Κεφάλαιο 3

Λογισμική Αρχιτεκτονική και Λειτουργικές Απαιτήσεις του Ρομποτικού Βραχίονα

3.1 Εισαγωγή

Στο παρόν κεφάλαιο παρουσιάζεται η λογισμική αρχιτεκτονική του ρομποτικού βραχίονα και η αντιστοίχισή της με τις λειτουργικές απαιτήσεις του συστήματος.
Ο σχεδιασμός του λογισμικού βασίζεται στο **ROS 2 (Robot Operating System 2)** και ακολουθεί σύγχρονες αρχές ανάπτυξης ρομποτικών συστημάτων, με έμφαση στη **δομική αρθρωτότητα (modularity)**, στον **σαφή διαχωρισμό αρμοδιοτήτων** και στη **δυνατότητα επεκτασιμότητας για ερευνητική χρήση**.

Η αρχιτεκτονική υλοποιείται σε κατανεμημένο περιβάλλον, με διαχωρισμό μεταξύ υπολογιστικών μονάδων υψηλών απαιτήσεων και υποσυστημάτων χαμηλού επιπέδου ελέγχου.

3.2 Συνολική Αρχιτεκτονική Συστήματος

Το σύστημα οργανώνεται σε τρία διακριτά επίπεδα:

1. **Επίπεδο Υλικού (Hardware Layer)**
   Περιλαμβάνει τον ρομποτικό βραχίονα, τους servo motors, τον οδηγό PWM PCA9685, τον Raspberry Pi 4 και τους αισθητήρες (π.χ. depth camera).
2. **Επίπεδο Χαμηλού Ελέγχου (Low-Level Control Layer)**
   Υλοποιείται στον Raspberry Pi 4 και αφορά τη μετατροπή εντολών ελέγχου αρθρώσεων σε φυσική ενεργοποίηση των actuators.
3. **Επίπεδο Υψηλού Ελέγχου και Αντίληψης (High-Level Control & Perception Layer)**
   Εκτελείται στον κεντρικό υπολογιστή (Laptop) και περιλαμβάνει σχεδιασμό κίνησης, αντίληψη περιβάλλοντος, οπτικοποίηση και εκτέλεση εργασιών.

Η επικοινωνία μεταξύ των επιπέδων πραγματοποιείται μέσω του **DDS middleware** του ROS 2, επιτρέποντας αξιόπιστη και διαφανή ανταλλαγή δεδομένων.

3.3 Λειτουργικές Απαιτήσεις Συστήματος

3.3.1 FR-1: Περιγραφή και χωρική αναπαράσταση του ρομπότ

Το σύστημα πρέπει να διατηρεί συνεπή γεωμετρική και κινηματική αναπαράσταση του ρομποτικού βραχίονα σε πραγματικό χρόνο.

Η απαίτηση αυτή υλοποιείται μέσω παραμετρικής περιγραφής του ρομπότ σε URDF, με χρήση της γλώσσας Xacro, και μέσω της δημοσίευσης του δέντρου μετασχηματισμών (TF tree). Με τον τρόπο αυτό εξασφαλίζεται κοινό σύστημα αναφοράς για όλα τα υποσυστήματα (σχεδιασμός κίνησης, αντίληψη, οπτικοποίηση).

3.3.2 FR-2: Χαμηλού επιπέδου έλεγχος αρθρώσεων

Το σύστημα πρέπει να μεταφράζει εντολές θέσης αρθρώσεων σε πραγματική κίνηση των servo motors με σταθερότητα και χρονική ομαλότητα.

Η απαίτηση αυτή καλύπτεται μέσω του framework **ros2_control**, το οποίο λειτουργεί ως επίπεδο αφαίρεσης μεταξύ του αλγοριθμικού ελέγχου και του φυσικού hardware.
Ο Raspberry Pi 4 αναλαμβάνει τον ρόλο του hardware interface, οδηγώντας τον PCA9685 για την παραγωγή σταθερού PWM σήματος προς τους servo motors.

3.3.3 FR-3: Σχεδιασμός κίνησης με αποφυγή συγκρούσεων

Το σύστημα πρέπει να είναι ικανό να σχεδιάζει τροχιές κίνησης λαμβάνοντας υπόψη κινηματικούς περιορισμούς, όρια αρθρώσεων και πιθανά εμπόδια στο περιβάλλον.

Η απαίτηση αυτή υλοποιείται μέσω του **MoveIt 2**, το οποίο παρέχει αλγορίθμους σχεδιασμού κίνησης, λύτες κινηματικής και μηχανισμό διαχείρισης σκηνής (planning scene) για έλεγχο συγκρούσεων.

3.3.4 FR-4: Αντίληψη περιβάλλοντος (Perception)

Το σύστημα πρέπει να είναι ικανό να αντιλαμβάνεται αντικείμενα στον χώρο εργασίας και να εκτιμά τη χωρική τους θέση.

Η αντίληψη βασίζεται σε δεδομένα RGB-D από αισθητήρα βάθους, τα οποία μετασχηματίζονται στο κοινό σύστημα αναφοράς του ρομπότ. Τα δεδομένα αυτά χρησιμοποιούνται για την εκτίμηση της τρισδιάστατης θέσης και του προσανατολισμού στόχων, επιτρέποντας λειτουργίες **vision-based manipulation**.

3.3.5 FR-5: Εκτέλεση σύνθετων εργασιών (Pick & Place)

Το σύστημα πρέπει να υποστηρίζει την εκτέλεση σύνθετων εργασιών υψηλού επιπέδου, όπως η διαδικασία pick-and-place.

Η λειτουργία αυτή υλοποιείται μέσω task-level κόμβου, ο οποίος οργανώνει τη λογική ακολουθία ενεργειών (προσέγγιση αντικειμένου, σύλληψη, μεταφορά και απόθεση), αξιοποιώντας τις υπηρεσίες σχεδιασμού και εκτέλεσης τροχιών του MoveIt.

3.3.6 FR-6: Προσομοίωση και ψηφιακό δίδυμο

Το σύστημα πρέπει να υποστηρίζει πλήρη προσομοίωση της λειτουργίας του χωρίς την ανάγκη φυσικού hardware.

Η απαίτηση αυτή καλύπτεται μέσω της ενσωμάτωσης περιβάλλοντος προσομοίωσης (Gazebo) και προσομοιωμένων controllers του ros2_control, επιτρέποντας την ανάπτυξη και δοκιμή του λογισμικού σε ασφαλές και ελεγχόμενο περιβάλλον.

3.3.7 FR-7: Καταγραφή και ανάλυση δεδομένων

Το σύστημα πρέπει να επιτρέπει την καταγραφή και αναπαραγωγή δεδομένων για σκοπούς debugging και ερευνητικής ανάλυσης.

Η λειτουργία αυτή υλοποιείται μέσω του εργαλείου **rosbag2**, το οποίο επιτρέπει την αποθήκευση δεδομένων αισθητήρων, καταστάσεων αρθρώσεων και τροχιών κίνησης.

3.4 Συμπεράσματα Κεφαλαίου

Το παρόν κεφάλαιο παρουσίασε τη λογισμική αρχιτεκτονική του ρομποτικού βραχίονα και τη σύνδεσή της με τις λειτουργικές απαιτήσεις του συστήματος.

Η υιοθετούμενη αρχιτεκτονική εξασφαλίζει υψηλό βαθμό αρθρωτότητας, επεκτασιμότητας και επαναχρησιμοποίησης, καθιστώντας το σύστημα κατάλληλο τόσο για εκπαιδευτική όσο και για ερευνητική χρήση.

ROS 2 Software Stack – Libraries, Installation & Purpose

Robot Model & TF

Library / Package	Install Location	Purpose
xacro	Laptop, Raspberry Pi	Parametric URDF description using macros.
robot_state_publisher	Laptop, Raspberry Pi	Publishes TF tree from joint states.
joint_state_publisher_gui	Laptop	GUI tool for manual joint testing.
tf2_ros	Laptop, Raspberry Pi	Coordinate frame transformations.
tf2_tools	Laptop	TF tree debugging tools.

Low-Level Servo Actuation (PCA9685)

Library / Package	Install Location	Purpose
i2c-tools	Raspberry Pi	Low-level I2C diagnostics and bus scan.
smbus2	Raspberry Pi	Python interface for I2C communication.
adafruit-blinka	Raspberry Pi	CircuitPython hardware abstraction for Linux SBCs.
adafruit-circuitpython-pca9685	Raspberry Pi	Stable 12-bit PWM generation for servo motors.
numpy	Raspberry Pi	Math utilities, filtering, angle-to-PWM mapping.
arm_hardware (custom)	Raspberry Pi	ROS 2 node translating joint commands to PWM signals.

ros2_control & Controllers

Library / Package	Install Location	Purpose
ros2_control	Raspberry Pi	Hardware abstraction and controller lifecycle framework.
ros2_controllers	Raspberry Pi	Standard joint, trajectory and state controllers.
joint_trajectory_controller	Raspberry Pi	Executes smooth joint trajectories.
joint_state_broadcaster	Raspberry Pi	Publishes joint states to

		ROS graph.
arm_ros2_control (custom)	Raspberry Pi	Hardware interface plugin for PCA9685-based servos.

Motion Planning (MoveIt 2)

Library / Package	Install Location	Purpose
moveit_ros_move_group	Laptop	Central motion planning and execution node.
moveit_planners	Laptop	Sampling-based planners (OMPL).
moveit_kinematics	Laptop	Forward and inverse kinematics solvers.
moveit_servo	Laptop	Real-time Cartesian control of end-effector.
moveit_task_constructor	Laptop (optional)	High-level composition of manipulation tasks.

Perception (Vision & Depth)

Library / Package	Install Location	Purpose
Depth camera driver (e.g. realsense2_camera)	Raspberry Pi	RGB-D acquisition and point cloud publishing.
image_transport	Laptop, Raspberry Pi	Efficient transport of image topics.
tf2_sensor_msgs	Laptop, Raspberry Pi	Transform sensor data between coordinate frames.
pcl_ros	Laptop, Raspberry Pi (optional)	Point cloud processing and filtering.
python3-opencv	Laptop, Raspberry Pi	Image processing and feature extraction.
arm_perception (custom)	Laptop / Raspberry Pi	Object detection and 3D pose estimation.

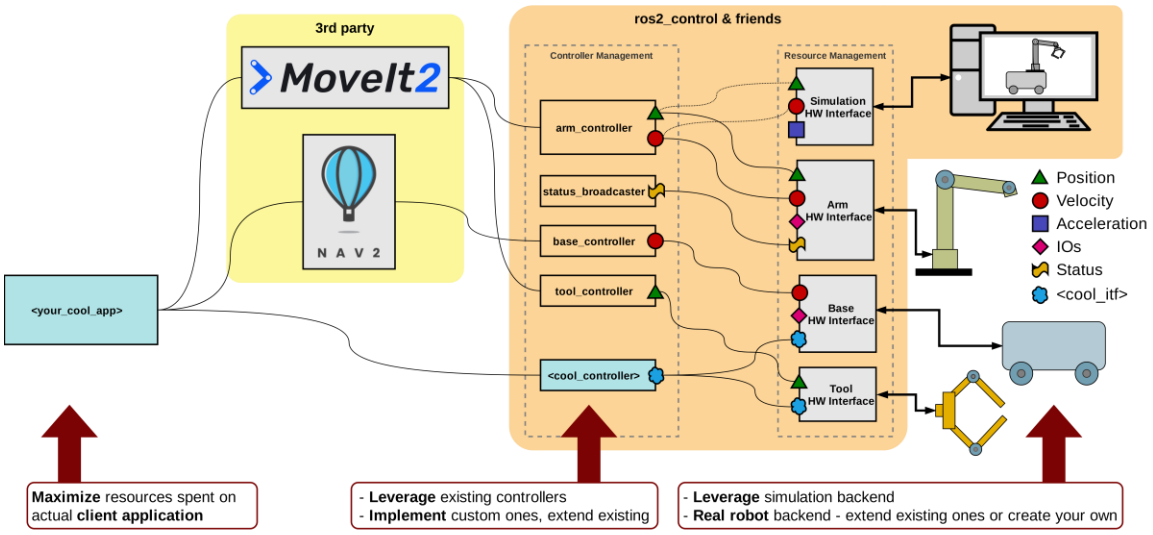
Simulation & Digital Twin

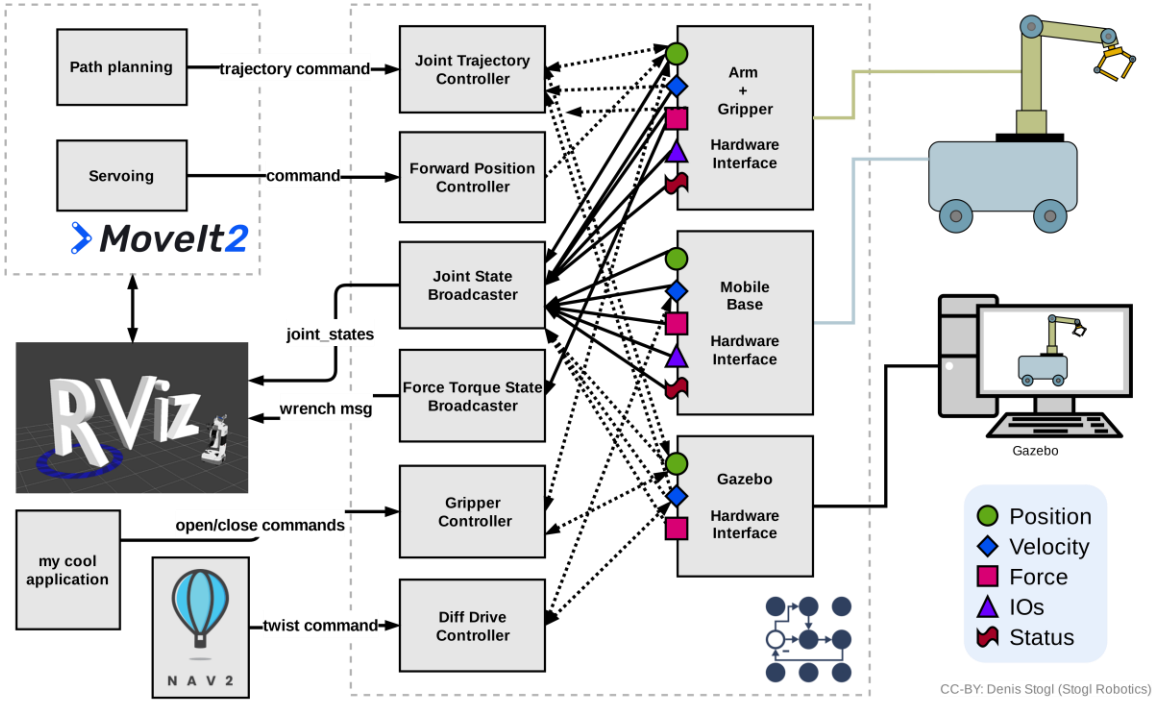
Library / Package	Install Location	Purpose
gazebo_ros_pkgs / ros_gz_*	Laptop	Physics-based robot simulation.
ros2_control simulation plugins	Laptop	Simulated joint control using same controllers.
MoveIt + RViz (simulation)	Laptop	Planning and visualization without physical hardware.

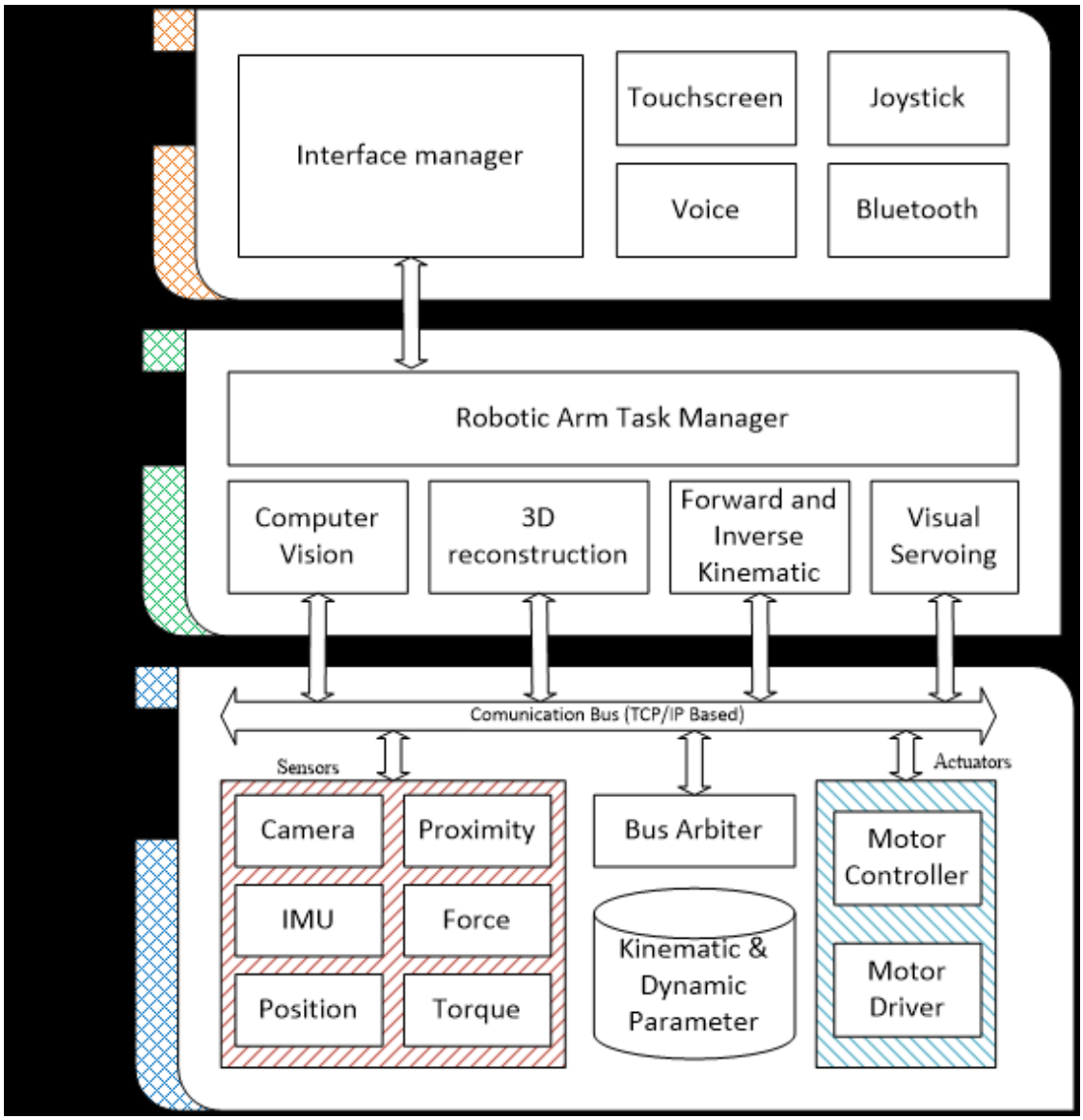
Logging & Analysis

Library / Package	Install Location	Purpose
rosbag2	Laptop, Raspberry Pi (optional)	Record and replay ROS topics.
ROS 2 CLI tools	Laptop, Raspberry Pi	Inspection and debugging of ROS graph.
foxglove_bridge	Laptop, Raspberry Pi	Live telemetry and

	(optional)	visualization dashboard.






Interface manager
Touchscreen
Joystick
Voice
Bluetooth
Robotic Arm Task Manager
Computer Vision
3D reconstruction
Forward and Inverse Kinematic
Visual Servoing
Comunication Bus (TCP/IP Based)
Sensors
Actuators
Camera
Proximity
IMU
Force
Position
Torque
Bus Arbiter
Kinematic & Dynamic Parameter
Motor Controller
Motor Driver