.1	5.2	0.1
60	0	8.1	0.1	7.9	0.0
-40	0	-5.3	0.1	-5.2	0.1
0	40	0.0	5.4	0.1	5.2
0	-40	0.2	-5.1	-0.1	-5.5
40	20	5.2	3.1	5.3	3.1
-20	40	-3.1	5.3	-2.9	5.3

Tabel IV
HASIL UJI GERAKAN SUDUT PADA ROBOT FISIK SELAMA 10 DETIK.

Kecepatan	Orientasi di Dunia Nyata	Orientasi Odometri
z (rad/min)	z (deg)	z (deg)
40	41.0	39.6
120	120.0	119.3
-40	-40.0	-39.6
-120	-119.0	-116.2

oleh robot. Seperti yang terlihat pada Tabel III dan Tabel IV, terdapat perbedaan antara keduanya yang disebabkan oleh ketidakakuratan yang ada pada sensor yang digunakan untuk mendapatkan nilai posisi dan orientasi. Walaupun begitu, terlepas dari adanya perbedaan tingkat akurasi, dengan menggunakan *node behavior* yang sama, robot tergolong mampu melakukan perintah gerakan yang sesuai ketika diujikan di simulasi maupun di dunia nyata.

VI. KESIMPULAN

Dari pembahasan yang telah dijelaskan di bagian sebelumnya, didapatkan bahwa dengan menggunakan sistem yang dirumuskan, sebuah robot, terutama SARs, dapat dikembangkan dan diujikan baik di lingkungan simulasi menggunakan robot virtual maupun di dunia nyata menggunakan robot fisik. Dengan ini, pengembangan robot akan lebih mudah untuk dilakukan, lebih hemat biaya, serta lebih aman untuk diujikan kepada pengguna karena dapat dikembangkan secara virtual di lingkungan simulasi, sembari tidak meninggalkan kemampuan untuk diujicobakan juga pada robot fisik.

PUSTAKA

- [1] P. Gonçalves, P. Torres, C. Alves, F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klapotcz, S. Magnenat, J. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, vol. 1, 2009.
- [2] M. Blatnický, J. Dižo, J. Gerlici, M. Sága, T. Lack, and E. Kuba, "Design of a robotic manipulator for handling products of automotive industry," *International Journal of Advanced Robotic Systems*, vol. 17, no. 1, 2020.
- [3] D. Feil-Seifer and M. J. Mataric, "Defining socially assistive robotics," in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, 2005, pp. 465–468.
- [4] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp, "Assistive gym: A physics simulation framework for assistive robotics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 169–10 176.
- [5] K. Takaya, T. Asai, V. Kroumov, and F. Smarandache, "Simulation environment for mobile robots testing using ros and gazebo," in *2016 20th*

- International Conference on System Theory, Control and Computing (ICSTCC)*, 2016, pp. 96–101.
- [6] O. Michel, "Cyberbotics Ltd. webots™: Professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, p. 5, 2004.
- [7] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.
- [8] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1321–1326.
- [9] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.
- [10] W. Qian, Z. Xia, J. Xiong, Y. Gan, Y. Guo, S. Weng, H. Deng, Y. Hu, and J. Zhang, "Manipulation task simulation using ros and gazebo," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, 2014, pp. 2594–2598.
- [11] M. Zhang, M. Lan, J. Lin, S. Wang, K. Liu, F. Lin, and B. M. Chen, "A high fidelity simulator for a quadrotor uav using ros and gazebo," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 002 846–002 851.
- [12] Y. Maruyama, S. Kato, and T. Azumi, "Exploring the performance of ros2," in *2016 International Conference on Embedded Software (EMSOFT)*, 2016, pp. 1–10.
- [13] A. Clegg, Z. Erickson, P. Grady, G. Turk, C. C. Kemp, and C. K. Liu, "Learning to collaborate from simulation for robot-assisted dressing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2746–2753, 2020.
- [14] I. Zamora, N. Lopez, V. Vilches, and A. Cordero, "Extending the openai gym for robotics: a toolkit for reinforcement learning using ros and gazebo," *arXiv preprint arXiv:1608.05742*, 2016.
- [15] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," *ICRA Workshop on Open Source Software*, vol. 3, 2009.
- [16] G. Pardo-Castellote, "Omg data-distribution service: architectural overview," in *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.*, 2003, pp. 200–206.
- [17] J. M. Schlesselman, G. Pardo-Castellote, and B. Farabough, "Omg data-distribution service (dds): architectural update," in *IEEE MILCOM 2004. Military Communications Conference, 2004.*, vol. 2, 2004, pp. 961–967 Vol. 2.
- [18] B. Gerkey, R. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," *Proceedings of the International Conference on Advanced Robotics*, 2003.
- [19] R. Dikairono, Setiawardhana, D. Purwanto, and T. A. Sardjono, "Cnn-based self localization using visual modelling of a gyrocompass line mark and omni-vision image for a wheeled soccer robot application," *International Journal of Intelligent Engineering and Systems*, 2020.
- [20] Muhtadin, R. M. Zanuvar, I. K. E. Purnama, and M. H. Purnomo, "Autonomous navigation and obstacle avoidance for service robot," in *2019 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, 2019, pp. 1–8.
- [21] M. Muhtadin, M. Arrazi, S. Ali, T. Pratama, D. Wicaksono, A. Putra, I. Ari, A. Maulana, O. Bramastyo, S. Asshakina, M. Attamimi, M. Arifin, M. Hery Purnomo, and D. Purwanto, "Ichiro robots winning robocup 2018 humanoid teensize soccer competitions," in *RoboCup 2018: Robot World Cup XXII*, 2019, pp. 425–435.
- [22] Z. Mohamed and G. Capi, "Development of a new mobile humanoid robot for assisting elderly people," *Procedia Engineering*, vol. 41, pp. 345 – 351, 2012.
- [23] H. Oliveira, A. Sousa, A. Moreira, and P. Costa, "Dynamical models for omni-directional robots with 3 and 4 wheels," *ICINCO 2008 - Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics*, vol. 1, 2008.
- [24] S. Rossi, M. Staffa, and A. Tamburro, "Socially assistive robot for providing recommendations: Comparing a humanoid robot with a mobile application," *International Journal of Social Robotics*, 2018.
- [25] J. Iqbal, M. Ul Islam, and H. Khan, "Modeling and analysis of a 6 dof robotic arm manipulator," *Canadian Journal on Electrical and Electronics Engineering*, vol. 3, pp. 300–306, 2012.
- [26] "Gazebo plugins," 2021. [Online]. Available: http://gazebosim.org/tutorials?tut=ros_gzplugins