(42)

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST–2 EXAMINATION – 2026

B.Tech–VIII Semester (CSE)

COURSE CODE (CREDITS): 25B1WEC832 (2) **MAX. MARKS:** 25

COURSENAME: Intelligent Robotics and Autonomous Systems

COURSE INSTRUCTORS: Dr. Vikas Baghel **MAX. TIME:** 1 Hour 30 Min

Note: (a) All questions are compulsory.

(b) The candidate is allowed to make suitable numeric assumptions wherever required for solving problems.

(c) Use of calculators is allowed.

Q.No	Question	CO	Marks
Q1	In ROS2-based robot modeling and simulation, explain (i) the purpose of URDF/XACRO tags `robot`, `link`, and `joint`, including required `visual`, `collision`, and `inertial` details for a wheel link and why URDF enforces a tree structure; and (ii) the difference between URDF and SDF, with the role of SDF tags `sdf`, `world`, `model`, `link`, and `plugin` for ROS2 topic bridging.	CO1	[5]
Q2	(a) A differential-drive robot has wheel radius $r = 0.1\,\text{m}$ and axle length $l = 0.5\,\text{m}$. If $\omega_r = 8\,\text{rad/s}$ and $\omega_l = 4\,\text{rad/s}$, compute the robot linear and angular velocities. And also write the planar kinematic model. (b) Using the same robot ($r = 0.1\,\text{m}$, $l = 0.5\,\text{m}$), determine inverse kinematics (wheel speeds) required for $v = 0.6\,\text{m/s}$ and $\omega = 0.8\,\text{rad/s}$. Also state whether the robot turns left or right.	CO1	[3] [3]
Q3	(a) For the nonlinear stochastic system $$x_k = f(x_{k-1}, u_k) + w_k, \qquad z_k = h(x_k) + v_k,$$ derive the EKF prediction equations for $\hat{x}_{k\|k-1}$ and $P_{k\|k-1}$, and define the Jacobians $F_k = \partial f/\partial x$ and $H_k = \partial h/\partial x$.	CO4	[3]

Q.No	Question	CO	Marks
	(b) In 2D occupancy-grid SLAM, a map cell has prior $p_0(c) = 0.5$ and receives three identical measurements with inverse sensor probability $p(c \mid z_t, x_t) = 0.7$. Using $$l_t(c) = l_{t-1}(c) + \log\frac{p(c \mid z_t, x_t)}{1 - p(c \mid z_t, x_t)} - l_0(c), \qquad p_t(c) = \frac{1}{1 + e^{-l_t(c)}},$$ compute the final occupancy probability after three updates.		[3]
Q4	Write a ROS 2 node (Python/C++) for obstacle avoidance using LiDAR data. (i)Subscribe to /scan (LaserScan) (ii) Detect obstacles within a threshold distance (iii) Publish velocity commands to /cmd_vel (Twist) (iv) Move forward if clear, else stop and turn (v) Proper node structure	CO3	[8]