INTERACTION BETWEEN ARTIFICIAL INTELLIGENCE AND MECHANICAL ENGINEERING

[1]Vihaan Agarwal, [2]Rohan Sahni, [3]Sarvjeet Herald

[1]Research Student, [2]Assistant Professor, [3]Professor
RoboGenius, India

Abstract: Mechanical Engineering is now incorporated with Artificial Intelligence to automate mechanical bodies in the past. They are together bringing machines much closer to human-level, by giving machines the power to make their own decisions. To understand this transformation, we look at their interaction and how both are working together to create new opportunities. In our review we found that there are few key differences or gaps between humans and robots and how this gap is blurring or being addressed with developments in coding and mechanical designs. The future of Mechanical Engineering and Artificial Engineering certainly offers exciting opportunities to shape a better society.

Index Terms **– Artificial Intelligence, Mechanical Engineering, Generative Designs, Robot Simulations**

I. INTRODUCTION

Robots are expected to be smart to make our day-to-day life easy. Technology is moving from performing pre-programmed tasks to embedding minds into machines so that they can think and work like humans, beyond human physical and mental capabilities. There are two important aspects to these robots or machines, artificially intelligent minds such as Alexa and Watson and the other is bodies (or the shape form such as Amazon Echo and Sophia) to act like humans. This review paper focuses on investigating both to understand interaction between the two.

Mechanical Engineering, in the most classic sense, deals with modelling and controlling robotic manipulators or mobile robots. We wish to call these robots and machines interchangeably in this paper. It often requires them to do specific and repetitive tasks, many times in an industrial setting and to do it with high precision. Designing a control architecture that achieves those tasks isn’t trivial, but we rarely see these robots as intelligent and being able to do just a pre-programmed task. Artificial Intelligence (A.I.) is a branch of computer science, studying how to design algorithms that simulate intelligent behavior like the human mind. There are many definitions of intelligence, but in this topic, we often refer it to as the ability to act based on sensor inputs and experience and being able to deal with uncertainties and unexpected events. It is not surprising then that, in recent times, these two scientific areas have found an intersection in what we sometimes refer it to as intelligent robotics or machines.

The future of Mechanical Engineering and Artificial Engineering certainly offers a variety of opportunities. The next section focuses on the gaps between robots attempting to mimic human behavior with A.I. in contrast to the humans. Section III focuses on Mechanical Designs and Engineering approaches and discusses legged and humanoid robots. Section IV focuses on approaches used to develop the minds of the robots so that they can think and act like humans. Last section concludes this article by looking at how the professional skill sets are changing and suggested lines of enquiries.

II. ROBOTS AND HUMANS

There are two types of sources of information that the body requires executing an action- body schema and body percept [1]. Body percept describes the body’s position at a particular instant whereas body schema that relates the body parts to its physical constraints. The body schema and percept, working together help us to understand the performance of the body. They help facilitate the imitation of similar actions, as it is much easier to do so than to program the robot to interact with its surroundings and then perform the tasks. The body percept stores the information necessary to perform the actions. With this the body schema possesses the capability to act and also have knowledge of which actions could be performed. Objective is to build computer-based models that describe, simulate, or emulate skills that humans use as they solve problems. However, such a design has limitations. The processes used to achieve the design is far removed from the way humans carry out these processes and much of the design can only be represented symbolically and not mathematically [2].

Intelligence embodied by robots can be viewed as a distributed attribute that starts with mechanical components and extends up through actuators and sensors, their local drivers and controllers, various subsidiary controllers, to usually but not always a central, high-level coordinating controller. Computers manipulate abstract bits and actuators and sensors deal in physical reality. Hexapod (6-legged) robots, for example, are most commonly configured in either two rows of 3 legs (3+3) or at 60 degrees from each other and at equal distance from the center. With some modifications, the equations can be adapted to find the torque required at each joint of an n-legged walking robot with "insect-like" leg configuration. A.I. help find the correct torque required in arm movement and determine the correct drive motor size, etc.

A.I. learns to control a physical robot throughout trial and error. This method, a form of reinforcement learning, is suitable for beating a human in a game. However, it may take a very long time to learn how to perform tasks that humans can do using their hands. In this context, OpenAI engineers have set up a task for the robot hand to manipulate a six-sided cube. The difference was that they added randomized components into its environment so that the A.I. would learn and better understand how to use the cube in a real-world environment [3]. Adding multiple factors helps A.I. to learn how to deal with the unexpected situations. Robots are made to remove repetitive tasks to become more humane. However, humans can learn new tasks with only some trials whereas it would take A.I. a lot of time and errors to learn it. It is still inferior to humans. Dactyl, for example, a system for manipulating objects using a Shadow Dexterous Hand, had to go through 100 years' worth of data. It needed a lot of computing power (about 6,144 CPUs and eight powerful Nvidia V100 GPUs) to learn one task. Same process, resources and new large datasets are needed to be repeated again for learning another task.

Although humanoid robots have come very close to simulating the human body [4], there still exist some key differences between the two. First, Actuators and existing material available to the engineers are not advanced enough yet to overcome some complex differences between the two bodies. Second, mechanical joints have a fixed center of rotation but in human bodies the center of rotation moves along with the joint. Third, Robots use hinge joints whereas the knee and elbow joints in human bodies are constrained by rolling contacts and ligaments. Fourth, the human spine has 10 DOF whereas humanoids have only 3 DOF. Actuators are not able to fully simulate human bodies, new and alternative approaches focusing on “human inspired” robot hardware and controllers will be an interesting life of inquiry in future. Next section discusses the design aspects.

III. MECHANICAL DESIGNS & ENGINEERING

Mechanical and automation engineers focus on designing and manufacturing machines using the principles of mechanics, materials and energy to design and produce and devices of all kinds and shapes. A good design minimizes the quality loss over the life of the design. Designs that are less sensitive to noise are robust. Quality loss is defined as the deviation from the desired performance. Whereas the noise factors are the undesirable, uncontrollable, and costly factors that cause functional criteria to deviate from target values. The design process is motivated by the realization that better Computer-aided design (CAD) tools are required and that to create better tools requires knowledge of how designers design.

Emerging approach is Generative Designs, including autonomous generation of electrical schematics based on diagrams, as well as placing components and routing traces through circuit boards that are based on those diagrams. This approach uses mathematical optimization, combined with finite element analysis to automatically create optimal part geometries. It uses an engineer’s constraints, including loads and fixed geometries as inputs and creates hundreds of the lightest, strongest, and most intricate designs as an output. Such an intelligent automation can deliver the lightest possible product in a short time with the loading conditions of choice. Generative designs are expected to be a common feature in CAD software of the future to benefit engineers and designers.

As the body of research on cognitive design and processes grows and as the field of cognitive modelling matures, a better understanding of the skills and the strategies used by the designers will also emerge [5]. Qualitative methods seem to be of limited use and effect, whilst quantitative methods may act as a positive restriction upon the emergence of the best design solutions. The difficulty arises in measuring and accounting for all the sources of quality loss or information gain over the life of a product. The idea of quantifying what is a “better" design is appealing to engineers, but little work has been done to relate the life-time performance of designs either to the process by which it was created or to quantifiable measures.

Legged robots have a very high potential to be applied in rough terrains due to their high degree of versatility as compared to wheeled or tracked vehicles [6]. While legged robots may have the capacity to outperform many humans and animals in cognitive tasks of reasoning and intelligence, they still lag behind even basic animals in terms of the physical skills of mobility that make them attractive in the first place. Many animals naturally learn to walk and even run within hours of being born, yet the ability to do both remains a challenge for legged robots. However, the new MIT Cheetah 3 robot possesses many critical improvements from its predecessor, the Cheetah 2 including an expanded range of motion, higher force production capabilities and an ability to control these forces in full 3D generality. It makes use of the new general control architecture [7].

Cheetah 3 robots can control ground reaction forces through proprioception, without the use of any force sensors, torque sensors, or series compliance at the joints or feet. It 3 has nearly identical actuators on all three degrees of freedom on each leg, enabling fully 3D control of ground reaction forces. New hip and knee designs allow the robot to operate identically forwards, backwards and flipped upside-down and potentially use its legs for simple manipulation tasks as well as locomotion. Its legs are serially actuated but to keep leg inertia low, the hip and knee actuators are co-axially located at the hip of each leg. Hip joints can rotate continuously, limited only by the length of the wires to the knee actuator, allowing the robot to potentially operate upside down, climb up tall obstacles or use its feet for manipulation above its body. As a simple performance metric, with the leg minimally extended, the robot is capable of producing a purely vertical ground reaction force of over 700 N, about 1.6 times the weight of the robot, per leg. And at 70% extension, a typical configuration during operation, vertical force capability exceeds 1000 N per leg [8].

Trotting, bounding, and pacing gaits are designed to mimic natural animal gaits by controlling the independent phases of each leg. This nominal gait plan is modified during unexpected contact events on the legs. Since the robot does not use any external environment sensors, a contact detection algorithm probabilistically fuses encoder measurements, estimated force, and expected gait phase to estimate the likelihood that each leg is in contact with an object. The robot can differentiate between normal operation, unexpected early contacts, and late missed contacts to adjust its control actions appropriately.

The controllers used on the robot make use of simplified control model templates to optimize ground reaction forces at the footstep locations. It has light limbs with low inertia as compared to the overall body. For this reason, we can reasonably simplify the control model to ignore the effects of the legs for planning ground reaction forces from the stance feet. One of the Cheetah 3 support leg control modes is a Balance Controller which enforces PD control on the center of mass and body orientation whilst also ensuring that foot forces satisfy friction constraints. This helps resolve an optimal distribution of leg forces that drive the approximate COM dynamics. As an alternative to the Balance Controller, the ground force control block could be replaced with a model-predictive controller (MPC) that reasons about trajectory outcomes over a longer temporal horizon. The resulting controller can anticipate and plan around periods of flight and under actuation.

Since the Cheetah does not have external environment sensors, the footstep locations are projected onto an assumed ground plane. To take advantage of the hybrid nature of legged locomotion, a virtual support polygon is defined to provide the desired CoM location that generalizes across all gaits. The robot maintains its forward momentum during the gait while using selected footstep locations to create a smooth reference trajectory that is automatically adapted to the footholds online. Informed nonlinear phase-dependent weighting strategy is implemented by exploiting the robot's knowledge of which feet will be the next to lift off or touchdown and when these state changes are scheduled to occur. To enable Cheetah to traverse stairs and sloped terrain without vision, measurements of each footstep location is used to approximate the local slope of the walking surface and adjust the robot's desired posture.

Cheetah 3 estimates its body states through a two-stage sensor fusion algorithm that decouples estimation of body orientation from an estimation of the body position and velocity. The main idea of the orientation filter is that the gyro provides an accurate reading of the high-frequency orientation dynamics, whereas the presence of a gravity bias on the accelerometer allows it to de-drift the estimate at a comparatively lower frequency. The second stage of the state estimation uses the orientation estimate along with kinematic measurements from the legs to estimate the base position and velocity.

Apart from the legged robots such as Cheetah, the research community is also focusing on commercially driven humanoids such as NAO which can be affordable, light weight, have high performance, are modular and are accessible to the people [9]. Idea is to perform everyday human tasks easily and smoothly. The robot's limbs are also easily replaceable with spare parts, just like changing a spare part in a car. The sagittal plane simulations helped improve the knee and pitch joint actuators and the frontal plane ones helped with hip and ankle roll joint actuators. It was observed that a high value of load to motor inertia could cause instability, whereas a low value would be easier to control but will reduce bandwidth. Custom designed integrated circuits were designed to control actuators and are responsible for servo control, bus control, sensor management and power converters. Each actuator is equipped with magnetic rotary encoders that give absolute outputs. The prototype got a velocity of 0.36km/s even though it was expected to reach 0.6km/s. It is expected to reach the target velocity with closed-loop control.

IV. ARTIFICIAL INTELLIGENCE

Computer based models are used for design processing and for decision making to perform actions by robots. These are used to express methods by which a computer may accomplish a specific task. There are several popular coding & stimulation platforms listed and compared in Table 1 and Table 2. Emerging techniques are leveraging observations of how humans think about the task of transferring knowledge to robots.

Soft Robotics Toolkit is a platform that allows users to design, fabricate, model and test silicon-based soft robots. The toolkit advocates the engineers to be more creative and help them to improve their creations by making use of already existing data. It is very well documented and supported by many embedded devices. Matlab Robotics System Toolkit provides tools and algorithms for designing, simulating, and testing manipulators, mobile robots, and humanoid robots. It also includes a library of commercially available industrial robot models that you can import, visualize, and simulate. The Robot Operating System (ROS) is a flexible framework for writing robot software. ROS is an open-source, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management [10,11]. Python programming language and its libraries are used for simulations and implementing A.I. features [12]. It is popular due to ease of extension, specialization, and no cost platform fee unlike Matlab which is a paid software.

Extending from the previous section, Cheetah has a tiered computing architecture which enables low-level leg and motor control to be run at higher loop frequencies than locomotion control and allows easy expansion of computing resources as needed for future sensing, planning, and navigation tasks. The locomotion computer receives user commands and logs data using Lightweight Communication and Marshalling. This will allow additional computers for vision, planning, and other tasks to easily communicate with the locomotion computer in the future. On the other hand, NAO is equipped with a CPU model that manages audio-video wifi and other advanced modules. Communications are made using WIFI or Ethernet.

Algorithms such as Q-learning and Evolutionary Algorithm are used to teach robots how to move [13]. It can be used to evolve a controller for the robot that can make it move with high precision for example in trajectory tracking, even without physical information about the robot. Evolutionary genetic algorithms are derived from the process of evolution: a natural process. Like in evolution, organisms adapt to their surroundings and make themselves a better fit for that environment. The organism may also undergo a random change that makes them better suited for that environment. Similar is applied in evolutionary algorithms in which best solutions are chosen or selected from random solutions. Changes can also be introduced randomly to better suit the needs and can be iterated until the desired output/performance is achieved.

SOFT ROBOTICS TOOLKIT	ROBOTIC SYSTEM TOOLKIT (MATLAB)	ROS
The Soft Robotics Toolkit is an online treasure trove of downloadable, open-source plans, how-to videos, and case studies to assist users in the design, fabrication, modeling, characterization, and control of soft robotic devices. It will provide researchers with a level of detail not typically found in academic research papers, including 3D models, bills of materials, raw experimental data, multimedia step-by-step tutorials, and case studies of various soft robot designs. It's only applicable for soft robots and not other kinds such as humanoids	For mobile robots, it includes algorithms for mapping, localization, path planning, path following, and motion control. For manipulators and humanoid robots, the toolbox includes algorithms for collision checking, trajectory generation, forward and inverse kinematics, and dynamics using a rigid body tree representation	It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.It requires coding knowledge and a good hardware/gpu support.

Table 1: Coding & Stimulation Toolkits Comparison

Python Packages	Simulators	Softwares
Pybullet Python Robot Simulator Pydy Sympy Sofa Pypi Pybotics Pysim Robosim Robotpy	S-robo TjBot (IBM) Mujoco OpenAI Gym Deepmind Control Suite Gazebo RoboDK Simscape	Robot Operating System (ROS) Modern Robotics Autonomous Robot Simulator Cyberbotics PLM Siemens Microsoft Robotics Developer Studio RoboAnalyzer

Table 2: Coding & Stimulation Toolkits

Trajectory tracking is used to ensure the robot's joints move in the directed path, but it's not always possible due to dynamic coupling. This problem can be solved using neural networks which can approximate any nonlinear function using weights, matrix multiplications and non-linearities. But determining the actual value of the torque to be inputted is still challenging. Here the evolutionary algorithms play an important role. Starting with the best solutions available initially and they are used to alter the parameters to reduce the error in each case. Reinforcement Learning (RL) has the potential to solve many complicated A.I. challenges. However, RL algorithms have become very complex and hard to understand. The Acme framework is used to make the RL algorithms easier to understand and reproduce and help the researchers to streamline the process of creating more creative algorithms such as one being used in Cheetah. Reinforcement learning provides an appealing alternative for automating the manual effort involved in the development of controllers. Two commonly employed deep reinforcement learning algorithms are - A3C and DDPG. However, designing learning objectives that elicit the desired behaviors from an agent will continue to require a great deal of skill-specific expertise.

V. CONCLUSION

Professional roles and skill set are changing as minds and bodies are combined to make smart robots. Skill Set required will be interchangeable. Structural and mechanical engineers will solve complex issues with using generative design and prefabrication automation in the modular industry [14]. Reduced cost of workforce automation will bring manufacturing closer to where people consume things and will create new types of manufacturers focused on customized and personalized machines. From this aspect, knowledge of 3D printing and generative design will also help to customize complex devices to specifications of the individuals, build and deliver in a very short time.

The world may have a robotic workforce. A.I. and automation, in the realm of the mechanical and design engineer, will allow our scope to expand to solving problems never before possible. When AI and generative takes care of the basics, professionals can spend more time on expressing their creativity. Their significant portions of time assembling geometries and doing otherwise menial work will be freed up. A.I. can also be used to optimize product characteristics and internal processes including planning, development, deployment, and maintenance. Therefore, learning A.I. is important for Mechanical Engineers, Design Engineers, Machine Component Designers, Technical Officers, Product Managers, Purchasing Officers, Sales and Maintenance Engineer, etc. This includes acquiring the datasets, understanding, working, and visualizing data for knowledge.

Although humanoid robots have come very close to simulating the human body, there are still 4 key differences between the two. We envisage that that “human inspired” robot hardware and controllers will be an interesting research enquiry. Generative designs are expected to be a common feature in CAD software of the future to benefit engineers and designers. But it is also important to understand the life-time performance of designs either to the process by which it was created or to quantifiable measures. Legged robots such as Cheetah have a very high potential to be applied in rough terrains but mobility and ability to learn still needs improvement [15,16]. A.I. approaches can blur the gap between humans and robots, from the perspective of what they can do and how they can do in a better way, but collecting human experience into a dataset for training is another interesting area to explore. Application of Evolutionary algorithms to the reinforcement learning problem will have quality initial knowledge to get started in an real-world environment and then will be able to learn on its own as humans do. The future of Mechanical Engineering and Artificial Engineering certainly offers exciting opportunities to shape a better society.

REFERENCES

[1] C. A. Acosta-Calderon, H. Hu, "Robot Imitation: Body Schema and Body Percept", Applied Bionics and Biomechanics, vol. 2, Article ID 450645, 18 pages, 2005.

[2] Schulz, Adriana, et al. “Interactive Robogami: An End-to-End System for Design of Robots with Ground Locomotion.” The International Journal of Robotics Research, vol. 36, no. 10, Sept. 2017, pp. 1131–1147, doi:10.1177/0278364917723465.

[3] E. Todorov, T. Erez and Y. Tassa, "MuJoCo: A physics engine for model-based control," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012, pp. 5026-5033, doi: 10.1109/IROS.2012.6386109.

[4] Yamane, Katsu, and Yoshihiko Nakamura. "Robot Kinematics and Dynamics for Modeling The Human Body". Springer Tracts in Advanced Robotics, 2010, pp. 49-60. Springer Berlin Heidelberg, doi:10.1007/978-3-642-14743-2_5.

[5] Dibia, Victor C. et al. "Tjbot". Proceedings of The 2017 CHI Conference Extended Abstracts On Human Factors In Computing Systems - CHI EA '17, 2017. ACM Press, doi:10.1145/3027063.3052965.

[6] Bledt, Gerardo et al. "Contact Model Fusion for Event-Based Locomotion In Unstructured Terrains". 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018. IEEE, doi:10.1109/icra.2018.8460904.

[7] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals, arXiv:2004.00784v3, 2020

[8] Kim, Sangbae, and Patrick M. Wensing. "Design of Dynamic Legged Robots". Foundations And Trends In Robotics, vol 5, no. 2, 2017, pp. 117-190. Now Publishers, doi:10.1561/2300000044. Accessed 12 Oct 2020.

[9] Gouaillier, David et al. "Mechatronic Design of NAO Humanoid". 2009 IEEE International Conference On Robotics And Automation, 2009. IEEE, doi:10.1109/robot.2009.5152516.

[10] Hsu, John, and Nate Koenig. "The Gazebo Simulator ss A Development Tool In ROS". Roscon Minneapolis 2012, 2012. Open Robotics, doi:10.36288/roscon2012-899035.

[11] Quigley, M. “ROS: an open-source Robot Operating System.” ICRA 2009 (2009).

[12] E. Coumans et al. Bullet physics library. Open source: bulletphysics. org, 15(49):5, 2013

[13] Grefenstette, John, “Evolutionary Algorithms in Robotics”, Naval Research Laboratory,1999

[14] Asfour, Tamim et al. "ARMAR-6: A Collaborative Humanoid Robot for Industrial Environments". 2018 IEEE-RAS 18Th International Conference on Humanoid Robots (Humanoids), 2018. IEEE, doi:10.1109/humanoids.2018.8624966.

[15] Bledt, Gerardo et al. "MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot". 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018. IEEE, doi:10.1109/iros.2018.8593885.

[16] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, V. Vanhoucke, G. Brain, and G. Deepmind. Sim-to-Real: Learning Agile Locomotion for Quadruped Robots. In RSS, 2018.