Humanoid Ankle Mechanism: Real-Time Control Architecture

Maysoon Ghandour[1,*], Naima Ait Oufroukh[1], Samer Alfayad[1], and Adrian Olaru[2]

[1]IBISC Laboratory, University of Paris Saclay, Evry, France
[2]Department of Robotics and Production System, of Robotics and Production System, Bucharest, Romania
Email: maysoon.ghandour@universite-paris-saclay.fr (M.G.); naima.aitoufroukh@univ-evry.fr (N.A.O.);
samer.alfayad@univ-evry.fr (S.A.); adrian.olaru2301@upb.ro (A.O.)
[*]Corresponding author

Manuscript received May 25, 2024; revised July 12, 2024; accepted September 1, 2024; published December 12, 2024

***Abstract*—The high demand for humanoid robots capable of operating effectively in various environments has promoted the development of software, hardware, and control architecture. Electro-hydraulic actuators have gained interest due to their power-to-weight and also their power-to-volume ratio, making them suitable for robotics applications. Nevertheless, this highlighted the importance of safety due to the high forces generated, which presents a critical challenge. A stable control system capable of operating in real-time is required for safe robot interaction without system failure. This paper aims to develop a real-time control architecture for an electro-hydraulic humanoid while presenting the hybrid mechanism of the ankle as a case study. The solution proposed is a distributed, open-source, real-time, and modular control architecture. The results demonstrated the potential for using EtherCAT and open-source software to develop a stable robot control architecture.**

***Keywords*—humanoid, real-time, EtherCAT, electro-hydraulic actuator**

I. Introduction

In recent decades, an increased interest has been in developing humanoids that can operate in unstructured areas. However, this comes with the drawback of having many actuated joints. Therefore, a reliable and adaptable control architecture is required to control the high number of degrees of freedom and achieve stability and safety of the robot.

HYDROïD is an electro-hydraulic actuated humanoid robot designed to navigate rough terrains, assist the elders, interact socially with humans, and assist humans in industries [1]. To guarantee the dynamic motion of HYDROïD, it is essential that all the robotic systems display high reactivity, including the actuators, sensors, and software architecture. Since the utilized actuator is critical in determining the robot's performance and overall capability. HYDROïD's latest generation is equipped with the Servo Electro-Hydraulic Actuator (SEHA) [2]. This actuator, SEHA, is a patented actuator that is featured by the power-to-volume and the power-to-weight ratios, which generate hydraulic energy locally at the actuated joints, providing each joint with the required power to prevent losses.

The designed control architecture and the controller distribution in the robot are the primary determinants of its dynamic balance and safety during user interaction. This raises the question of how best to arrange the physical placement of the control architecture. A centralized approach can be adopted, with all the robot joints being controlled by one control unit. It is simple but constrained by the large amount of computation needed. The decentralized approach is another solution in which the control is distributed among the different joints, each with decision-making capabilities and without a central authority, allowing for more efficient control and sensor data monitoring with less computational power. However, this method has two significant drawbacks: i) coordination issues and ii) delays because of the communication between the different controllers. A distributed control architecture is a solution with multiple controllers physically distributed among the robot's joints with central coordination. The controllers communicate over a network, and each has the decision-making abilities. In a distributed architecture, coordination and synchronization between the distributed units are challenging and necessitate robust communication.

A significant challenge while developing humanoids is the requirement of a real-time environment. This is essential to ensure the robot's stability and safety, especially for complex robots collaborating with humans. Hence, to guarantee the deterministic behavior and handle the interrupts within a predefined time frame, a Real-Time Operating System (RTOS) [3].

Moreover, the choice of the communication field bus used to communicate between the high-level controller and the low-level controller is critical as it limits the update rate in the system. Despite advances in low-level controllers and modern communication, some robot update rates are relatively low due to communication fieldbus limitations. Therefore, humanoid communication protocol must meet certain standards, including reliability, speed, stability, and the ability to handle all necessary data. HUBO robot [4] uses CANbus with 100 Hz as the main control frequency. Also, iCub, which uses CAN, achieved a 100 Hz at the control loops [5]. Although some efforts were made to improve performance using parallel CAN networks, the update rate achieved is still low. EtherCAT bus, on the other side, is used in some robots like RoboSimian [6], Hydra [7]. These robots achieved a 1kHz update rate. Also, using EtherCAT, a 2 kHz for the updated rate is achieved in the LOLA robot achieved [8].

A real-time and open-source control architecture using EtherCAT communication protocol developed for an electro-hydraulic humanoid is presented in this paper. The work includes an analysis of the ankle mechanism and the actuator utilized, the implementation of both software and the joint levels, and the communication between them.

The overview of HYDROïD robot is represented in the section II, a case study showing the modeling of a single joint in the robot, the Inverse Geometric Model (IGM), and the Inverse Kinematic Model (IKM) of the hybrid ankle mechanism are shown in Section III. The proposed solution for the control architecture is shown in Section IV. The simulations, the experiments conducted to evaluate the proposed solution, and the results achieved are presented in

Section V. Lastly, the conclusion and the outlined future works are represented in Section VI.

II. HYDROïD Humanoid Robot Overview

HYDROïD is a full-size humanoid robot comprising 51 DoFs emulating the human joints. 36 of HYDROïD's joints that are for the body are hydraulically actuated with a special hydraulic integration to reduce tubes [9], and 15 electrically actuated for the head. HYDROïD's initial version is developed with electro-hydraulic actuators and double-stage servo valves. Fig. 1 shows the robot with the general characteristics and the kinematic structure.

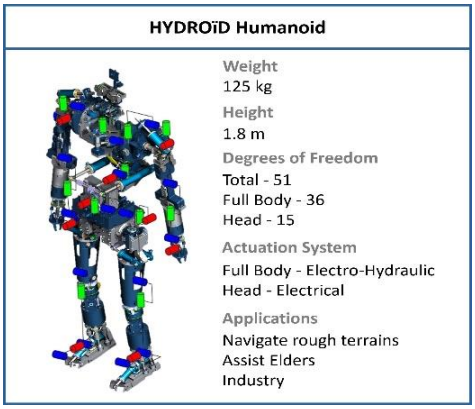

Fig. 1. HYDROïD humanoid robot.

Upon studying the human morphology, the different mechanisms in HYDROïD were developed. These mechanisms achieve the necessary range of motion, speed, and torque and, at the same time, are lightweight and compact, respecting human appearance [10]. Two hybrid mechanisms are developed for HYDROïD, each consisting of a parallel structure held by a rotary hydraulic actuator. A mechanism characterized by achieving in the pitch axis a wide range of motion is proposed for the shoulder, hip, and torso [11], while another mechanism characterized by its compactness is dedicated to the wrist and the ankle of the robot [12].

III. Case Study Determining the Computational Load

Understanding the model and the complexity of the actuator used in the robot joints is a preliminary step to designing and selecting the different components in the control architecture. Determining the complexity of the mechanisms and, hence, the computations required is crucial to developing an efficient control architecture. Two existing layers can be identified: i) the mechanisms and ii) the joints in the mechanism that consists of a rotary hydraulic actuator driven by a two-stage flapper servo valve, as shown in Fig. 2. This section presents a study conducted to investigate the modeling of the joint level. Additionally, it examines the ankle mechanism's inverse kinematics and inverse geometry.

A. Modeling of Electro-Hydraulic Actuator

HYDROïD's joints consist of either linear hydraulic actuators and double-stage servo valve [13] or rotary hydraulic actuators driven by two-stage servo valves. In this case study, we will consider the rotary actuators.

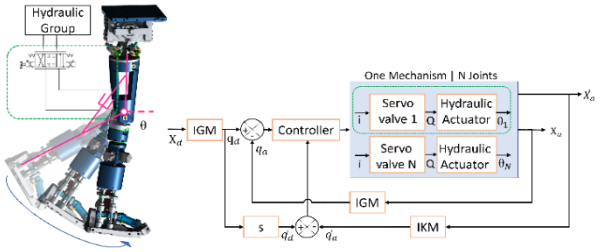

Fig. 2. Control loop for a single mechanism in HYDROïD humanoid.

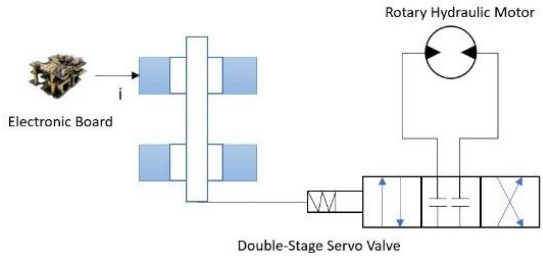

Fig. 3. Schematic representing the joint level consisting of a rotating hydraulic motor and a double-stage servo valve controlled with an electronic board.

The joint level that is shown in Fig. 3 consists of a two-stage flapper servo valve controlled by a current from an electronic board and a rotating hydraulic motor. Table 1 shows the definitions of the symbols that are used throughout this section.

Table 1. Terms, symbols and units

Terms	Symbol	Units
Engine's geometric capacity	qm	cm^3
Flow losses gradient	am	$cm^5/daNs$
Motor's moment of inertia	JM	$daNcms^2/rad$
Stator's moment of inertia	Js	$daNcms^2/rad$
Resisting moment	M_r	daNcm
Moment of friction	M_f	daNcms
Moment losses gradient	bm	daNcms/rad
Coefficient of dry friction	c_{f_u}	-
Hydraulic motor shaft's angular speed	ω	rad/s
Oil's modulus of elasticity	E	daN/cm^2
Damping factor	ξ	-
Natural pulsation	ω_n	rad/s
Servo valve output flow	Q_{current}	cm^3/s

The rotary hydraulic transfer function is shown in Eq. (1).

$$\boldsymbol{H(s)} = \frac{\boldsymbol{\omega(s)}}{\boldsymbol{Q(s)}} = \frac{\boldsymbol{a_0}}{\boldsymbol{b_2 s^2 + b_1 s + b_0}} \tag{1}$$

The transfer function coefficients shown in Eq. (1) are represented in Eqs. (2)–(5).

$$a_0 = \frac{q_m\left(1 - c_{f_u}\right)}{2\pi} \tag{2}$$

$$b_0 = \left(\frac{q_m}{2\pi}\right)^2 \left(1 - c_{f_u}\right) + a_m b_m \tag{3}$$

$$b_1 = (J_m + J_s)a_m + \frac{q_m}{2E} b_m \tag{4}$$

$$b_2 = (J_m + J_s)\frac{q_m}{2E} \tag{5}$$

The damping factor ξ is represented in Eq. (6) and the natural pulsation ω_n is presented in Eq. (7).

$$\xi = \frac{b_1}{2\sqrt{b_0 \cdot b_2}} \tag{6}$$

$$\omega_n = \sqrt{\frac{b_0}{b_2}} \tag{7}$$

The response is over-damped if the damping factor ξ exceeds one. The angular velocity ω of the servo valve driven hydraulic motor is presented in Eq. (8).

$$\boldsymbol{\omega_1} = \frac{\boldsymbol{a_0}}{\boldsymbol{b_0}} \cdot \boldsymbol{Q}_{\text{current}} \cdot \left(\mathbf{1} + \frac{\boldsymbol{b_2} \cdot \boldsymbol{e}^{-\boldsymbol{b_1 t}}}{\boldsymbol{b_1 - b_2}} - \frac{\boldsymbol{b_1} \cdot \boldsymbol{e}^{-\boldsymbol{b_2 t}}}{\boldsymbol{b_1 - b_2}}\right) - \frac{\boldsymbol{M_f} \cdot \boldsymbol{a_M} \cdot \left(\mathbf{1} - \boldsymbol{e}^{-\frac{\boldsymbol{b_0 t}}{\boldsymbol{a_M} \cdot (\boldsymbol{J_M} + \boldsymbol{J_S})}}\right)}{\boldsymbol{b_0}} \tag{8}$$

The response is critically damped if the damping factor ξ is unity, and the resulted angular velocity is shown in Eq. (9).

$$\boldsymbol{\omega_2} = \frac{\boldsymbol{a_0}}{\boldsymbol{b_0}} \cdot \boldsymbol{Q}_{\text{current}} \cdot [\mathbf{1} - \boldsymbol{e}^{-\boldsymbol{\omega_n t}} \cdot (\mathbf{1} - \boldsymbol{\omega_n t})] - \frac{\boldsymbol{M_f} \cdot \boldsymbol{a_M} \cdot \left[\mathbf{1} - \boldsymbol{e}^{-\frac{\boldsymbol{b_0 t}}{\boldsymbol{a_M} \cdot (\boldsymbol{J_M} + \boldsymbol{J_S})}}\right]}{\boldsymbol{b_0}} \tag{9}$$

Finally, the response is underdamped if the damping factor is below unity, and the resulted angular velocity is outlined in Eq. (10).

$$\boldsymbol{\omega_3} = \frac{\boldsymbol{a_0}}{\boldsymbol{b_0}\sqrt{\mathbf{1}-\boldsymbol{\xi}^2}} \cdot \boldsymbol{Q}_{\text{current}} \cdot \left[\mathbf{1} - \boldsymbol{e}^{-\boldsymbol{\omega_n \xi t}} \cdot \sin\left(\boldsymbol{\omega_n}\sqrt{\mathbf{1}-\boldsymbol{\xi}^2} \cdot \boldsymbol{t} + \arctan\left(\frac{\sqrt{(\mathbf{1}-\boldsymbol{\xi}^2)}}{\boldsymbol{\xi}}\right)\right)\right] - \frac{\boldsymbol{M_f} \cdot \boldsymbol{a_M} \cdot \left[\mathbf{1} - \boldsymbol{e}^{-\frac{\boldsymbol{b_0 t}}{\boldsymbol{a_M} \cdot (\boldsymbol{J_M} + \boldsymbol{J_S})}}\right]}{\boldsymbol{b_0}} \tag{10}$$

B. Kinematic Model of Ankle Hybrid Mechanism

HYDROïD's ankle mechanism compromises 3 DoF. This mechanism is a hybrid mechanism that has serial and parallel substructures [14]. Fig. 4 shows the ankle's kinematic structure. Four closed loops are found in the ankle mechanism named Ch_j, and each have the links: Co_1, Co_2, Co_3, Co_1^j, Co_2^j, Co_3^j, Co_4^j, Co_5^j, and Co_6^j with j = 1, 2, 3, 4. The joint q_s (r_1^1, r_1^3). While joint q_f is actuated by the two linear actuators (r_1^2, r_1^4). The roll, yaw and pitch angles identify the end effector's orientation with respect to the base and grouped in vector X_a = (θ_r, θ_p, θ_y). The roll rotation θ_r and the linear actuator length r_{ij} and inputs to hybrid mechanism while the angles q_v, q_s, and q_f are the outputs.

1) Inverse Geometric model

The stroke of the linear actuator is determined by the IGM. Hence, r_{1j} will be calculated through the IGM for a given end effector position. The roll rotation is achieved by the serial part, and hence, $qv = \theta r$. Nevertheless, the parallel mechanism achieves pitch and yaw rotations, which require more calculations. The following notations are used while calculating the IGM:

- Loop_j for j = 1, 2, 3, 4 is the j^{th} closed kinematic chain. The vectors q = (q_s, q_f, q_v) are the mechanism outputs.
- r_1^j for j = 1, 2, 3, 4 are the linear joints which are the inputs of the mechanism.
- In the closed loop j, the rotation of the joint i is represented by θ_i^j.
- r_1^j and q_v are the only joints that are active while other joints are passive.

The open mechanism that results from cutting the z_5^j joints as presented in Fig. 4 is used to carry out the IGM.

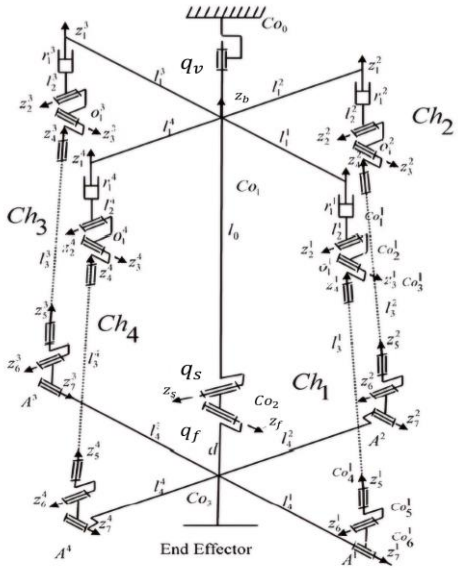

Fig. 4. HYDROïD's ankle mechanism kinematic structure.

The IGM of the hybrid ankle mechanism is presented as follows:

For Loop_1 and Loop_3, results of the IGM for the rotational joints are shown in Eqs. (11) and (12).

$$\theta_3^j = \arcsin\left(\frac{d \cdot S_{q_v}}{l_3^j}\right) \tag{11}$$

$$\theta_2^j = \arcsin\left(\frac{\left(d \cdot S_{q_s} \cdot C_{q_f}\right) + \left(l_4^j \cdot C_{q_s}\right) - l_1^j}{l_3^j \cdot C_3^j}\right) \tag{12}$$

The results of the IGM for the linear active joints are shown in Eq. (13).

$$r_1^j = l_0 - l_2^j - l_4^j \cdot S_{q_s} + d \cdot C_{q_s} C_{q_f} - l_3^j C_2^j C_3^j \tag{13}$$

In a similar manner, the relations for Loop_2 and Loop_4 are established. The IGM results for the rotational joints are established in Eqs. (14) and (15).

$$\theta_3^j = \arcsin\left(\frac{d \cdot S_{q_v} + l_4^j \cdot C_{q_f} - l_1^j}{l_3^j}\right) \tag{14}$$

$$\theta_2^j = \arcsin\left(\frac{d \cdot S_{q_s} \cdot C_{q_f} - l_4^j \cdot S_{q_s} S_{q_f}}{l_3^j \cdot C_3^j}\right) \tag{15}$$

Therefore, the IGM result of the active joints in these loops is shown in Eq. (16).

$$r_1^j = l_0 - l_2^j - l_4^j \cdot C_{q_s} \cdot S_{q_f} + d \cdot C_{q_s} C_{q_f} - l_3^j C_2^j C_3^j \tag{16}$$

2) Inverse Kinematic model

The IKM model is used for robot simulation, motion planning, and robot control, which will be incorporated into the control architecture.

The vector $\dot{X}_a = (\dot{\theta}_r, \dot{\theta}_p, \dot{\theta}_y)$ represent the angular velocities of the rotations around the roll, yaw, and pitch axes, and the vector $\dot{r}_a = (\dot{r}_1^1, \dot{r}_1^2, \dot{r}_1^3, \dot{r}_1^4, \dot{r}_1^5, q_1)^t$ represents the velocities of the active joint determined by the IKM.

The kinematic composition formula results in Eq. (17) which represents the j^{th} closed loop kinematics. $T_{c_{S_p/S_b}}$ is represents the foot's kinematic wrench in reference to the parallel mechanism's base, $T^j_{c_{S_1/S_b}}$ is the j^{th} linear actuator kinematic wrench in reference to the parallel mechanism's base, $T^j_{c_{S_2/S_1}}$ is the j^{th} cable kinematic wrench in reference to j^{th} linear actuator, and $T^j_{c_{S_p/S_2}}$ is the foot's kinematic wrench in reference to the j^{th} cable.

$$T_{c_{S_p/S_b}} = T^j_{c_{S_1/S_b}} + T^j_{c_{S_2/S_1}} + T^j_{c_{S_p/S_2}} \tag{17}$$

Eq. (18) shows this relation represented with individual screws. With $\dot{\theta}_i^j$ being the derivative of θ_i^j and $\$_k^j$ being the k^{th} joint kinematic screw in the j^{th} closed loop.

$$T_{c_{S_p/S_b}} = \dot{r}_1^j \$_1^j + \dot{\theta}_2^j \$_2^j + \dot{\theta}_3^j \$_3^j + \dot{\theta}_4^j \$_4^j + \dot{\theta}_5^j \$_5^j + \dot{\theta}_6^j \$_6^j + \dot{\theta}_7^j \tag{18}$$

$\$_1^{R_j}$ is the reciprocal screw of the active variable r_1^j in the closed chain, and it satisfies the condition shown in Eq. (19).

$$\$_i^j \$_1^{R_j} = 0, i = 2, 3, 4, 5, 6, 7 \tag{19}$$

The working reference frame is R_1^j which is placed on O_1^j of the closed chain and parallel to R_b. Eq. (20) shows the coordinates of the j^{th} cable projected in the working reference frame.

$$A^j O_1^j = \left[U_j\ V_j\ W_j\right]_{R_1^j} \tag{20}$$

The desired screw is resulted by solving Eq. (19) as shown in Eq. (21).

$$\$_1^{R_j} = \frac{1}{\sqrt{U_j^2 + V_j^2 + W_j^2}} \left[U_j\ V_j\ W_j\ 0\ 0\ 0\right] \tag{21}$$

Multiplying Eq. (21) by Eq. (18) and selecting A^j, as the j^{th} closed loop working point as in Eq. (17), the resultant relation is presented in Eq. (22).

$$\$_1^{R_j} T_{c_{S_p/S_b}}(A^j) = \dot{r}_1^j \$_1^j \$_1^{R_j} \tag{22}$$

The foot's kinematic wrench in reference to the base S_b and projected on R_b is shown in Eq. (23).

$$T_{c_{S_p/S_b}}(A_0) = \dot{q}_s z_s + \dot{q}_f z_f = \left[\dot{q}_f C_{q_s} - \dot{q}_s\ \dot{q}_f S_{q_s}\right] \tag{23}$$

Due to the hybrid mechanism, q_v is the primer rotational joint that is independent of the other joints q_s and q_f. Hence, placing Eqs. (21) and (23) in Eq. (22) for the four closed chains, results in the Eq. (24) that shows the kinematic model.

$$\begin{bmatrix} L_4^1 W^1 & L_4^1 S_{qs} V^1 & 0 \\ 0 & L_4^2 (C_{qs} W^2 - S_{qs} U^2) & 0 \\ -L_4^3 W^3 & -L_4^3 S_{qs} V^3 & 0 \\ 0 & L_4^4 (C_{qs} W^4 - S_{qs} 4) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_s \\ \dot{q}_f \\ \dot{q}_v \end{bmatrix} = \begin{bmatrix} W^1 & 0 & 0 & 0 & 0 \\ 0 & W^2 & 0 & 0 & 0 \\ 0 & 0 & W^3 & 0 & 0 \\ 0 & 0 & 0 & W^4 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{r}_1^1 \\ \dot{r}_1^2 \\ \dot{r}_1^3 \\ \dot{r}_1^4 \\ \dot{q}_v \end{bmatrix} \tag{24}$$

Eq. (24) has the classical matrix form:

$$A \cdot \dot{q} = B \cdot \dot{r} \tag{25}$$

Moreover, Eq. (26) shows the equation of the global kinematic variable $\dot{X}$.

$$\dot{X} = \dot{q}_s Z_b + \dot{q}_v Z_v + \dot{q}_f Z_f = \begin{bmatrix} \dot{q}_s S_{q_v} + \dot{q}_f C_{q_v} C_{q_s} \\ -\dot{q}_s C_{q_v} + \dot{q}_f S_{q_v} C_{q_s} \\ \dot{q}_v + \dot{q}_f S_{q_s} \end{bmatrix} \begin{bmatrix} 0 & C_{q_v} C_{q_s} & S_{q_v} \\ 0 & S_{q_v} C_{q_s} & -C_{q_v} \\ 1 & S_{q_s} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_v \\ \dot{q}_f \\ \dot{q}_s \end{bmatrix} \tag{26}$$

Eq. (26) can be reformulated and written as in Eq. (27).

$$\dot{X} = D \cdot \dot{q} \tag{27}$$

The kinematic model of the ankle mechanism is the result of replacing Eq. (27) in Eq. (25) as shown in Eq. (28).

$$A \cdot D^{-1} \cdot \dot{X} = B \cdot \dot{r} \tag{28}$$

Based on the IGM and the IKM a control of the ankle mechanism can be implemented as shown in Fig. 5.

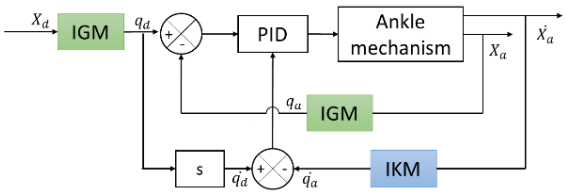

Fig. 5. Control of the hybrid ankle mechanism using the IGM and IKM.

A code on Matlab is developed to estimate the time and number of executions of each algorithm to implement an efficient control structure. The results are presented in Table 2. The selected microcontroller at the joint level is expected to be able to handle all the required computations. Hence, the selected MCU has the following characteristics: i) CPU speed of 480 MHz, ii) 1027 DMIPS, so 480 million cycles are executed per second, and iii) 1027 million instructions are performed per second.

Table 2. Estimated computational cost for the control of the ankle mechanism

Algorithm	Execution Time (ms)	Number of Operations
IGM	5.3	52
IKM	15.4	150
PID	4.6	15

IV. Proposed Real-Time Control Architecture

A distributed and real-time architecture is the proposed control architecture for HYDROïD. Moreover, this architecture is based on open-source software and EtherCAT

communication protocol.

At the joint level, a customized electronic board is developed. This board is responsible for monitoring the different integrated sensors in the actuator, including force position, temperature, and pressure, and for controlling the actuator. FreeRTOS, an open-source and real-time firmware designed for embedded systems, is integrated into the board. FreeRTOS is suitable for applications requiring hard or soft real-time capabilities, and it is open-source and, hence, cost-effective. Different threads have been developed for the control of the actuator [15].

A distributed approach is adopted for the control of the humanoid. One board controls one mechanism, as shown in Fig. 6, ensuring the robot's human-like appearance without bulky cables and reducing complexity. The communication interface between the different boards is selected to be the EtherCAT communication protocol with 100 Mbits/s bandwidth. The advantage of EtherCAT is that it has high bandwidth and enables soft and hard real-time communication. This communication protocol follows the Master-Slave topology, with the central PC identified as the EtherCAT master and the joint controllers identified as the EtherCAT slave devices.

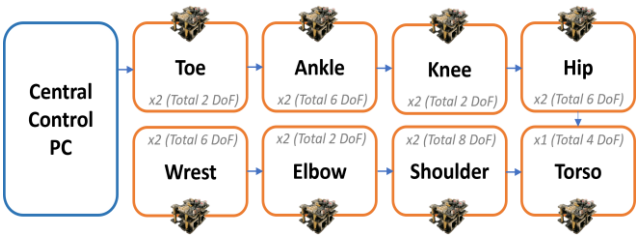

Fig. 6. Joint controllers distributions in HYDROïD, one controller for each mechanism with a total of 15 controllers.

A real-time operating system is developed on the master PC to control the robot. Ubuntu was selected, and an RT-preemptible kernel was implemented to guarantee the real-time characteristics. Middleware is used to reduce complexity, simplify the software design, and enhance the software application [16]. OROCOS, a real-time middleware developed for robotic applications, is utilized for hardware abstraction from the environment. OROCOS supports the Real-Time Toolkit (RTT) library, which provides an infrastructure to build real-time robotics applications [17]. Moreover, it is component-based, and ports holding a message are defined to connect the components. Messages are transmitted with a defined frequency while respecting real-time restrictions. Therefore, precise control is created for robotic systems. OROCOS middleware uses the RTT- Simple Open Master EtherCAT (SOEM) library to manage the communication between the master and the slaves, ensuring data transmission within the specified and predefined time.

On OROCOS, the electronic EtherCAT slave board is represented by a driver, and messages are developed based on the board's frame structure. Moreover, a component with a period of 5 kHz is implemented for the robot's high-level control. Robot Operating System (ROS) [18] is integrated with OROCOS, taking advantage of the numerous packages available. Hence, the proposed control architecture enables the control of the hydraulic actuator with ROS. Exchanging data with ROS is safe and doesn't affect the real-time performance in OROCOS because it has an integrated interface for ROS. Besides, the robot's motion planning and a GUI that enables the direct control of the actuator are implemented on ROS. The software architecture that is implemented for the control of HYDROïD is represented in Fig. 7.

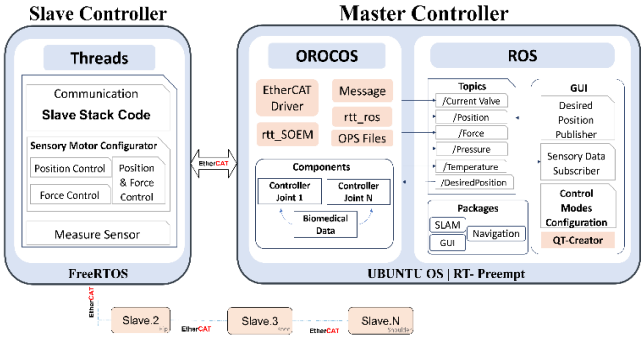

Fig. 7. Developed control architecture layers for controlling HYDROïD.

V. Experimental Validation

The dynamic behavior analysis, utilizing transfer functions and the inverse Laplace transform, was performed using MatLab-Simulink. This analysis is based on the rotary hydraulic motor driven by a servo valve, resulting in a second-order proportional transfer function. Multiple corrections were implemented to improve the servo valve's performance. Table 3 outlines the specific transfer functions and parameters applied during this process

Table 3. Corrections applied to the servo valve

Correction Type	Correction Function	Parameter (s)
RC Correction	$H(s) = \frac{1}{Ts+1}$	T = 0.006
Anticipation Correction	$H(s) = \frac{T_d s+1}{T_i s+1}$	T_d = 0.01 T_i = 0.001
Inertial Correction	$H(s) = \frac{T_d s+1}{T_i s+1}$	T_d = 0.0001 T_i = 0.001
1st Order Inertial Correction	$H(s) = \frac{1}{T_i s+1}$	T_i = 0.02
2nd Order Inertial Correction	$H(s) = \frac{1}{T_{i_2} s^2 + T_{i_1} s+1}$	T_{i1} = 0.0005 T_{i2} = 0.02

Fig. 8 shows the enhanced results of the servo valve using the RC correction. The servo valve performance without any correction is shown in the left graph "a", while that with the RC correction is presented in the right graph "b". The RC correction resulted in reduced overshoot and increased response time.

Fig. 9, on the other hand, shows the effects of implementing anticipation and inertial correction. The effect of the anticipation correction is shown in the graph on the left "a", while the graph on the right "b" represents the effects of the inertial corrections on the performance of the servo valve. The results show that the first-order inertial correction decreases transient time and overshoot, but the response time increases, which results in a slower system. While anticipation correction resulted in an increased overshoot and reduced response time. The transient time increased, ensuring promptness while maintaining accuracy. However, at the stability limit

The enhancements of the first and second-order inertial corrections are represented in Fig. 10. The left graph "a" is

the results of the first-order inertial correction, while the right graph "b" is the results of the second-order corrections. The second-order inertial correction resulted in a delayed response time and increased transient time, but the operation is stable.

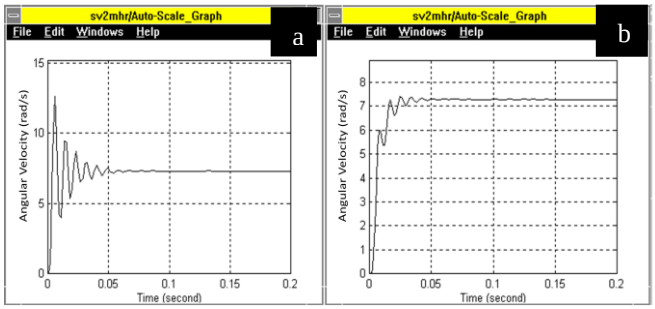

Fig. 8. Results of the simulation of the rotary hydraulic actuator driven by a servo valve, the results of the servo valve simulation without any correction are represented in graph a, and the results of the RC correction are represented in graph b.

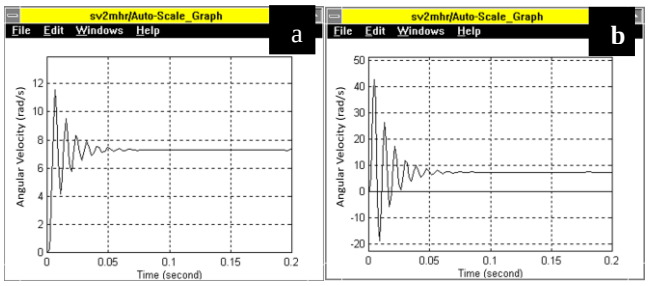

Fig. 9. Results of anticipation and inertial corrections applied to the rotary hydraulic actuator driven by a servo valve, graph A shows the inertial correction, while graph B shows the anticipation correction.

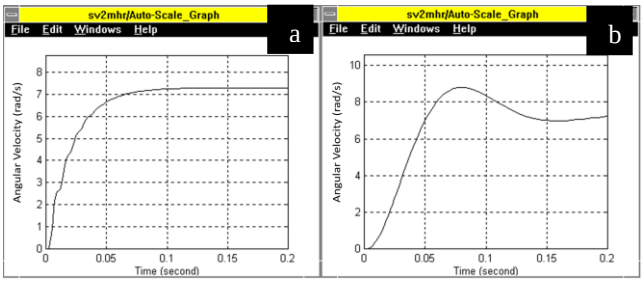

Fig. 10. Results of inertial correction applied to the rotary hydraulic actuator driven by a servo valve. Graph a shows the results of the first-order inertial correction, while graph b shows the second-order inertial correction results.

The proposed control architecture will be operated on an update rate of 5 kHz. The real-time capability of the developed system is tested on a high level by cyclic test. The results show the latency using the rt- preempt kernel didn't exceed 58 µs. The network latency on the low level was tested by connecting 8 boards, and the maximum latency didn't exceed 167 µs. Hence, the system didn't exceed the desired and expected cycle time of 200 µs (5 kHz).

To assess the performance of the overall control architecture, four mechanisms of HYDROïD were operated on: the right and left knee and the left and right hip. Therefore, gait control was applied to control these mechanisms. The control algorithm was implemented on the high-level controller, which sends the desired position to the low-level controller, which in return is responsible for controlling the servo valve current. Fig. 11 demonstrates the robot's functioning, showcasing human-like motion.

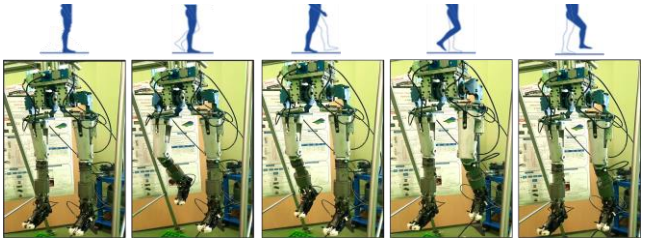

Fig. 11. HYDROïD in operation with operating the hip and knee using the proposed control architecture.

VI. Conclusion

Electro-hydraulic actuators characterized by their force-to-volume and force-to-volume ratios are suitable for robotics applications requiring high forces. This paper presents a real-time control structure for an electrohydraulic humanoid, HYDROïD. A model of a single joint of the HYDROïD system, comprising a rotary hydraulic actuator driven by a servo valve, was developed and analyzed to enhance the system's performance and to facilitate the development of an efficient control architecture with reduced complexity. Additionally, the IGM and the IKM were calculated to estimate the computational cost to implement a control loop.

The development of the control architecture necessitates a consideration of the software's adaptability. In other words, adding sensors and actuators to the system should be straightforward without encountering significant complications. Furthermore, the software should guarantee deterministic properties, thereby ensuring the safe interaction of the robot with its environment in real time. In addition, it is necessary to ensure a high bandwidth for the high-level and low-level controllers.

This paper presents the control architecture developed for HYDROïD. EtherCAT communication protocol is selected for the proposed architecture. A customized electronic board, identified as an EtherCAT slave, is designed to monitor the sensors and drive the servo valve with FreeRTOS firmware implemented. These slaves are distributed among the mechanisms with a total of 15 boards for controlling the whole robot. On the master PC, Ubuntu, with a preemptible kernel, is the operating system implemented to guarantee deterministic property. A hybrid middleware that bridges OROCOS and ROS is developed on the master PC to monitor and control the robot.

The results show the enhancement of the servo valve performance upon applying different corrections to the system. Besides, the developed control architecture was tested on the robot's hip and ankle, achieving a human-like motion.

As a future work, Ubuntu 22.04, the real-time capable version of Ubuntu, will be utilized. Also, ROS2 [19], will be integrated into the system. Besides, the entire robot will be controlled, and the robot's motion planning will be executed.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Methodology, M.G.; data curation, M.G.; conceptualization, M.G. and S.A.; software, M.G.; formal analysis, M.G.; validation, M.G.; resources, M.G. and

N.A.O.; investigation, M.G.; writing original draft, M.G.; writing, reviewing and editing, S.A., N.A.O. and A.O.; visualization, M.G. and A.O.; project administration, S.A.; supervision, S.A. and N.A.O.; funding acquisition, S.A. All authors have read and agreed to the published version of the manuscript.

FUNDING

This research is funded by KALYSTA Actuation Company and the industrial excellence chair between the University of Evry and Kalysta Actuation.

REFERENCES

[1] A. A. H. Ibrahim, A. Ammounah, S. Alfayad, S. Tliba, F. B. Ouezdou, and S. Delaplace, "Hydraulic robotic leg for HYDROïD robot: Modeling and control," *Journal of Robotics and Mechatronics*, vol. 34, no. 3, pp. 576–587, 2022.

[2] S. Alfayad, M. Kardofaki, and M. Sleiman. (May 2022). US20220145868A1-Hydraulic actuator with overpressure compensation. [Online]. Available: https://patents.google.com/patent/US20220145868A1/en?inventor=Samer+Alfayad

[3] S. Fischmeister and P. Lam, "Time-aware instrumentation of embedded software," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 4, pp. 652–663, Nov. 2010. doi: 10.1109/TII.2010.2068304

[4] I.-W. Park, J.-Y. Kim, J. Lee, and J.-H. Oh, "Mechanical design of humanoid robot platform KHR-3 (KAIST Humanoid Robot 3: HUBO)," in *Proc. 5th IEEE-RAS International Conference on Humanoid Robots*, 2005, pp. 321–326. doi: 10.1109/ICHR.2005.1573587

[5] F. Nori, S. Traversaro, J. Eljaik, F. Romano, A. del Prete, and D. Pucci, "iCub whole-body control through force regulation on rigid non-coplanar contacts," *Front. Robot. AI*, vol. 2, Mar. 2015. doi: 10.3389/frobt.2015.00006

[6] S. Karumanchi *et al*., "Team RoboSimian: Semi-autonomous mobile manipulation at the 2015 DARPA robotics challenge finals," *Journal of Field Robotics*, vol. 34, no. 2, pp. 305–332, 2017. doi: 10.1002/rob.21676

[7] H. Kaminaga *et al*., "Mechanism and control of whole-body electro-hydrostatic actuator driven humanoid robot hydra," in *Proc. 2016 International Symposium on Experimental Robotics*, D. Kulić, Y. Nakamura, O. Khatib, and G. Venture, Eds., Cham: Springer International Publishing, 2017, pp. 656–665. doi: 10.1007/978-3-319-50115-4_57

[8] F. Sygulla *et al.*, "An EtherCAT-based real-time control system architecture for humanoid robots," in *Proc. 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, Aug. 2018, pp. 483–490. doi: 10.1109/COASE.2018.8560532

[9] M. Sleiman, K. Khalil, A. Olaru, and S. AlFayad, "Design and validation of new methodology for hydraulic passage integration in carbon composite mechanisms," *Applied Sciences*, vol. 14, no. 11, 4378, 2024.

[10] S. Alfayad, F. B. Ouezdou, and F. Namoun. (Aug. 04, 2011). US20110185837A1-Humanoid robot implementing a spherical hinge with coupled actuators. [Online]. Available: https://patents.google.com/patent/US20110185837A1/en

[11] S. Alfayad, A. M. Tayba, F. B. Ouezdou, and F. Namoun, "Kinematic synthesis and modeling of a three degrees-of-freedom hybrid mechanism for shoulder and hip modules of humanoid robots," *Journal of Mechanisms and Robotics*, vol. 8, 041017, Apr. 2016. doi: 10.1115/1.4033157

[12] A. Abdellatif Hamed IBRAHIM, S. Alfayad, A.-C. Hildebrandt, F. B. Ouezdou, N. Mechbal, and Y. Zweiri, "Development of a new hydraulic ankle for HYDROïD humanoid robot," *Journal of Intelligent & Robotic Systems*, vol. 92, Oct. 2018. doi: 10.1007/s10846-017-0750-z

[13] S. Alfayad, M. Kardofaki, M. Sleiman, and R. Arlot. (May 25, 2023). WO2023088972A1-Cylinder with integral position sensor. [Online]. Available: https://patents.google.com/patent/WO2023088972A1/en?inventor=Samer+Alfayad&sort=new

[14] S. Alfayad, F. B. Ouezdou, and F. Namoun, "New 3-DOFs hybrid mechanism for ankle and wrist of humanoid robot: Modeling, simulation, and experiments," *ASME. J. Mech. Des.*, vol. 133, no. 2, 021005, 2011.

[15] M. Ghandour, S. Jleilaty, N. A. Oufroukh, S. Olaru, and S. Alfayad, "Real-time EtherCAT-based control architecture for electro-hydraulic humanoid," *Mathematics*, vol. 12, no. 9, 1405, 2024.

[16] A. Elkady and T. Sobh, "Robotics middleware: A comprehensive literature survey and attribute-based bibliography," *Journal of Robotics*, vol. 2012, May 2012. doi: 10.1155/2012/959013

[17] H. Bruyninckx, "Open robot control software: The OROCOS project," in *Proc. the 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, May 2001, vol. 3, pp. 2523–2528. doi: 10.1109/ROBOT.2001.933002

[18] M. Quigley, K. Conley, B. Gerkey, J. Faust *et al*., "ROS: An open-source robot operating system," in *Proc. ICRA Workshop on Open Source Software*, 2009. vol. 3.

[19] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, May 2022. doi: 10.1126/scirobotics.abm6074