CRONE 2018
Conference Proceedings

CRABOT: A six-legged platform for environmental exploration and object manipulation

Oscar A. Silva[1*,2], **Pascal Sigel**[1], **Warren Eaton**[1], **Cristian Osorio**[1], **Eduardo Valdivia**[1], **Nicolás Frois**[1], **Felipe Vera**[1]

***For correspondence:**
oscar.silva@innovacionyrobotica.com (O.A.S.)

[1] Innovación y Robótica Estudiantil, Universidad Técnica Federico Santa María; [2] Centro de Innovación y Robótica

Abstract The advantage of using legs instead of wheels are several, including better mobility and maximizes energy consumption. Moreover, a leg can be taken into account as a useful arm if the design promotes it. Following this idea, here we describe a six-legged platform called CRABOT, a 3D-printed robot able to be self-sustained by at least four of its legs, leaving other two for robot-object interactions. Integrating a 4D camera, CRABOT has been designed for exploration of environments, while is capable of manipulating objects with any of its legs. This article is focused on the hardware design of the platform and its inverse kinematics required for achieving movement. As possible future extensions, there is still an open issue on the design of different manipulators attachable to the platform.

Introduction

Legged robotic platforms are well known for being highly useful for displacement on unknown, complex environments (***Bekker, 1960***). While the use of wheels is easier in terms of control dynamics, the use of legs can improve not only the reachable space, but also it can increase the displacement velocity, energy use, mobility, among others (***Song and Waldron, 1989***). Instead, wheels platforms are severely limited by the roughness of the terrain, environments commonly present under search and rescue situations ***Murphy et al.*** (***2008***). ***Torres Redondo*** (***2015***) has presented a recent analysis introducing the advantages of using legs in outer-space.

Currently, there are several types of legged robotic platforms, usually identified by the number of legs, commonly ranging between one to eight (***Torres Redondo, 2015***). There are, among others, biped humanoid robots, quadrupeds, hexapods and octopods, insect-like (e.g., ***Chen et al.*** (***2018***)) or bigger animals (e.g., ***Kalouche*** (***2017***)). Despite the advantages for displacement, we believe that the legs are also significantly meaningful when a platform has to interact with the environment, given that the legs by itself can act as manipulators. In that sense, here we propose the design of a six-legged robotic platform called CRABOT, which has been designed for a later integration of different tools that can be attached and used on any of its legs, becoming a Leg-Arm Hybrid robot (***Kajita and Espiau, 2008***). Then, the platform would be potentially capable of doing sample acquisitions or other complex object manipulations, among several other interactions, using tools contained by each leg. The tools could be either different or equals, which automatically reduces the problem of positioning the platform in an exact position near the object in use.

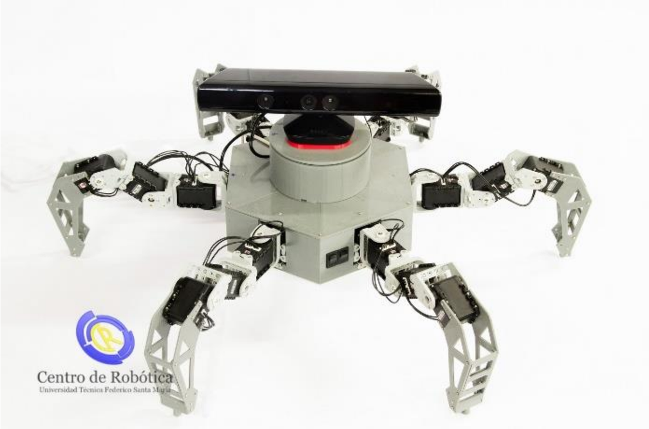

Figure 1. CRABOT, a six-legged 3D-printed platform designed for exploration and object manipulation. Platform features are listed in table 1.

Design of the platform

Structural design and characteristics

In the seek of developing a multiple-purpose platform which could integrate the benefits using of legs, as described before, we have built a hexapod robot, which is similar to a crab (shown in figure 1). All the structural parts of the robots are 3D-printed, including its chassis and legs.

This proposal considers a six-legged platform for an easily positioning and supporting of the robot using generally four legs, leaving at least two of them free for object interaction. Similar to the MELMANTIS-1 platform (***Koyachi et al., 2002***), CRABOT is able to transit in rough terrain while it can use a manipulator when interactions with objects are necessary. CRABOT is able to stand in four or five of its legs, which leaves one or two manipulators free of use, only depending on what are the goals of the manipulation. It can interact with objects with maximum size and weight as the ones detailed in table 1. All the designs of the platform parts are open, and available at a Github repository:
https://github.com/osilvam/Crabot

Figure 2 show the rotational points of the designed legs, without including any ending manipulator. Each leg has four degrees of freedom (DOF), whose last three DOF are used for movements for displacement, being enough for letting CRABOT move around in almost any kind of environment. The last DOF, which is attached to the platform center (i.e., the torso) is in charge of increasing the reaching space for reaching and manipulation purposes, acting as a rotating shoulder for each limb.

While currently each leg ends into a tip, its internal space is aimed for tool storage and deployment on later developments to the platform (see figure 2). Following that design, the platform could incorporate a maximum of six different manipulators, used also as standing points when

Object Manipulation	
Object weight	100 [gr]
Object size	5×5 [cm]
Body characteristics	
Robot weight	2.5 [kg]
Carrying weight	300 [gr]
Autonomy and displacement	
Battery duration	15 [min] (Battery: 5000A a 6×1.3 [Ah])
Walking speed	5 [cm/s]

Table 1. Platform characteristics

needed. Given the radial symmetries of the platform design, the addition of manipulators (e.g., a platform with eight legs/manipulators) only imply architectural modifications at the torso.

Hardware components

Mobile actuators

Each limb of the platform is composed of four Smart Serial Servo from Dynamixel[1] (model AX-18^a, with a maximum torque of 1.8 [Nm]). Prior tests using servos from the same series but with 1.2 [Nm] produced an unstable gate lacking the strength for self-sustain the weight of the platform. These servomotors are easily connected and controlled by their serial capability. Also, as their name introduces, these servos are *smart* as they include an internal driver which not only allows them to reach an exact positioning but also provides to an external controller meaningful information as internal motor temperature, the level of current consumption, the instant torque being applied, among others. This motors also include a gearbox necessary for applying the levels of mechanical torque required for moving a platform as CRABOT.

Sensors

CRABOT includes an inertial measurement unit (IMU), which gives information about the absolute inclination of the robotic torso, computed by estimating pitch, yaw and roll, and its acceleration on each of the three common coordinate axes. The platform also includes a 4D Kinect sensor, incorporating an RGB camera and proximity sensing, information that allows the platform to explore and reconstruct unknown environments. The Kinect is replaceable with any other suitable 4D camera, in terms of the task.

The platform also includes sensors integrated to its servo motors. These sensors give CRABOT online information about how each actuator is interacting with the environment, information that can be used for computing the strength applied in a given task, mean and maximum current consumption, supported load, among others.

Drivers and controllers

For managing all the robot motor signals, the platform includes the CM-700 driver (and its complementary board CM-700 SUB board), which is developed by ROBOTIS specifically designed for controlling Dynamixel servo motors. The use of this drivers allows to easily manage and control several motors in parallel. This board is connected to a central controller unit. As the CM-700 board is connected through TTL/RS485, to be used with any computer an interface to USB has to be considered, as the USB Downloader LN-101 from ROBOTIS.

[1] Dynamixel is a line of actuators from ROBOTIS. More information can be found at http://en.robotis.com/.

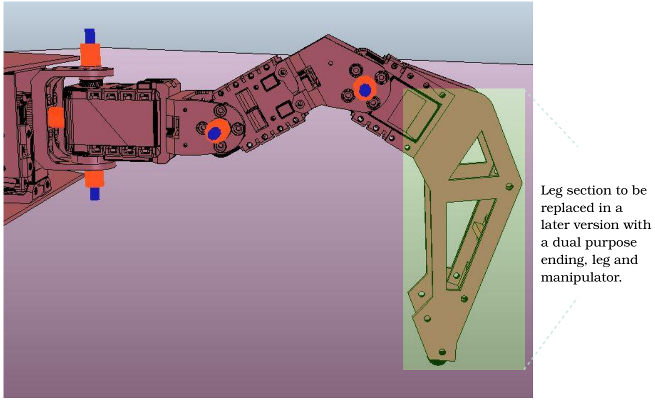
Leg section to be replaced in a later version with a dual purpose ending, leg and manipulator.

Figure 2. Leg design. Each leg is composed by four degrees of freedom, which are shown here by the orange sticks.

The central controller unit in charge of the behavior of the platform is a microcomputer that is placed inside the torso. CRABOT was developed for using the ODROID board from Hardkernel [2], but any computer unit could be attached to the platform.

Movement algorithms

Similar to a person and other animals, a robotic platform has to perform several actions in order to complete a task. For just displacement, we can think in moving forward, turning, changing the center of mass for allowing platform inclinations, among others. For any required movement, there are common descriptions that need to be addressed: the platform forward and inverse kinematics.

Direct kinematics

In order to reach any place at the space, the different motors have to act and rotate. To address this problem, which has infinite possibilities and is solved by inverse kinematics, we will start describing a representational framework to acquire the direct kinematics of the platform. Figure 3 shows a representation of each link and a different coordinate axes at each joint. Then, at each coordinate axes, each joint has to rotate with respect of its own coordinates. Then, the movement of the tip of each leg can be thought as a function f that depends on a set of angles corresponding to the different DOF of a limb, and which gives as a result a point in the space with coordinates (x, y, z), as

$$\vec{f}(\vec{\theta}) = \vec{X} \tag{1}$$

where $\vec{\theta} = (\theta_1, \theta_2, ..., \theta_n)$ is the set of angles and $\vec{X} = (x, y, z)$. Moreover, $\vec{f}(\vec{\theta})$ can be expressed as a matrix multiplication, using a transform matrix considering rotation and translation for each coordinate axis.

Then, from the base to the tip of each limb, the ending point can be computed as

$$\vec{f}(\vec{\theta}) R_1(\theta_1) R_2(\theta_2) T_1(\vec{d}_1) R_3(\theta_3) T_2(\vec{d}2) R_4(\theta_4) T_3(\vec{d}_3) \tag{2}$$

where the $R(\theta)$ and $T(d)$ correspond to a matrix rotation by a given angle θ and a matrix translation given a distance d, respectively, as shown in Box 1.

Then, by using the representation in equation (1), the position of the tip of any leg of the platform can be obtained just knowing the rotational angles of each of motor of that leg.

Now, with this information, we are not capable of computing angles when we need to reach a point, but rather we are able to now the position of the tip given certain rotational angles. In order to know how to reach any point, we need the inverse of equation (1), finding the inverse kinematics of the platform.

[2] Specifications and more information about the ODROID board can be found at https://www.hardkernel.com.

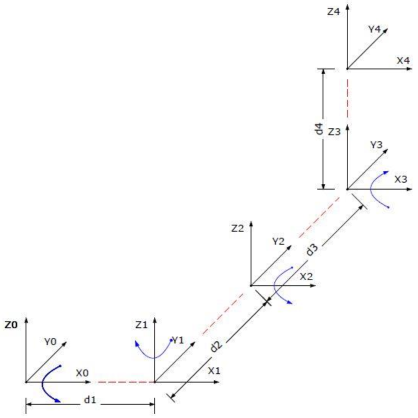

Figure 3. Diagram showing the representation of each coordinate axes of a leg, one for each joint plus one representing the traslation from the last joint to the tip.

Box 1. Example of rotation and translation matrices

Here, examples of rotational matrices are shown for each coordinate: $Rot(x,\theta)$, $Rot(y,\theta)$ and $Rot(z,\theta)$. Also, a translation matrix $Trans(x_0,y_0,z_0)$ is presented. These matrices are the bases for representing the position of the tip of a leg by the relative coordinates of the robot.

$$Rot(x,\theta)=\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & cos(\theta) & -sin(\theta) & 0\\ 1 & sin(\theta) & cos(\theta) & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad Rot(y,\theta)=\begin{bmatrix}cos(\theta) & 0 & sin(\theta) & 0\\ 0 & 1 & 0 & 0\\ -sin(\theta) & 0 & cos(\theta) & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$Rot(z,\theta)=\begin{bmatrix}cos(\theta) & -sin(\theta) & 0 & 0\\ sin(\theta) & cos(\theta) & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad Trans(x_0,y_0,z_0)=\begin{bmatrix}1 & 0 & 0 & x_0\\ 0 & 1 & 0 & y_0\\ 0 & 0 & 1 & z_0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Inverse kinematics

In order to find rotational angles that would position the tip of a leg in a certain position in the space, we can express equation (1) using Taylor's series. Then, each coordinate can be expressed as

$$x = f(\vec{\theta}_0) + \frac{\partial f(\vec{\theta})}{\partial \theta_1}\Big|_{\vec{\theta_0}} \cdot (\theta_1 - \theta_{0_1}) + \cdots + \frac{\partial f(\vec{\theta})}{\partial \theta_n}\Big|_{\vec{\theta_0}} \cdot (\theta_n - \theta_{0_1}) + error\left((\vec{\theta} - \vec{theta}_0)^2\right) \tag{3}$$

Now, if we express all the coordinates following equation (3), we can use the matrix form

$$\begin{bmatrix}x\\ y\\ z\end{bmatrix} = \begin{bmatrix}f_1(\vec{\theta}_0)\\ f_2(\vec{\theta}_0)\\ f_3(\vec{\theta}_0)\end{bmatrix} + \begin{bmatrix}\frac{\partial f_1}{\partial \theta_1} & \cdots & \frac{\partial f_1}{\partial \theta_n}\\ \frac{\partial f_2}{\partial \theta_1} & \cdots & \frac{\partial f_2}{\partial \theta_n}\\ \frac{\partial f_3}{\partial \theta_1} & \cdots & \frac{\partial f_3}{\partial \theta_n}\end{bmatrix}_{\vec{\theta_0}} \begin{bmatrix}\theta_1 - \theta_{0_1}\\ \vdots\\ \theta_n - \theta_{0_n}\end{bmatrix} \tag{4}$$

There, we have removed the error as its value is not meaningful on the results. Then, equation (4) can be rewritten as

$$\vec{X} = \vec{F}(\vec{\theta}_0) + J_{\vec{\theta}} F(\vec{\theta}) \mid_{\vec{\theta_0}} \cdot (\vec{\theta} - \vec{\theta}_0) \tag{5}$$

where $J_{\vec{\theta}} F(\vec{\theta})$ is the Jacobian matrix of the spatial coordinates of the direct kinematics. Then, it is possible to inverse this equation, obtaining $\vec{\theta}$:

$$\vec{\theta} = J_{\vec{\theta}}^{-1} F(\vec{\theta})\Big|_{\vec{\theta_0}} \left(\vec{X} - \vec{F}(\vec{\theta}_0)\right) + \vec{\theta}_0 \tag{6}$$

Moreover, we can take the first derivative of equation (6) to obtain the angular velocity necessary for optimal movement, considering a desired (known) linear velocity:

$$\frac{\partial \vec{\theta}}{\partial t} = J_{\vec{\theta}}^{-1} F(\vec{\theta})\Big|_{\vec{\theta_0}} \cdot \frac{\partial \vec{X}}{\partial t} \tag{7}$$

Using equations (6) and(7), we are able to calculate angles for reaching a certain point, and to control the movement velocity in order to not produce wrong, possible harmful, movements of each leg.

In order to keep all the movements smooth, for any given point, the algorithm that reach a place takes into consideration the initial point and the final point. Using this two points, a trajectory is constructed (which depends on the given task, but could be linear), from which a closer point to the initial position is extracted. The ending point for computing the new motor angles, following equations (6) and (7), is the closer one extracted from the constructed trajectory. Once that the new angles are obtained, the movement of a leg is triggered and performed.

The developed software for controlling the platform, including a program that runs in a server for remote control using a web-based application usable in mobile devices, are also available at the online repository (https://github.com/osilvam/Crabot).

Conclusions

CRABOT is a multipurpose platform developed for remote exploration and object manipulation. The current advances allows to control the platform, with described mathematical descriptions of its direct and inverse kinematics. The platform is capable to sustain up to two legs in the air for object interaction, which makes it a very versatile legged robot.

While the necessary hardware and mathematical approaches for controlling the platform are here described, there is still an open problem about the specific design of different manipulators that could be attached to its legs. Extending the platform with an analysis and design of manipulators can contribute, as explained here, to the development of robots with lower energy consumption ratios while exploiting the mobility advantages of using legs.

References

Bekker MG. Off-the-road locomotion: research and development in terramechanics. University of Michigan Press; 1960.

Chen Y, Doshi N, Goldberg B, Wang H, Wood RJ. Controllable water surface to underwater transition through electrowetting in a hybrid terrestrial-aquatic microrobot. Nature communications. 2018; 9(1):2495.

Kajita S, Espiau B. Legged robots. In: *Springer handbook of robotics* Springer; 2008.p. 361–389.

Kalouche S. GOAT: A legged robot with 3D agility and virtual compliance. In: *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on* IEEE; 2017. p. 4110–4117.

Koyachi N, Adachi H, Izumi M, Hirose T, Senjo N, Murata R, Arai T. Multimodal control of hexapod mobile manipulator MELMANTIS-1. In: *Proceedings of 5th International Conference on Climbing Walking Robots*; 2002. p. 471–478.

Murphy RR, Tadokoro S, Nardi D, Jacoff A, Fiorini P, Choset H, Erkmen AM. Search and rescue robotics. In: *Springer handbook of robotics* Springer; 2008.p. 1151–1173.

Song SM, Waldron KJ. Machines that walk: the adaptive suspension vehicle. MIT press; 1989.

Torres Redondo J. Analysis and optimization for a hexapod walking robot for planetary missions. PhD thesis, Industriales; 2015.