

# Module 9: Pick and Place Robot

## Contents

<b>Module 9 - Pick &amp; Place Robot</b>	<b>5</b>
9.1.1 Role in Modern Manufacturing Automation . . . . .	5
9.1.2 Robot Architecture Comparison and Selection Criteria . . . . .	5
Cartesian (Gantry) Robots . . . . .	5
SCARA (Selective Compliance Assembly Robot Arm) . . . . .	6
Delta (Parallel Linkage) Robots . . . . .	7
9.1.3 Performance Specifications and Selection Framework . . . . .	8
Critical Performance Metrics . . . . .	8
Selection Decision Matrix . . . . .	9
9.1.4 Integration with CNC Manufacturing Systems . . . . .	9
9.1.5 Module Roadmap . . . . .	10
References . . . . .	11
<b>Module 9 - Pick &amp; Place Robot</b>	<b>12</b>
Maintenance Schedule . . . . .	12
Linear Motion System Maintenance . . . . .	12
Rotational Joint Maintenance . . . . .	13
End Effector Maintenance . . . . .	13
Vision System Maintenance . . . . .	13
Pneumatic System Maintenance . . . . .	14
Electrical System Maintenance . . . . .	14
Controller and Software Maintenance . . . . .	15
Calibration and Accuracy Verification . . . . .	15
Documentation . . . . .	15
Predictive Maintenance . . . . .	15
Spare Parts Recommendations . . . . .	16
References . . . . .	16
<b>Module 9 - Pick &amp; Place Robot</b>	<b>16</b>
Systematic Troubleshooting Approach . . . . .	16
Mechanical Issues . . . . .	17
Positioning Errors . . . . .	17
Excessive Vibration . . . . .	17
Binding or Sticking . . . . .	18
Gripper Issues . . . . .	18
Failed Picks (Vacuum) . . . . .	18

Failed Picks (Mechanical Gripper)	19
Dropped Parts	20
Vision System Issues	20
Part Not Found	20
Incorrect Position Reported	21
Slow Vision Processing	21
Electrical and Control Issues	21
Motor Does Not Move	21
Encoder Errors	22
Communication Faults	23
Software and Programming Issues	23
Program Faults or Crashes	23
Inconsistent Behavior	24
Safety System Issues	24
Unexpected Stops	24
Safety System Bypass Attempt	25
Emergency Procedures	25
Troubleshooting Tools	25
References	25
<b>Module 9 - Pick &amp; Place Robot</b>	<b>26</b>
Summary	26
Practical Implementation	27
Economic Considerations	27
Future Trends	27
Learning Pathways	28
Resources for Further Learning	28
Closing Thoughts	28
Next Steps	29
References	29
<b>Module 9 - Pick &amp; Place Robot</b>	<b>30</b>
9.2.1 Cartesian (Gantry) Robot Architecture	30
Structural Configuration	30
Forward Kinematics	30
Structural Analysis: Beam Deflection	30
Natural Frequency Analysis	31
Linear Motion System Selection	31
9.2.2 SCARA Robot Architecture	32
Kinematic Configuration	32
Forward Kinematics	32
Inverse Kinematics	32
Joint Torque Analysis	33
Harmonic Drive Gearboxes	33
Z-Axis Counterbalance	33
9.2.3 Delta Robot Parallel Kinematic Architecture	34
Structural Topology	34
Inverse Kinematics (Analytical Solution)	34

Forward Kinematics (Numerical Solution Required)	35
Workspace Geometry	35
Parallel Kinematic Advantages	35
Link Design for High-Speed Operation	35
9.2.4 Architecture Selection Framework	36
Performance Trade-offs	36
Application-Specific Recommendations	36
References	37
<b>Module 9 - Pick &amp; Place Robot</b>	<b>37</b>
Vacuum Grippers	37
Mechanical Grippers	38
Magnetic Grippers	38
Specialized End Effectors	38
Quick-Change Systems	39
Sensors	39
Design Guidelines	39
References	39
<b>Module 9 - Pick &amp; Place Robot</b>	<b>40</b>
Machine Vision Fundamentals	40
Camera Selection	40
Lens Selection	40
Lighting Techniques	41
Vision Processing	41
Common Vision Tasks	41
2D Vision System Example	42
3D Vision	42
Proximity Sensors	42
Force and Torque Sensors	43
Sensor Integration	43
Vision Software	43
Calibration and Accuracy	44
References	44
<b>Module 9 - Pick &amp; Place Robot</b>	<b>44</b>
Motion Controllers	44
Coordinate Systems	45
Kinematics Algorithms	45
Trajectory Planning	46
Velocity Profiling	46
Multi-Axis Coordination	46
Control Loops	46
Path Planning Strategies	47
Homing and Calibration	47
Motion Programming	47
Performance Optimization	48
Real-Time Control	48

References . . . . .	48
<b>Module 9 - Pick &amp; Place Robot</b>	<b>49</b>
Integration Architectures . . . . .	49
Communication Protocols . . . . .	49
Machine Door Control . . . . .	50
Fixture and Part Handling . . . . .	50
Workpiece Identification . . . . .	50
Multi-Machine Cells . . . . .	51
LinuxCNC Integration . . . . .	51
Safety Integration . . . . .	52
Conveyor Integration . . . . .	52
Automated Tool Changing . . . . .	52
Cell Controller Software . . . . .	52
Example Cell Configurations . . . . .	53
Cycle Time Optimization . . . . .	53
Data Collection and Traceability . . . . .	53
References . . . . .	53
<b>Module 9 - Pick &amp; Place Robot</b>	<b>54</b>
Programming Methods . . . . .	54
Program Structure . . . . .	55
Position Management . . . . .	56
Vision-Guided Programming . . . . .	56
Error Handling and Recovery . . . . .	57
Advanced Programming Techniques . . . . .	58
Simulation and Testing . . . . .	59
Program Optimization . . . . .	60
Documentation . . . . .	60
References . . . . .	61
<b>Module 9 - Pick &amp; Place Robot</b>	<b>61</b>
Applicable Standards . . . . .	61
Risk Assessment . . . . .	62
Safeguarding Methods . . . . .	62
Safety Functions . . . . .	63
Operating Modes . . . . .	64
Collaborative Operation . . . . .	64
Safety Circuit Design . . . . .	64
Warning Devices . . . . .	65
Validation and Testing . . . . .	65
Documentation . . . . .	65
Common Violations and Mistakes . . . . .	66
Best Practices . . . . .	66
References . . . . .	66
<b>Module 9 - Pick &amp; Place Robot</b>	<b>67</b>
Cycle Time Analysis . . . . .	67

Motion Optimization . . . . .	67
Gripper Optimization . . . . .	68
Vision Processing Optimization . . . . .	68
Controller and Software Optimization . . . . .	69
Multi-Robot and Parallel Operations . . . . .	69
Part Presentation Optimization . . . . .	69
Energy Efficiency . . . . .	70
Overall Equipment Effectiveness (OEE) . . . . .	70
Case Study: Optimization Example . . . . .	70
Benchmarking and Testing . . . . .	71
Continuous Improvement . . . . .	71
References . . . . .	71

## Module 9 - Pick & Place Robot

### 9.1.1 Role in Modern Manufacturing Automation

Pick and place robots represent the convergence of precision motion control, machine vision, and process integration—automating material handling tasks that constitute 30-50% of manufacturing cycle time across industries from electronics assembly (0.1-10g components at 200+ picks/minute) to automotive parts handling (5-100 kg castings at 5-15 cycles/minute). These systems eliminate repetitive manual labor while achieving positioning repeatability (+/-0.01-0.05 mm typical) impossible for human operators, enabling lights-out manufacturing and just-in-time production strategies that reduce inventory costs 40-60% compared to batch processing.

#### Economic Drivers:

1. **Labor cost reduction:** Single robot replaces 1.5-3 FTE operators (\$45k-65k/year each), achieving ROI in 1.5-3 years for 2-shift operation
2. **Cycle time reduction:** Automated part transfer typically 2-4× faster than manual handling (3-5 seconds robot vs. 8-15 seconds operator)
3. **Quality improvement:** Consistent placement eliminates human error (drop rates <0.01% vs. 0.1-0.5% manual), reducing scrap and rework costs
4. **Flexibility:** Programmable systems adapt to product changes within minutes (vs. hours/days for dedicated automation), supporting high-mix manufacturing

#### Market Context (2024):

- Global industrial robot installations: 3.1 million units (IFR data)
- Pick and place robots: 35-40% of industrial robot market share
- Annual growth rate: 12-15% (driven by e-commerce warehousing, electronics miniaturization)
- Average system cost: \$25k-150k (Cartesian) to \$180k-500k (SCARA/Delta with vision)

### 9.1.2 Robot Architecture Comparison and Selection Criteria

#### Cartesian (Gantry) Robots

##### Kinematic Structure:

Three orthogonal linear axes (X, Y, Z) define rectangular workspace with simple forward kinematics:

$$\mathbf{P}_{end-effector} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d_X \\ d_Y \\ d_Z \end{bmatrix}$$

where  $d_X, d_Y, d_Z$  are commanded axis displacements (no coordinate transformation required).

#### Performance Characteristics:

Parameter	Typical Range	Design Driver
<b>Workspace</b>	500×500×300 mm to 6,000×3,000×1,500 mm	Application footprint
<b>Repeatability</b>	+/-0.02-0.05 mm	Mechanical stiffness, encoder resolution
<b>Payload</b>	1-100+ kg	Beam deflection, motor sizing
<b>Speed</b>	3-10 m/s linear, 2-5 m/s <sup>2</sup> acceleration	Belt/screw drive limits, vibration control
<b>Cycle time</b>	2-5 seconds typical	Path length + acceleration profile

**Advantages:** - **Rigid structure:** Moment arm constant across workspace (vs. SCARA cantilever), enabling heavy payloads - **Scalable workspace:** Add length to axes independently without kinematic complexity - **Simple programming:** Cartesian coordinates match CAD/CAM models directly - **High precision:** Ball screw drives achieve +/-0.01 mm repeatability over 2-4 m travel

**Limitations:** - **Large footprint:** Gantry frame requires 1.5-2× workspace area for structural supports - **Lower speed:** Linear drives limited to ~10 m/s vs. 15-20 m/s for rotational joints (SCARA) - **Higher cost:** Three independent drive systems vs. shared drive components (Delta)

**Typical Applications:** CNC machine loading (lathes, mills), palletizing, large part handling (automotive castings 20-80 kg), bin picking with extended reach

#### SCARA (Selective Compliance Assembly Robot Arm)

##### Kinematic Structure:

Two rotational joints ( $\theta_1, \theta_2$ ) plus vertical linear axis ( $d_Z$ ) create cylindrical workspace. Forward kinematics:

$$x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \quad (1)$$

$$y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \quad (2)$$

$$z = d_Z \quad (3)$$

where  $L_1, L_2$  are link lengths (typically 300-600 mm each).

##### Inverse Kinematics:

Given target position  $(x, y)$ , solve for joint angles:

$$\theta_2 = \pm \arccos \left( \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2} \right)$$

$$\theta_1 = \arctan \left( \frac{y}{x} \right) - \arctan \left( \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2} \right)$$

Two solutions exist (elbow-up/elbow-down configurations), requiring singularity avoidance algorithms.

#### Performance Characteristics:

- **Workspace:** Annular cylinder ( $R_{\text{outer}} = L_1 + L_2 = 600\text{-}1,200$  mm,  $R_{\text{inner}} = |L_1 - L_2| = 50\text{-}200$  mm)
- **Repeatability:**  $\pm 0.01\text{-}0.02$  mm (better than Cartesian due to rotary encoders on short moment arms)
- **Speed:** 150-300+ picks/minute (rotational acceleration  $5\text{-}10$  rad/s<sup>2</sup> enables fast horizontal motion)
- **Payload:** 1-20 kg typical (limited by cantilever loading at full extension)

**Advantages:** - **Compact footprint:** Overhead mounting frees floor space beneath workspace  
- **High speed:** Rotational dynamics achieve 0.5-1 second cycle times for short moves - **Cost-effective:** Two motors + vertical actuator simpler than three Cartesian axes

**Limitations:** - **Workspace shape:** Annular ring with unusable center region and dead zones at singularities - **Payload variation:** Cantilever moment varies with radius (max torque at full extension = payload  $\times (L_1 + L_2)$ ) - **Complex kinematics:** Inverse solution requires trigonometry, potential singularities at  $\theta_2 = 0^\circ$  or  $\pm 180^\circ$

**Typical Applications:** Electronics assembly (PCB component placement), small parts handling (<5 kg), high-speed sorting, clean room operations (pharmaceutical, semiconductor)

#### Delta (Parallel Linkage) Robots

##### Kinematic Structure:

Three actuated arms connect to end-effector via parallelogram linkages, maintaining constant orientation. Inverse kinematics solved via constraint equations:

$$f_i(\theta_i, x, y, z) = 0, \quad i = 1, 2, 3$$

where each constraint represents circle intersection (arm length constraint).

##### Performance Characteristics:

- **Workspace:** Inverted cone ( $\varnothing 400\text{-}800$  mm  $\times$  200-400 mm height)
- **Repeatability:**  $\pm 0.1\text{-}0.5$  mm (lower than serial robots due to accumulated linkage tolerances)
- **Speed:** 200-300+ picks/minute (parallel structure distributes inertia across three motors)

- **Payload:** 0.5-10 kg (lightweight linkages optimize acceleration over load capacity)

**Advantages:** - **Extreme speed:** Parallel actuation achieves 10-15 g acceleration (vs. 1-3 g for serial robots) - **Low moving mass:** End-effector carries only linkages, motors remain stationary - **Compact footprint:** Overhead mounting with minimal floor space requirement

**Limitations:** - **Limited workspace:** Small volume relative to robot size (workspace  $\approx$  10-15% of installation envelope) - **Orientation constraint:** End-effector remains parallel to base (no wrist articulation in basic designs) - **Complex calibration:** Parallel kinematics sensitive to link length errors ( $\pm 0.1$  mm linkage tolerance  $\Rightarrow$   $\pm 0.5$  mm positioning error)

**Typical Applications:** High-speed packaging (food, pharmaceuticals at 200-300 picks/minute), light parts sorting ( $< 2$  kg), vision-guided pick from conveyor, rapid assembly operations

### 9.1.3 Performance Specifications and Selection Framework

#### Critical Performance Metrics

##### 1. Repeatability vs. Accuracy

- **Repeatability:** Standard deviation of position when returning to same commanded point ( $\sigma = \pm 0.01-0.05$  mm typical)
- **Accuracy:** Deviation from true Cartesian position across workspace (systematic error 0.1-1.0 mm, correctable via calibration)

Most applications specify repeatability (consistent placement matters more than absolute position accuracy, which can be calibrated).

##### 2. Cycle Time Breakdown

Total cycle time comprises:

$$t_{cycle} = t_{approach} + t_{grip/vacuum} + t_{transfer} + t_{release} + t_{return}$$

**Example calculation** (SCARA picking 50 mm  $\times$  50 mm part, 300 mm transfer distance):

- Approach move:  $t = \sqrt{\frac{4d}{a}}$  where  $d = 100$  mm (Z-axis descent),  $a = 5$  m/s<sup>2</sup>  $\Rightarrow t = 0.28$  s
- Vacuum stabilization: 0.15 s (pressure ramp to -60 kPa)
- Transfer move: Trapezoidal profile at  $v_{max} = 3$  m/s,  $a = 8$  m/s<sup>2</sup>  $\Rightarrow t = 0.45$  s
- Release: 0.05 s (valve venting)
- Return: 0.35 s (optimized path)

**Total: 1.28 seconds = 47 picks/minute**

Optimization targets approach and transfer times (60-70% of cycle), requiring path planning and acceleration profiling (Section 9.5).

##### 3. Payload Capacity and Duty Cycle

Rated payload assumes 100% duty cycle (continuous operation). Intermittent operation allows 120-150% overload:



$$\text{Intermittent payload} = \text{Rated payload} \times \sqrt{\frac{100\%}{\text{Duty cycle}\%}}$$

For 10 kg rated robot at 40% duty cycle:

$$\text{Peak payload} = 10 \times \sqrt{\frac{100}{40}} = 15.8 \text{ kg}$$

### Selection Decision Matrix

Factor	Choose Cartesian When:	Choose SCARA When:	Choose Delta When:
<b>Workspace</b>	Large volume (>2 m <sup>3</sup> ), irregular shape	Compact area (<1 m <sup>2</sup> ), overhead clearance	Small area (<0.5 m <sup>3</sup> ), overhead mounting
<b>Payload</b>	>20 kg, moment loading	1-20 kg, point loads	<5 kg, lightweight components
<b>Speed</b>	<30 picks/min acceptable	50-150 picks/min required	>150 picks/min critical
<b>Precision</b>	+/-0.01-0.02 mm repeatability	+/-0.01-0.02 mm repeatability	+/-0.1-0.5 mm acceptable
<b>Footprint</b>	Floor space available (gantry frame)	Minimal floor space (overhead mount)	Minimal floor space (overhead mount)
<b>Cost budget</b>	\$35k-80k entry-level	\$50k-120k mid-range	\$80k-200k high-speed
<b>Programming</b>	Simple Cartesian coordinates	Moderate kinematic complexity	Complex calibration required

### 9.1.4 Integration with CNC Manufacturing Systems

Pick and place robots extend CNC machine utilization from 60-70% (operator-loaded) to 85-95% (automated loading), enabling:

#### 1. Machine Tending

- **Lathe loading:** Robot presents bar stock or castings to chuck, removes finished parts, places in inspection station
- **Mill loading:** Palletized workpieces transferred from fixture queue to machine table, coordinated with tool changes
- **Multi-machine cells:** Single robot services 2-4 CNC machines during cutting operations (overlapped cycle)

#### Typical Sequence:

1. Robot checks machine status (M-code signal: "Ready for load")
2. Open machine door (servo-actuated or pneumatic interlock)
3. Remove finished part (wait for spindle stop + coolant flush)
4. Transfer to unload station (vision inspection optional)

5. Pick new workpiece from fixture queue
6. Place in machine work envelope (precision locating pins  $\pm 0.02$  mm)
7. Close door + send ready signal to CNC controller
8. CNC begins machining cycle while robot services next machine

## 2. Part Inspection and Sorting

Vision-guided robots measure critical dimensions ( $\pm 0.01$  mm optical CMM), sort by tolerance class:

- **Pass:** Route to shipping container
- **Rework:** Stack for secondary operations
- **Scrap:** Reject bin with defect logging

Integration with SPC (Statistical Process Control) provides real-time feedback to CNC controller, adjusting tool offsets to maintain tolerances.

## 3. Assembly Operations

Sequential pick and place builds subassemblies:

- **Fastener insertion:** Screws, rivets placed with 6-axis force control (Section 9.3)
- **Adhesive dispensing:** Robot positions parts while dispenser applies measured bead
- **Press-fit assembly:** Force-limited insertion prevents part damage (monitoring via current feedback)

### 9.1.5 Module Roadmap

This module progresses through comprehensive engineering coverage of pick and place systems:

- **Section 9.2:** Robot architecture (kinematic analysis, workspace optimization, structural design)
- **Section 9.3:** End effector design (vacuum grippers, mechanical grippers, force sensing, quick-change systems)
- **Section 9.4:** Vision and sensor systems (2D/3D vision, pattern recognition, calibration, lighting techniques)
- **Section 9.5:** Motion control (trajectory planning, acceleration profiling, path optimization, vibration suppression)
- **Section 9.6:** CNC integration (communication protocols, M-code handshaking, safety interlocks, coordinated motion)
- **Section 9.7:** Programming methodologies (teach pendant, offline programming, simulation, program structures)
- **Section 9.8:** Safety systems (risk assessment, guarding, light curtains, collaborative operation)
- **Section 9.9:** Performance optimization (cycle time reduction, throughput analysis, multi-robot coordination)
- **Section 9.10:** Maintenance (preventive schedules, consumable replacement, calibration verification)
- **Section 9.11:** Troubleshooting (common faults, diagnostic procedures, systematic problem-solving)
- **Section 9.12:** Conclusion (integration with Modules 1-16, future technologies, career pathways)

Mastery of pick and place robotics—from kinematic fundamentals through vision-guided programming to systematic troubleshooting—positions the engineer to design, integrate, optimize, and maintain automated material handling systems that multiply CNC manufacturing productivity while maintaining the precision and reliability demanded by modern production environments.

---

---

## References

### 1. Industry Standards

- ISO 10218-1:2011 - Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots
- ISO 10218-2:2011 - Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems
- ANSI/RIA R15.06-2012 - Industrial Robots and Robot Systems - Safety Requirements
- ISO 9283:1998 - Manipulating industrial robots - Performance criteria and related test methods

### 2. Technical Books

- Craig, J.J. (2017). *Introduction to Robotics: Mechanics and Control* (4th ed.). Pearson
- Siciliano, B. et al. (2009). *Robotics: Modelling, Planning and Control*. Springer
- Groover, M.P. (2014). *Automation, Production Systems, and Computer-Integrated Manufacturing*. Pearson
- Nof, S.Y. (1999). *Handbook of Industrial Robotics* (2nd ed.). Wiley

### 3. Manufacturer Documentation

- ABB Robotics - IRB Series Technical Reference Manuals
- FANUC Robotics - M-Series Technical Handbooks
- Epson Robots - SCARA and Cartesian Robot User Guides
- Omron Adept - Quattro Delta Robot Technical Specifications

### 4. Technical Papers

- Clavel, R. (1990). "Device for the Movement and Positioning of an Element in Space." US Patent 4,976,582 (original Delta robot patent)
- Miller, K. (2004). "Optimal Kinematic Design of Spatial Parallel Manipulators." *International Journal of Robotics Research*, 23(2), 127-140
- Merlet, J.P. (2006). "Jacobian, Manipulability, Condition Number and Accuracy of Parallel Robots." *ASME Journal of Mechanical Design*, 128(1), 199-206

### 5. Online Resources

- Robotic Industries Association (RIA) - [robotics.org](http://robotics.org)
  - International Federation of Robotics (IFR) - [ifr.org](http://ifr.org)
  - IEEE Robotics and Automation Society - [ieee-ras.org](http://ieee-ras.org)
  - CNCzone Pick and Place Forum - [cnczone.com/forums/automation](http://cnczone.com/forums/automation)
-

## Module 9 - Pick & Place Robot

Regular maintenance ensures reliable operation, prevents unexpected downtime, and extends equipment lifespan.

### Maintenance Schedule

**Daily Checks (Operators)** - Visual inspection of robot and guarding - E-stop function test - Check for unusual noises or vibrations - Verify error-free startup - Clean work area and robot surfaces

**Weekly Maintenance** - Inspect cables and hoses for wear - Check pneumatic connections for leaks - Verify gripper operation - Clean vision system camera and lighting - Check part presence sensors

**Monthly Maintenance** - Lubricate linear guides and ball screws - Inspect timing belts for wear and tension - Check motor mounting bolts (torque) - Test all safety interlocks - Verify positioning accuracy - Clean filters (vacuum, pneumatic)

**Quarterly Maintenance** - Replace vacuum cups (if worn) - Inspect and lubricate bearings - Check encoder alignment and signals - Calibrate vision system - Test emergency stop response time - Update and backup programs

**Annual Maintenance** - Replace timing belts (preventive) - Bearing inspection and replacement if needed - Motor brake inspection (if equipped) - Complete safety system validation - Re-calibration of robot positioning - Cable replacement (if showing wear) - Electrical connection inspection and tightening

### Linear Motion System Maintenance

#### Linear Guides

Lubrication: - Oil or grease depending on type - Follow manufacturer specifications - Typical: Oil every 100km travel or monthly - Wipe excess to prevent contamination

Inspection: - Check for smooth motion (no binding) - Excessive play indicates wear - Listen for grinding noises - Measure backlash

#### Ball Screws

Lubrication: - Oil or grease (NLGI Grade 2) - Lubricate nut and support bearings - Typical: Monthly for continuous operation

Inspection: - Backlash measurement (dial indicator) - Preload adjustment if excessive - Check for corrosion or pitting

#### Timing Belts

Tension Check: - Press belt sideways (should deflect 1-2% of span) - Too loose: positioning errors, skipping - Too tight: bearing wear, belt damage

Wear Inspection: - Cracked or frayed edges - Tooth damage - Glazing or hardening

Replacement: - Every 2000-5000 hours depending on load - Replace both belt and tensioner - Re-tension after break-in period (20 hours)

## **Rotational Joint Maintenance**

### **Bearings**

Inspection: - Rotate by hand (smooth, no roughness) - Check radial and axial play - Listen for grinding or clicking

Lubrication: - Sealed bearings: No maintenance - Open bearings: Grease annually - Use correct grease type (manufacturer spec)

Replacement Indicators: - Excessive noise - Rough rotation - Increased play - Visible corrosion or damage

### **Harmonic Drives / Gear Reducers**

Lubrication: - Follow manufacturer specifications - Typically grease, specific type required - Every 2000-10000 hours

Backlash Check: - Lock output, measure input rotation - Should be <1-3 arcmin - Excessive backlash indicates wear

## **End Effector Maintenance**

### **Vacuum Grippers**

Suction Cups: - Inspect for cracks, tears, hardening - Replace if damaged or reduced performance - Typical lifespan: 6-12 months - Keep spares on hand

Vacuum Generator: - Venturi: Clean filter, check for clogs - Pump: Replace vanes if worn (follow pump spec) - Check vacuum level with gauge

Valves and Tubing: - Inspect for leaks (hissing sound) - Replace cracked or kinked tubing - Clean or replace solenoid valves if sticky

### **Mechanical Grippers**

Jaw Alignment: - Check parallel motion - Adjust guide rods if misaligned

Soft Jaw Pads: - Inspect for wear, tears - Replace when hardened or damaged

Pneumatic Cylinder: - Check for air leaks - Lubricate (inline lubricator or manual) - Inspect rod for scratches or bends

### **Magnetic Grippers**

Electromagnet: - Check coil resistance (should match spec) - Inspect for overheating - Verify switching (energize/de-energize)

Permanent Magnet: - Clean pole face (remove debris) - Check switching mechanism - Verify holding force

## **Vision System Maintenance**

### **Camera**

Cleaning: - Lens: Microfiber cloth, lens cleaner - Sensor: Compressed air (if accessible) - Frequency: Weekly or as needed

Inspection: - Loose mounting bolts - Cable connection security - Image quality check (sharpness, contrast)

### **Lighting**

LED Condition: - Check for failed LEDs - Reduced brightness indicates aging - Typical lifespan: 50,000 hours

Cleaning: - Wipe diffusers and reflectors - Remove dust and debris

**Calibration** - Re-calibrate monthly or after any mechanical changes - Use calibration grid or known reference - Document calibration results

## **Pneumatic System Maintenance**

**Air Compressor** - Follow compressor manufacturer maintenance - Drain condensate daily - Change oil (if oil-lubricated)

### **Air Treatment**

Filter: - Check pressure drop across filter - Replace element when dirty (typically 6-12 months) - Drain condensate from bowl

Regulator: - Verify set pressure - Check for leaks - Inspect gauge accuracy

Lubricator (if used): - Fill oil reservoir - Adjust drip rate (1-2 drops per minute) - Use correct pneumatic oil

**Leak Detection** - Listen for hissing - Apply soap solution to connections - Measure pressure drop over time (system off) - Repair leaks immediately (energy waste)

## **Electrical System Maintenance**

### **Connections**

Inspection: - Tighten terminal screws (torque to spec) - Check for corrosion - Inspect for loose wires - Look for overheating signs (discoloration)

### **Cables**

Inspection: - Flex cables at moving joints - Look for cuts, abrasion, cracks - Check strain relief - Replace if insulation damaged

**Cable Management** - Ensure proper routing (no pinching) - Check cable carriers (drag chains) - Verify bend radius not exceeded

### **Motor and Drive Cooling**

Fans: - Clean fan blades and filters - Verify airflow (hand test) - Replace if noisy or failing

Heat Sinks: - Clean dust from fins - Verify thermal contact - Check for overheating (thermometer or IR camera)

## Controller and Software Maintenance

**Backup Programs** - Weekly backup of robot programs - Store off-system (USB, network) - Version control for program changes - Document changes in logbook

**Firmware Updates** - Check for updates periodically - Read release notes for bug fixes - Test updates on non-production system first - Keep previous version available for rollback

**Battery Backup (if applicable)** - Check battery voltage - Replace every 1-3 years - Prevents program loss on power failure

**Log Files** - Review error logs for trends - Identify recurring faults - Clear old logs to free space

## Calibration and Accuracy Verification

### Positioning Accuracy

Test Procedure: 1. Jog robot to test position 2. Measure actual position (CMM, indicator) 3. Compare to commanded position 4. Repeat 10-20 times 5. Calculate repeatability (standard deviation)

Acceptance Criteria: - Repeatability:  $\pm 0.05\text{mm}$  (Cartesian),  $\pm 0.02\text{mm}$  (SCARA) - Accuracy:  $\pm 0.1\text{mm}$  after calibration

Re-calibration: - Annually or after accuracy degrades - After mechanical repairs or adjustments

**Vision System Calibration** - Test with known calibration grid - Measure part positions and compare - Re-calibrate if error  $> 0.1\text{mm}$

**Force/Torque Sensor Calibration** - Use known weights - Zero offset with no load - Verify reading accuracy - Re-calibrate annually

## Documentation

**Maintenance Log** - Date and time - Maintenance performed - Parts replaced - Operator/technician name - Observations (wear, issues)

**Parts Inventory** - Common wear items (belts, cups, filters) - Lead time for special parts - Minimum stock levels - Supplier contact info

**Manuals and Drawings** - Mechanical drawings - Electrical schematics - Component datasheets - Manufacturer manuals

## Predictive Maintenance

### Condition Monitoring

Vibration Analysis: - Baseline measurement when new - Periodic re-measurement - Increased vibration indicates bearing wear

Current Monitoring: - Servo drive current logs - Increased current indicates friction or binding - Detect degrading performance before failure

Position Error: - Track following error over time - Increasing error indicates mechanical wear

**Data-Driven Maintenance** - Log cycle counts (picks performed) - Schedule maintenance by usage, not just time - Example: Replace belt every 1 million cycles

**Failure Mode Analysis** - Track failure types and causes - Prioritize maintenance on weak points  
- Stock critical spare parts

## Spare Parts Recommendations

**Critical Spares (Keep on Hand)** - Vacuum cups (2-4 sets) - Timing belts (1 set) - Fuses and circuit breakers - Sensor replacements (proximity, photoelectric) - Pneumatic valves and fittings - Cable set (motor, encoder)

**Secondary Spares (Longer Lead Time OK)** - Bearings (linear, rotational) - Motors and drives - Vision camera and lens - Gripper assemblies

---

**Next:** 9.11 Troubleshooting

---

## References

### 1. Maintenance Standards

- ISO 13849-1:2015 - Safety-related parts of control systems
- ISO 9283:1998 - Manipulating industrial robots - Performance criteria
- Preventive maintenance best practices - Industry guidelines

### 2. Lubrication and Bearings

- THK Linear Guide Maintenance Manual
- HIWIN Linear Guideway Lubrication Guide
- SKF Bearing Maintenance Handbook

### 3. Calibration Procedures

- Robot manufacturer calibration documentation
- Laser tracker calibration systems - API, Faro, Leica
- Vision system calibration techniques

### 4. Reliability Engineering

- Moubray, J. (1997). *Reliability-Centered Maintenance*. Industrial Press
  - MTBF and MTTR analysis techniques
  - Predictive maintenance and condition monitoring
- 

## Module 9 - Pick & Place Robot

Common issues with pick and place systems and systematic approaches to diagnosis and repair.

### Systematic Troubleshooting Approach

**Step 1: Observe and Document** - What is the symptom? (missed picks, positioning error, fault code) - When did it start? (after what event or change) - How often? (intermittent, constant,



increasing frequency) - Under what conditions? (specific part, position, speed)

**Step 2: Isolate the Problem** - Which subsystem? (mechanical, electrical, vision, control) - Single component or system interaction? - Test subsystems independently

**Step 3: Form Hypothesis** - Based on symptoms, what could cause this? - List possible causes (most to least likely)

**Step 4: Test Hypothesis** - Check each possibility systematically - Use measurements, not just guesses

**Step 5: Implement Solution** - Make one change at a time - Verify fix resolves issue - Document solution

## **Mechanical Issues**

### **Positioning Errors**

**Symptom:** Robot does not reach commanded position accurately

#### **Possible Causes:**

1. Belt Tension
  - Check: Press belt, measure deflection
  - Fix: Adjust tensioner to spec
2. Backlash in Drive System
  - Check: Measure position reversal error
  - Fix: Adjust ball screw preload, tighten couplings
3. Encoder Issues
  - Check: Verify encoder counts match motion
  - Fix: Re-align encoder, replace if damaged
4. Loose Mounting Bolts
  - Check: Torque all structural bolts
  - Fix: Tighten to specification, use thread-locker
5. Worn Linear Guides
  - Check: Excessive play, rough motion
  - Fix: Replace worn guide blocks
6. Thermal Expansion
  - Check: Positioning worse after warmup?
  - Fix: Allow warmup period, temperature compensation

**Diagnostic Procedure:** 1. Move to position multiple times, measure repeatability 2. If repeatable but inaccurate: Calibration issue 3. If non-repeatable: Mechanical looseness or wear

### **Excessive Vibration**

**Symptom:** Robot shakes during motion, poor surface finish, noise

#### **Possible Causes:**

1. Resonance
  - Check: Vibration at specific frequency?

- Fix: Adjust acceleration, add damping, change speed
- 2. Loose Components
  - Check: Inspect all bolts and connections
  - Fix: Tighten securely
- 3. Worn Bearings
  - Check: Rough rotation, play in shaft
  - Fix: Replace bearings
- 4. Imbalanced Moving Mass
  - Check: Vibration worse at high speed?
  - Fix: Add counterweight, redistribute mass
- 5. Belt Resonance
  - Check: Belt flutter during motion
  - Fix: Increase tension, shorter belt span
- 6. Control Tuning
  - Check: PID gains too high (oscillation)
  - Fix: Reduce gains, optimize tuning

## **Binding or Sticking**

**Symptom:** Axis moves rough or stops, motor stalls

### **Possible Causes:**

1. Misalignment
  - Check: Linear guides parallel, ball screw aligned
  - Fix: Re-align components
2. Over-Tightening
  - Check: Excessive preload on bearings or guides
  - Fix: Adjust to proper preload
3. Debris
  - Check: Chips or contamination in guides
  - Fix: Clean thoroughly, add covers
4. Insufficient Lubrication
  - Check: Dry appearance, squeaking
  - Fix: Lubricate per schedule
5. Damaged Components
  - Check: Bent shaft, damaged ball screw
  - Fix: Replace damaged parts
6. Collision Damage
  - Check: Recent crash or impact?
  - Fix: Inspect and replace bent or broken parts

## **Gripper Issues**

### **Failed Picks (Vacuum)**

**Symptom:** Part not picked up, vacuum switch not triggered

### **Possible Causes:**

1. Insufficient Vacuum
  - Check: Vacuum gauge reading
  - Fix: Increase air pressure (venturi), check pump
2. Leaks
  - Check: Hissing sound, soap solution test
  - Fix: Tighten fittings, replace damaged tubing
3. Worn Suction Cups
  - Check: Cracks, hardening, tears
  - Fix: Replace cups
4. Valve Malfunction
  - Check: Manual actuation, electrical signal
  - Fix: Clean or replace valve
5. Part Surface Issues
  - Check: Part too porous, textured, or curved
  - Fix: Use foam cups, multiple cups, or different gripper type
6. Timing
  - Check: Vacuum applied before contact?
  - Fix: Adjust program sequence

#### **Diagnostic Procedure:**

1. Manually actuate vacuum (bypass controller)
  - Works: Electrical/control issue
  - Fails: Pneumatic/mechanical issue
2. Check vacuum level at generator
  - Low: Compressor pressure, venturi issue
  - OK: Leak downstream
3. Plug gripper tube, check vacuum
  - Low: Valve or generator problem
  - OK: Cup or part interface issue

#### **Failed Picks (Mechanical Gripper)**

**Symptom:** Jaws do not close on part, part slips

#### **Possible Causes:**

1. Part Misalignment
  - Check: Part position vs. jaw position
  - Fix: Adjust pick position, improve vision
2. Insufficient Grip Force
  - Check: Air pressure, cylinder size
  - Fix: Increase pressure, larger cylinder
3. Jaw Worn or Damaged
  - Check: Jaw pad condition
  - Fix: Replace pads
4. Cylinder Sticking
  - Check: Smooth cylinder stroke

- Fix: Lubricate, replace seals
5. Proximity Sensor False Trigger
    - Check: Sensor output vs. actual position
    - Fix: Adjust sensor position, replace if failed

## **Dropped Parts**

**Symptom:** Part falls during transport

### **Possible Causes:**

1. Vacuum Loss
  - Check: Vacuum level during motion
  - Fix: Check for leaks, increase vacuum reserve
2. Excessive Acceleration
  - Check: Dropped during rapid motion?
  - Fix: Reduce acceleration, increase grip force
3. Part Weight Exceeds Capacity
  - Check: Gripper specification vs. actual part weight
  - Fix: Use larger gripper, reduce speed
4. Vibration
  - Check: Parts drop during rough motion
  - Fix: Reduce vibration, improve grip

## **Vision System Issues**

### **Part Not Found**

**Symptom:** Vision system fails to locate part

### **Possible Causes:**

1. Lighting
  - Check: Image too dark, too bright, glare
  - Fix: Adjust lighting intensity, angle, or use polarization
2. Focus
  - Check: Image blurry
  - Fix: Adjust lens focus, aperture
3. Part Outside Field of View
  - Check: Part position vs. camera view
  - Fix: Adjust camera position, widen FOV
4. Occlusion
  - Check: Part obscured by other objects
  - Fix: Improve part presentation, staging
5. Algorithm Threshold
  - Check: Threshold settings too tight
  - Fix: Adjust vision parameters, retrain
6. Camera Malfunction
  - Check: No image or corrupted image
  - Fix: Check cable, power, replace camera

**Diagnostic Procedure:** 1. Capture and save image 2. Review image quality (lighting, focus, framing) 3. Test vision algorithm offline with saved image 4. Adjust parameters or improve image quality

### **Incorrect Position Reported**

**Symptom:** Vision reports wrong position, robot picks wrong location

#### **Possible Causes:**

1. Calibration Error
  - Check: Known position test piece
  - Fix: Re-calibrate camera-to-robot transform
2. Lens Distortion
  - Check: Straight lines appear curved
  - Fix: Calibrate intrinsic parameters, use undistortion
3. Camera Movement
  - Check: Camera mounting secure
  - Fix: Tighten camera mount
4. Wrong Algorithm
  - Check: Detects wrong feature
  - Fix: Adjust parameters, use different algorithm
5. Part Variant
  - Check: Part type different than expected
  - Fix: Train for multiple variants

### **Slow Vision Processing**

**Symptom:** Vision cycle time too long

#### **Possible Causes:**

1. High Resolution Image
  - Check: Image size
  - Fix: Reduce resolution, use ROI (region of interest)
2. Complex Algorithm
  - Check: Processing time per step
  - Fix: Simplify, use faster algorithm
3. CPU Overload
  - Check: CPU usage during processing
  - Fix: Faster computer, optimize code, GPU acceleration
4. Network Latency
  - Check: Networked camera or PC
  - Fix: Direct connection, faster network

### **Electrical and Control Issues**

#### **Motor Does Not Move**

**Symptom:** No motion when commanded

**Possible Causes:**

1. E-Stop Activated
  - Check: E-stop button status
  - Fix: Reset e-stop, investigate why triggered
2. Drive Fault
  - Check: Drive display, fault code
  - Fix: Clear fault, address root cause
3. Enable Signal Missing
  - Check: Drive enable input
  - Fix: Check wiring, controller output
4. No Command Signal
  - Check: Step/direction or analog command at drive
  - Fix: Controller output, cable connection
5. Motor Cable Disconnected
  - Check: Cable connections at motor and drive
  - Fix: Reconnect, check for damage
6. Software Limit Exceeded
  - Check: Controller limit status
  - Fix: Jog opposite direction, adjust limits

**Diagnostic Procedure:**

1. Check drive power LED
  - Off: Power supply issue
  - On: Proceed to step 2
2. Check drive enable signal
  - Not present: Controller or wiring issue
  - Present: Proceed to step 3
3. Manually jog from controller
  - Works: Program issue
  - Fails: Drive, motor, or feedback issue
4. Check fault codes on drive
  - Displays fault: Address specific fault
  - No fault: Motor or cable issue

**Encoder Errors**

**Symptom:** Encoder fault, following error, motor runaway

**Possible Causes:**

1. Encoder Cable Disconnected
  - Check: Cable connections
  - Fix: Reconnect, secure strain relief
2. Encoder Power Missing
  - Check: +5V at encoder

- Fix: Check power supply, wiring
- 3. Noise Interference
  - Check: Encoder signal quality (oscilloscope)
  - Fix: Shielded cable, proper grounding, route away from power cables
- 4. Encoder Misalignment
  - Check: Encoder mounting
  - Fix: Re-align encoder to motor shaft
- 5. Encoder Damaged
  - Check: No signal or erratic counts
  - Fix: Replace encoder

## **Communication Faults**

**Symptom:** Controller cannot communicate with drives, I/O, or vision system

### **Possible Causes:**

1. Cable Disconnected
  - Check: Physical connections
  - Fix: Reconnect, secure
2. Wrong Baud Rate / Settings
  - Check: Communication parameters match
  - Fix: Configure both devices identically
3. Termination Missing
  - Check: Network termination resistors (RS-485, CAN)
  - Fix: Add terminators at both ends
4. Network Conflict
  - Check: Duplicate addresses or IP conflicts
  - Fix: Assign unique addresses
5. Electrical Noise
  - Check: Intermittent errors during motor motion
  - Fix: Shielded cables, proper grounding
6. Failed Component
  - Check: Substitute known-good device
  - Fix: Replace failed device

## **Software and Programming Issues**

### **Program Faults or Crashes**

**Symptom:** Program stops with error, unexpected behavior

### **Possible Causes:**

1. Syntax Error
  - Check: Error message, line number
  - Fix: Correct syntax
2. Unreachable Position
  - Check: Position beyond limits or singularity
  - Fix: Adjust position, check kinematics

3. Missing Subroutine
  - Check: Called subroutine exists
  - Fix: Add missing routine or fix call
4. Variable Out of Range
  - Check: Array index, division by zero
  - Fix: Validate inputs, add error checking
5. Timeout
  - Check: Waiting for signal that never comes
  - Fix: Check I/O, add timeout handling

## **Inconsistent Behavior**

**Symptom:** Robot behaves differently between runs

### **Possible Causes:**

1. Uninitialized Variables
  - Check: Variables set before use
  - Fix: Initialize at program start
2. Race Conditions
  - Check: Timing-dependent behavior
  - Fix: Add synchronization, dwells
3. External Interference
  - Check: Operator interaction, other equipment
  - Fix: Isolate robot, add interlocks
4. Hardware Intermittent Fault
  - Check: Sensors, connections
  - Fix: Replace suspect components

## **Safety System Issues**

### **Unexpected Stops**

**Symptom:** Robot stops during operation, no operator action

### **Possible Causes:**

1. Light Curtain Triggered
  - Check: Beam broken, alignment
  - Fix: Clear obstruction, realign
2. Safety Mat Activation
  - Check: Mat status
  - Fix: Clear mat, check for debris
3. Guard Interlock Open
  - Check: Guard switch status
  - Fix: Close guard, check switch alignment
4. Vibration False Trigger
  - Check: Timing of stops vs. motion
  - Fix: Secure sensors, adjust sensitivity



## Safety System Bypass Attempt

**Symptom:** Operator wants to bypass safety devices

**Root Causes:** - Nuisance trips (false alarms) - Perceived production pressure - Lack of understanding

**Solutions:** - Address root cause of nuisance trips - Reinforce training on safety importance - Management support for safety priority - Never allow bypassing safety devices

## Emergency Procedures

**Robot Collision** 1. Press e-stop immediately 2. Lockout/tagout power 3. Inspect for damage (bent parts, cracked welds) 4. Check alignment and function before restart 5. Investigate root cause

**Fire or Smoke** 1. Press e-stop 2. Evacuate personnel 3. Call emergency services 4. Use appropriate fire extinguisher (electrical Class C) 5. Do not restart until inspected

**Trapped Personnel** 1. Press e-stop 2. Lockout/tagout 3. Manually release gripper or move axes 4. Provide first aid 5. Investigate and prevent recurrence

## Troubleshooting Tools

**Diagnostic Equipment** - Multimeter (voltage, continuity) - Oscilloscope (signal quality, timing) - Dial indicator (position accuracy) - Vacuum gauge - Infrared thermometer - Vibration analyzer

**Software Tools** - Controller diagnostics and logs - Vision system test mode - Drive configuration software - Network analyzer (Wireshark for Ethernet)

**Documentation** - Electrical schematics - Mechanical drawings - Software backups and version history - Maintenance logs

---

**Next:** 9.12 Conclusion

---

## References

1. **Troubleshooting Methodology**
  - Systematic fault diagnosis techniques
  - Root cause analysis (RCA) - 5 Whys, Fishbone diagrams
  - Failure Mode and Effects Analysis (FMEA)
2. **Diagnostic Tools**
  - Oscilloscope usage - Tektronix, Keysight application notes
  - Network protocol analyzers - Wireshark documentation
  - Motor drive diagnostic software - Manufacturer manuals
3. **Motion System Troubleshooting**
  - Servo system tuning and diagnostics
  - Encoder troubleshooting guides

- Mechanical alignment and backlash measurement
4. **Vision System Diagnostics**
- Camera and lighting troubleshooting
  - Image quality assessment techniques
  - Calibration verification procedures
- 

## Module 9 - Pick & Place Robot

### Summary

Pick and place robots transform manufacturing operations by automating repetitive material handling tasks with precision, speed, and reliability. This module covered the complete spectrum of pick and place system design, integration, and operation.

### Key Topics Covered:

**Robot Architectures** - Cartesian, SCARA, and Delta configurations each offer distinct advantages. Cartesian robots provide simplicity and rigidity for heavy loads. SCARA systems deliver speed in compact footprints. Delta robots achieve the highest speeds for light parts. Architecture selection depends on application requirements, workspace constraints, and performance targets.

**Kinematics and Motion Control** - Understanding forward and inverse kinematics enables accurate positioning. Trajectory planning optimizes cycle time while respecting mechanical limits. Modern motion controllers coordinate multiple axes, blend motion paths, and provide feedback control for precise, repeatable operation.

**End Effectors** - Vacuum grippers suit smooth, flat surfaces. Mechanical grippers handle irregular shapes with positive force. Magnetic grippers excel with ferrous materials. Selection requires analysis of part characteristics, grip force requirements, and cycle time constraints.

**Vision and Sensors** - Machine vision locates parts, adapts to position variations, and performs quality inspection. Cameras, lighting, and image processing algorithms work together to guide robot motion. Proximity sensors, force sensors, and encoders provide feedback for reliable operation.

**CNC Integration** - Pick and place robots integrated with CNC machines create automated manufacturing cells. Communication via digital I/O, serial protocols, or industrial Ethernet enables coordination between systems. Multi-machine cells maximize equipment utilization and enable lights-out manufacturing.

**Programming** - Robot programs define motion sequences, gripper actions, and decision logic. Methods range from teach pendants to text-based coding to graphical interfaces. Vision-guided programming and error handling create robust, adaptive systems.

**Safety** - Pick and place systems present serious hazards requiring proper safeguarding. Physical guards, interlocked gates, presence-sensing devices, and emergency stops protect personnel. Compliance with ISO 12100, ISO 10218, and ANSI/RIA R15.06 ensures legal and ethical operation.

**Performance Optimization** - Cycle time reduction through motion optimization, gripper improvements, and vision processing acceleration directly impacts throughput and ROI. Systematic anal-

ysis identifies bottlenecks and guides optimization efforts.

**Maintenance and Troubleshooting** - Regular maintenance prevents failures and extends equipment life. Systematic troubleshooting approaches isolate problems efficiently. Documentation and spare parts support rapid recovery from faults.

## Practical Implementation

Building a pick and place system requires integration of mechanical, electrical, and software disciplines:

**Mechanical Design** - Select architecture appropriate for application - Design rigid structures to minimize deflection - Choose quality linear motion components - Implement proper cable management - Ensure adequate workspace clearances

**Electrical System** - Size motors and drives for required performance - Implement safety-rated circuits - Use industrial-grade sensors and I/O - Provide proper grounding and shielding - Plan for expansion and modification

**Control System** - Choose controller supporting required kinematics - Implement motion planning and coordination - Integrate vision and sensor feedback - Develop robust error handling - Provide operator interface

**Safety and Compliance** - Perform risk assessment - Implement appropriate safeguarding - Install emergency stop circuits - Validate safety functions - Document thoroughly

## Economic Considerations

Pick and place automation delivers rapid ROI when properly applied:

**Cost Components** - Mechanical: \$2,000-\$20,000 (DIY to industrial) - Motors and drives: \$1,000-\$10,000 - Controller: \$500-\$5,000 - Vision system: \$500-\$5,000 - End effectors: \$200-\$2,000 - Safety systems: \$1,000-\$5,000 - Integration labor: Variable

**Return on Investment** - Labor savings: Eliminating 1-2 operators - Increased throughput: 2-10x vs. manual - Improved quality: Consistent placement, reduced errors - Flexibility: Quick changeover between part types - Typical payback: 6 months to 2 years

**Build vs. Buy** - DIY build: 30-50% of commercial cost, higher integration effort - Commercial system: Higher initial cost, warranty and support - Hybrid: Commercial robot with custom tooling and integration

## Future Trends

**Advanced Vision** - 3D vision for bin picking - AI-based object recognition - Adaptive learning algorithms

**Collaborative Operation** - Safer human-robot interaction - Reduced guarding requirements - Flexible workspace sharing

**Connectivity** - IoT integration (MQTT, OPC UA) - Cloud-based monitoring and analytics - Predictive maintenance - Digital twin simulation

**Miniaturization** - Desktop pick and place under \$5,000 - Accessible to small shops and makers  
- Educational applications

**Modular Systems** - Standardized interfaces - Quick reconfiguration - Reusable components across applications

## Learning Pathways

**Beginner Projects** 1. Simple XY Cartesian pick and place (no vision) 2. Add vacuum gripper and part presence sensor 3. Integrate basic vision for part location 4. Implement safety interlocks and e-stop

**Intermediate Projects** 1. SCARA robot with teach pendant 2. CNC machine loading/unloading cell 3. Vision-guided sorting system 4. Multi-machine cell coordination

**Advanced Projects** 1. Delta robot for high-speed packaging 2. 3D vision bin picking 3. Collaborative robot with force sensing 4. Integrated manufacturing cell with MES communication

## Resources for Further Learning

**Standards and Regulations** - ISO 12100: Machinery safety - ISO 10218: Robot safety - ANSI/RIA R15.06: Industrial robot safety - ISO 13849: Safety control systems

**Technical References** - Robotics textbooks (Craig, Spong, Siciliano) - Vision system documentation (OpenCV, Halcon) - Controller manuals (LinuxCNC, Mach3) - Component datasheets (motors, drives, sensors)

**Online Communities** - CNCzone forums - LinuxCNC community - Robotics Stack Exchange - DIY pick and place groups

**Commercial Training** - Robot manufacturer training courses - Vision system workshops - Safety certification programs

## Closing Thoughts

Pick and place robots represent a highly practical application of automation technology. They combine mechanical design, motion control, sensors, and programming in systems that deliver measurable value. Success requires understanding the fundamentals, careful planning, quality execution, and ongoing optimization.

Whether building a desktop system for electronics assembly or a large Cartesian robot for CNC machine tending, the principles remain consistent: design for the application requirements, integrate subsystems thoughtfully, implement proper safety measures, and continuously improve performance.

The skills developed in this module apply broadly to automation, robotics, and manufacturing engineering. Pick and place systems serve as an excellent platform for learning integration, troubleshooting, and optimization—capabilities that transfer to more complex automation challenges.

As manufacturing continues to evolve toward greater automation and flexibility, pick and place robots will play an increasingly central role. Understanding their design, implementation, and operation positions you to contribute meaningfully to modern manufacturing technology.

## Next Steps

After completing this module, consider:

1. **Design a System** - Specify a pick and place robot for a real application in your shop or workplace
2. **Build a Prototype** - Construct a simple Cartesian or SCARA system
3. **Integrate Vision** - Add camera-based part localization
4. **Connect to CNC** - Automate loading/unloading of an existing machine
5. **Optimize Performance** - Measure and reduce cycle time
6. **Share Knowledge** - Document your system and help others learn

The practical experience gained from hands-on implementation cements theoretical knowledge and reveals nuances not apparent in textbooks. Start with a simple project, achieve success, and progressively tackle more complex challenges.

---

## Module 9 Complete

**Previous Module:** Module 8 - Waterjet Systems

**Next Module:** Module 10 - Robotic Arm Systems

**Return to:** Course Overview

---

## References

1. **Comprehensive Robotics Texts**
    - Craig, J.J. (2017). *Introduction to Robotics: Mechanics and Control* (4th ed.). Pearson
    - Siciliano, B. et al. (2009). *Robotics: Modelling, Planning and Control*. Springer
    - Lynch, K.M. & Park, F.C. (2017). *Modern Robotics*. Cambridge University Press
  2. **Industrial Automation**
    - Groover, M.P. (2014). *Automation, Production Systems, and Computer-Integrated Manufacturing*. Pearson
    - Nof, S.Y. (2009). *Springer Handbook of Automation*. Springer
  3. **Standards and Safety**
    - ISO 10218-1/2:2011 - Robot safety standards
    - ANSI/RIA R15.06-2012 - Industrial robot safety requirements
    - ISO 12100:2010 - Machinery safety principles
  4. **Online Communities and Resources**
    - Robotic Industries Association (RIA) - [robotics.org](http://robotics.org)
    - IEEE Robotics and Automation Society - [ieee-ras.org](http://ieee-ras.org)
    - CNCzone Forums - [cnczone.com](http://cnczone.com)
    - ROS Community - [ros.org](http://ros.org)
-

## Module 9 - Pick & Place Robot

### 9.2.1 Cartesian (Gantry) Robot Architecture

#### Structural Configuration

Cartesian robots employ three orthogonal linear axes (X, Y, Z) arranged in portal/gantry configuration—horizontal X-Y platform spans workspace, vertical Z-axis carries end-effector. This architecture provides rectangular workspace with 1:1 correspondence between commanded coordinates and physical motion (no kinematic transformation).

#### Common Configurations:

1. **Portal/Gantry:** X-axis beam spans Y-axis rails elevated above work surface (clearance for parts/fixtures)
2. **Cantilever:** X-axis extends from one Y-axis rail (asymmetric loading, lower cost)
3. **Ceiling-mounted:** Inverted gantry suspends from overhead structure (maximizes floor space)

#### Forward Kinematics

Direct mapping from joint space to Cartesian space:

$$\mathbf{P}_{tool} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d_X \\ d_Y \\ d_Z \end{bmatrix} + \mathbf{P}_{tool\_offset}$$

where  $d_X, d_Y, d_Z$  are motor encoder positions (converted to linear displacement via pitch:  $d = N_{counts} \times \frac{pitch}{encoder\_resolution}$ ).

**No inverse kinematics required**—target Cartesian position directly commands motor positions.

#### Structural Analysis: Beam Deflection

X-axis beam deflection limits payload and affects positioning accuracy. For simply-supported beam with center load:

$$\delta_{max} = \frac{WL^3}{48EI}$$

where: -  $W$  = total load (payload + Z-axis + end-effector), N -  $L$  = beam span (X-axis travel + 2 × overhang), m -  $E$  = Young's modulus (aluminum 6061-T6: 69 GPa, steel: 200 GPa) -  $I$  = second moment of area (beam cross-section), m<sup>4</sup>

#### Example Calculation:

For 1,200 mm span aluminum extrusion beam (80×80 mm,  $I = 2.1 \times 10^{-6}$  m<sup>4</sup>) carrying 15 kg load:

$$\delta = \frac{15 \times 9.81 \times 1.2^3}{48 \times 69 \times 10^9 \times 2.1 \times 10^{-6}} = \frac{2,545}{6.94 \times 10^6} = 0.37 \text{ mm}$$

**Design guideline:** Deflection < 0.5 mm for +/-0.05 mm repeatability (deflection contributes ~50% of positioning error budget).

**Mitigation strategies:** - Increase beam section modulus (larger extrusion: 100×100 mm □  $\delta$  reduces 60%) - Add intermediate support at mid-span (reduces effective  $L$  by 2×, deflection by 8×) - Use steel or carbon fiber beam (higher  $E$ )

### Natural Frequency Analysis

Gantry structure must avoid resonance with motor excitation frequencies (stepper: 50-500 Hz, servo: 100-2,000 Hz). First bending mode natural frequency:

$$f_n = \frac{\lambda_n^2}{2\pi L^2} \sqrt{\frac{EI}{m}}$$

where  $\lambda_1 = 3.516$  for simply-supported beam,  $m$  = mass per unit length (kg/m).

**Target:**  $f_n > 50$  Hz (ensures 10:1 margin below typical stepper resonance range).

For aluminum beam ( $m = 2.7$  kg/m, parameters from deflection example):

$$f_n = \frac{3.516^2}{2\pi \times 1.2^2} \sqrt{\frac{69 \times 10^9 \times 2.1 \times 10^{-6}}{2.7}} = 64 \text{ Hz}$$

Acceptable for stepper-driven system; servo-driven requires  $f_n > 100$  Hz (requires stiffer/lighter beam).

### Linear Motion System Selection

Component	Type	Typical Specification	Application
<b>X-axis (long span)</b>	Belt drive (GT3)	Span 1-6 m, speed 5-10 m/s	Fast travel, moderate precision
<b>Y-axis (medium span)</b>	Ball screw (C5)	Span 0.5-2 m, pitch 10-20 mm	High precision, backlash <0.01 mm
<b>Z-axis (vertical)</b>	Ball screw + counter-balance	Stroke 200-500 mm, pitch 5-10 mm	Precision positioning against gravity
<b>Linear guides</b>	HGR15-25	Preload: medium, clearance <5 μm	X-Y-Z axes (rail size scales with span)

### Belt drive sizing example (X-axis):

For 3 m/s maximum speed, 5 m/s<sup>2</sup> acceleration, 20 kg moving mass:

- Required motor torque:  $T = \frac{F \times r}{i}$  where  $F = ma = 20 \times 5 = 100$  N,  $r = \frac{\text{pitch}}{2\pi}$  (pulley radius),  $i$  = gear ratio
- Using 20-tooth GT3 pulley (pitch 3 mm):  $r = \frac{20 \times 3}{2\pi} = 9.55$  mm = 0.00955 m
- Motor torque (direct drive,  $i = 1$ ):  $T = 100 \times 0.00955 = 0.955$  N·m
- Select NEMA 34 stepper (3 N·m holding torque) □ 3:1 safety factor

## 9.2.2 SCARA Robot Architecture

### Kinematic Configuration

SCARA (Selective Compliance Assembly Robot Arm) employs two revolute joints (shoulder  $\theta_1$ , elbow  $\theta_2$ ) in horizontal plane plus prismatic Z-axis. “Selective compliance” refers to rigid vertical axis (high stiffness for insertion tasks) with compliant horizontal plane (absorbs lateral misalignment).

### Forward Kinematics

End-effector position from joint angles:

$$x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \quad (4)$$

$$y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \quad (5)$$

$$z = d_Z \quad (6)$$

where  $L_1$  = proximal link length (shoulder to elbow),  $L_2$  = distal link length (elbow to wrist).

### Inverse Kinematics

Given target  $(x, y)$ , solve for joint angles using law of cosines:

$$\cos \theta_2 = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2}$$

$$\theta_2 = \pm \arccos \left( \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2} \right)$$

Two solutions exist (“elbow-up” and “elbow-down”), typically select  $\theta_2 > 0$  (elbow-up) to avoid workspace obstacles.

$$\theta_1 = \arctan \left( \frac{y}{x} \right) - \arctan \left( \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2} \right)$$

### Singularity conditions:

1. **Full extension:**  $\theta_2 = 0^\circ$  □ elbow locked, loss of one DOF (cannot move perpendicular to arm)



2. **Full retraction:**  $\theta_2 = \pm 180^\circ$  □ arms overlap, kinematic indeterminacy

**Workspace:** Annulus with outer radius  $R_{max} = L_1 + L_2$  and inner radius  $R_{min} = |L_1 - L_2|$ .

### Joint Torque Analysis

Shoulder joint supports elbow link, payload, and end-effector; maximum torque at full extension:

$$\tau_{shoulder} = (m_{link2} \times L_1 + m_{payload} \times (L_1 + L_2)) \times g \times \cos \theta_1$$

**Example:**  $L_1 = 400$  mm,  $L_2 = 300$  mm, link mass 2 kg, payload 5 kg, horizontal configuration ( $\theta_1 = 0^\circ$ ):

$$\tau = (2 \times 0.4 + 5 \times 0.7) \times 9.81 \times 1 = (0.8 + 3.5) \times 9.81 = 42.2 \text{ N}\cdot\text{m}$$

Select servo motor with continuous torque  $> 50 \text{ N}\cdot\text{m}$  (includes 20% safety margin + dynamic acceleration torque).

### Harmonic Drive Gearboxes

SCARA joints commonly use harmonic (strain wave) gear reducers:

- **Ratio:** 50:1 to 100:1 typical
- **Backlash:**  $< 1$  arcmin ( $0.017^\circ$ ) enables  $\pm 0.01$  mm repeatability at 500 mm radius
- **Efficiency:** 70-85% (vs. 90-95% planetary gears, but zero-backlash critical for precision)

### Torque calculation with gearbox:

Required motor torque =  $\frac{\tau_{load}}{i \times \eta}$  where  $i$  = gear ratio,  $\eta$  = efficiency

For  $42.2 \text{ N}\cdot\text{m}$  load, 80:1 ratio, 75% efficiency:

$$\tau_{motor} = \frac{42.2}{80 \times 0.75} = 0.70 \text{ N}\cdot\text{m}$$

Select 1 kW servo motor (rated  $3.2 \text{ N}\cdot\text{m}$  @ 3,000 RPM) □ adequate torque and speed.

### Z-Axis Counterbalance

Vertical axis lifts payload against gravity; counterbalance spring reduces motor torque:

$$F_{motor} = F_{payload} - F_{spring} = (m_{payload} + m_{end-effector}) \times g - k \times \Delta z$$

where  $k$  = spring constant (N/mm),  $\Delta z$  = compression from neutral position.

**Design goal:** Balance at mid-stroke □ motor provides only acceleration forces, not gravity compensation.

For 5 kg payload, 200 mm stroke:

$$k = \frac{mg}{z_{stroke}/2} = \frac{5 \times 9.81}{0.1} = 490 \text{ N/m} = 0.49 \text{ N/mm}$$

Use two 0.25 N/mm springs in parallel (each 250 mm free length, 30 mm wire diameter).

### 9.2.3 Delta Robot Parallel Kinematic Architecture

#### Structural Topology

Delta robot consists of three kinematic chains (120° radial symmetry) connecting fixed base to moving platform via parallelogram linkages. Each chain has one actuated revolute joint at base; parallelogram constrains platform orientation parallel to base (3-DOF: X, Y, Z translation).

#### Link Nomenclature:

- **Proximal arm:** Base motor shaft to upper parallelogram joint (200-400 mm carbon fiber tube)
- **Distal linkage:** Parallelogram structure maintaining parallel orientation (400-800 mm length)
- **Platform:** Triangular end-effector mount (50-150 mm triangle)

#### Inverse Kinematics (Analytical Solution)

Given platform position  $(x, y, z)$ , compute motor angles  $\theta_1, \theta_2, \theta_3$ :

For each chain  $i$ , platform joint position:

$$\mathbf{P}_i = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + R_i \begin{bmatrix} \cos \alpha_i \\ \sin \alpha_i \\ 0 \end{bmatrix}$$

where  $R_i$  = platform radius,  $\alpha_i = (i - 1) \times 120^\circ$  (radial position).

Proximal arm endpoint (motor shaft):

$$\mathbf{B}_i = \begin{bmatrix} R_b \cos \alpha_i \\ R_b \sin \alpha_i \\ 0 \end{bmatrix}$$

where  $R_b$  = base radius.

Constraint:  $|\mathbf{P}_i - \mathbf{B}_i - L_p \mathbf{u}_i| = L_d$  (distal link length), where  $\mathbf{u}_i$  = proximal arm unit vector.

Solving yields:

$$\theta_i = 2 \arctan \left( \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \right)$$

where  $A, B, C$  are geometric coefficients (expressions omitted for brevity; see Clavel 1988 for full derivation).

## Forward Kinematics (Numerical Solution Required)

Determining platform position from motor angles requires solving three simultaneous constraint equations (three spherical shells intersect at platform position). No closed-form solution exists; numerical methods (Newton-Raphson) converge in 3-5 iterations.

**Control implication:** Inverse kinematics computed rapidly in real-time (1-2 mus); forward kinematics used only for trajectory verification.

## Workspace Geometry

Platform workspace is intersection of three spherical shells (each centered on proximal arm end-point):

$$\text{Workspace} = \bigcap_{i=1}^3 \{ \mathbf{P} : R_{min,i} \leq |\mathbf{P} - \mathbf{B}_i| \leq R_{max,i} \}$$

Resulting shape: inverted cone or truncated ellipsoid.

**Typical dimensions:** For base radius 300 mm, platform radius 100 mm, proximal arm 200 mm, distal link 600 mm:

- Workspace diameter: 600-800 mm (at Z = -300 mm)
- Workspace height: 300-400 mm
- Volume: ~0.15 m<sup>3</sup> (15% of robot installation envelope)

## Parallel Kinematic Advantages

1. **High acceleration:** Motors on fixed base (no moving motor mass) + parallel load distribution  
□ 10-15 g achievable
2. **Accuracy amplification:** Motor errors divide across three chains (geometric averaging improves repeatability)
3. **High stiffness:** Closed kinematic loops provide structural rigidity (natural frequency 50-100 Hz vs. 20-40 Hz serial robots)

## Link Design for High-Speed Operation

Distal links experience 10-15 g centripetal acceleration; carbon fiber tubes minimize inertia:

### Material properties:

- Carbon fiber/epoxy: Density 1,600 kg/m<sup>3</sup>, Young's modulus 140 GPa
- Aluminum 6061-T6: Density 2,700 kg/m<sup>3</sup>, Young's modulus 69 GPa

### Specific stiffness comparison:

$$\frac{E}{\rho_{carbon}} = \frac{140 \times 10^9}{1600} = 87.5 \times 10^6 \text{ m}^2/\text{s}^2$$

$$\frac{E}{\rho_{aluminum}} = \frac{69 \times 10^9}{2700} = 25.6 \times 10^6 \text{ m}^2/\text{s}^2$$

Carbon fiber provides 3.4× higher specific stiffness □ enables longer links with lower mass and higher natural frequency.

**Typical link:** OD 20 mm, ID 18 mm carbon tube, length 600 mm, mass 80 g (vs. 250 g aluminum)

## 9.2.4 Architecture Selection Framework

### Performance Trade-offs

Characteristic	Cartesian	SCARA	Delta
<b>Workspace shape</b>	Rectangular	Annular cylinder	Inverted cone
<b>Workspace utilization</b>	70-80%	50-60%	10-15%
<b>Positioning accuracy</b>	+/-0.05 mm	+/-0.02 mm	+/-0.1-0.5 mm
<b>Maximum speed</b>	5-10 m/s	10-15 m/s	15-25 m/s
<b>Payload capacity</b>	50-100+ kg	5-20 kg	0.5-5 kg
<b>Structural complexity</b>	Low (3 linear axes)	Medium (2 rotary + 1 linear)	High (parallel kinematics)
<b>Programming complexity</b>	Low (Cartesian)	Medium (inverse kinematics)	High (calibration-sensitive)
<b>Cost</b>	\$35k-80k	\$50k-120k	\$80k-250k

### Application-Specific Recommendations

**Choose Cartesian when:** - Large/irregular workspace required (>2 m<sup>3</sup>) - Heavy payloads (>20 kg) - Simple programming desired (Cartesian coordinates) - CNC machine tending (extended reach, z-axis clearance)

**Choose SCARA when:** - Compact floor space (overhead mounting) - High speed required (50-150 picks/min) - Precision assembly (<+/-0.02 mm) - Electronics manufacturing (PCB handling, component placement)

**Choose Delta when:** - Extreme speed critical (>200 picks/min) - Lightweight parts (<2 kg) - Overhead pick from conveyor - Food/pharmaceutical packaging (washdown capability)

Mastery of robot architecture—Cartesian beam deflection analysis, SCARA inverse kinematics, Delta parallel workspace geometry—enables engineers to select optimal configuration for application requirements and design custom systems when commercial solutions inadequate.

## References

### 1. Kinematics and Robotics

- Craig, J.J. (2017). *Introduction to Robotics: Mechanics and Control* (4th ed.). Pearson
- Spong, M.W., Hutchinson, S., & Vidyasagar, M. (2020). *Robot Modeling and Control* (2nd ed.). Wiley
- Angeles, J. (2007). *Fundamentals of Robotic Mechanical Systems*. Springer

### 2. Delta Robot Design

- Clavel, R. (1988). "DELTA: A Fast Robot with Parallel Geometry". *18th International Symposium on Industrial Robots*
- Pierrot, F., & Company, O. (1999). "H4: A New Family of 4-DOF Parallel Robots". *IEEE/ASME AIM*
- Gosselin, C., & Angeles, J. (1988). "The Optimum Kinematic Design of a Planar Three-Degree-of-Freedom Parallel Manipulator". *Journal of Mechanisms, Transmissions, and Automation in Design*, 110(1), 35-41

### 3. Structural Analysis

- Young, W.C., Budynas, R.G., & Sadegh, A.M. (2011). *Roark's Formulas for Stress and Strain* (8th ed.). McGraw-Hill
- Timoshenko, S.P., & Gere, J.M. (2009). *Theory of Elastic Stability*. Dover Publications

### 4. Mechanical Components

- THK Linear Motion Systems - Linear Guide Technical Catalog
- HIWIN - Linear Guideway Technical Reference
- Harmonic Drive LLC - Component Set Technical Data
- ISO 281:2007 - Rolling bearings - Dynamic load ratings and rating life

### 5. Material Properties

- ASM Metals Handbook - Aluminum Alloy Properties (6061-T6)
  - Matweb Material Property Database - [www.matweb.com](http://www.matweb.com)
  - Hexcel Carbon Fiber Composites - Technical datasheets
- 

## Module 9 - Pick & Place Robot

End effectors physically interact with parts. Selection depends on part geometry, material, weight, and surface characteristics.

### Vacuum Grippers

**Principles** - Use suction to hold smooth, flat surfaces - Fast pick/release - Gentle on delicate parts

**Components** - Suction cups (bellows, flat, foam) - Vacuum source (venturi or pump) - Vacuum sensors - Solenoid valves

### Force Calculation

$$F = P \times A \times n \times \eta$$

- P = vacuum pressure (kPa)
- A = cup area (mm<sup>2</sup>)

- n = number of cups
- eta = efficiency (0.7-0.9)

### **Vacuum Generation**

Venturi: - Uses compressed air - Fast response - Typical: -60 to -80 kPa

Pump: - Electric rotary vane - Higher vacuum: -80 to -95 kPa - Lower air consumption

**Cup Types** - Bellows: Compensate for height variation - Flat: Maximum grip force - Foam: Conforms to texture

**Applications** - PCB handling - Sheet metal - Glass and flat plastics - Smooth packaging

### **Mechanical Grippers**

**Parallel Jaw Grippers** - Two jaws move in parallel - Pneumatic or electric actuation - Handles irregular shapes

#### **Grip Force Required**

$$F_{\text{grip}} = (m \times a \times SF) / (2 \times \mu)$$

- m = mass (kg)
- a = acceleration (m/s<sup>2</sup>)
- SF = safety factor (2-4)
- mu = friction coefficient

**Jaw Types** - Soft jaws: Urethane padding - Form-fitting: Machined to part - V-block: Centers round parts - Serrated: High friction

#### **Actuation**

Pneumatic: - Fast (0.1-0.3 sec) - Fixed force - Simple control

Electric: - Variable position/force - Encoder feedback - More flexible

**Three-Jaw Grippers** - Self-centering - Good for round parts - Higher complexity

### **Magnetic Grippers**

**Electromagnetic** - Coil around iron core - Controllable on/off - Works only on ferrous materials - Fast pick/release

**Permanent Magnets** - Switchable mechanism - No electrical power - Manual or pneumatic switch

**Applications** - Steel sheet handling - Ferrous parts - High duty cycle operations

### **Specialized End Effectors**

**Needle Grippers** - Pins grip inside holes - For rings, washers - Requires precision alignment

**Hook Grippers** - Catches part features - Simple mechanism

**Adhesive Pads** - Tacky silicone or gel - For difficult materials - Limited lifespan

**Custom 3D Printed** - Rapid prototyping - Part-specific geometry - Lower strength

## Quick-Change Systems

**Tool Changer Interface** - Mechanical coupling - Pneumatic/electrical pass-through - Tool identification sensors

**Benefits** - Handle multiple part types - Reduce changeover time - Automated tool selection

## Sensors

**Part Presence** - Proximity sensors - Optical through-beam - Vacuum pressure switches

**Position Feedback** - Encoders (electric grippers) - Reed switches (pneumatic)

**Force Sensing** - Load cells - Prevent part damage - Verify grip security

## Design Guidelines

**Selection Criteria** - Part surface (smooth vs. textured) - Material (ferrous, plastic, etc.) - Weight and acceleration - Cycle time requirements - Damage tolerance

**Safety Factors** - Vacuum: 10-20× weight - Mechanical grip: 2-4× dynamic force - Magnetic: 5-10× weight

---

**Next:** 9.4 Vision Systems and Sensors

---

## References

### 1. Gripper Manufacturers

- SCHUNK Grippers and Gripping Systems - Technical Catalogs
- SMC Pneumatics - Vacuum Equipment and Grippers
- DESTACO - End Effector Solutions
- Piab Vacuum Technology - Suction Cup Selection Guide

### 2. Engineering Handbooks

- Monkman, G.J. et al. (2007). *Robot Grippers*. Wiley-VCH
- Cutkosky, M.R. & Wright, P.K. (1986). "Modeling Manufacturing Grips and Robotic Hands"

### 3. Vacuum Technology

- SMC Vacuum Technology Reference Guide
- Piab Vacuum Handbook
- ISO 6358:2013 - Pneumatic components - Flow characteristics

### 4. Material Properties

- Engineering friction coefficient tables
  - MatWeb Material Property Database
  - Permanent magnet specifications - K&J Magnetics
-

## Module 9 - Pick & Place Robot

Vision systems and sensors enable robots to locate parts, verify placement, inspect quality, and adapt to position variations.

### Machine Vision Fundamentals

**Purpose** - Part localization (X, Y, rotation) - Presence/absence detection - Quality inspection (defects, orientation) - Barcode/OCR reading - Measurement and gauging

**System Components** - Camera (CCD or CMOS sensor) - Lens (fixed focal length or adjustable) - Lighting (LED ring, backlight, dome) - Image processing computer - Software (OpenCV, Halcon, Cognex VisionPro)

### Camera Selection

#### Sensor Types

CCD (Charge-Coupled Device): - High image quality - Better low-light performance - Higher cost - Slower frame rates

CMOS (Complementary Metal-Oxide Semiconductor): - Lower cost - Faster frame rates - Lower power consumption - Common in modern systems

**Resolution** - VGA: 640×480 (sufficient for simple tasks) - 1MP: 1280×1024 (general purpose) - 5MP+: High-precision inspection

Field of view calculation:

$$FOV = (\text{sensor\_size} / \text{focal\_length}) \times \text{working\_distance}$$

**Frame Rate** - 30 fps: Standard for pick-and-place - 60+ fps: High-speed applications - Triggered capture: Synchronized with motion

**Interface** - USB 3.0: Simple, moderate speed - GigE: Longer cables, industrial standard - Camera Link: High bandwidth, expensive

### Lens Selection

**Focal Length** - Short (8-16mm): Wide field of view - Medium (25-50mm): General purpose - Long (>50mm): Narrow field, distant objects

**Aperture (f-stop)** - Low f-number (f/1.4-f/2.8): More light, shallow depth of field - High f-number (f/8-f/16): Less light, greater depth of field

**Depth of Field** Critical for 3D objects:

$$DOF \approx (2 \times N \times C \times d^2) / f^2$$

- N = f-stop number
- C = circle of confusion
- d = working distance
- f = focal length



## Lighting Techniques

### Lighting Types

Ring Light: - Mounts around lens - Even illumination - Reduces shadows - General purpose

Backlight: - Part silhouette - Edge detection - Hole/slot inspection - High contrast

Dome Light: - Diffuse illumination - Eliminates reflections - Good for shiny surfaces

Structured Light: - Projects pattern (lines, grid) - 3D measurement - Height profiling

**Color** - White: General purpose - Red (630nm): High contrast, deep penetration - Blue (470nm): High resolution - IR (850nm+): See through some plastics

**Polarization** - Reduces glare from reflective surfaces - Polarizing filters on light and camera

### Vision Processing

**Image Acquisition** - Triggered by robot position or external signal - Consistent lighting critical - Exposure time vs. motion blur trade-off

**Pre-Processing** - Noise reduction (Gaussian blur, median filter) - Contrast enhancement - Color space conversion (RGB to grayscale or HSV)

### Feature Extraction

Edge Detection: - Sobel, Canny algorithms - Finds part boundaries

Blob Analysis: - Identifies connected regions - Measures area, centroid, orientation

Template Matching: - Correlates image with known pattern - Returns position and rotation - Sensitive to scale and lighting changes

**Pattern Recognition** - Feature-based (SIFT, SURF, ORB) - Robust to rotation, scale, partial occlusion - Higher computational cost

**Calibration** - Camera intrinsics (lens distortion) - Camera-to-robot transformation - Pixel-to-mm conversion

Calibration grid method:

$\text{mm\_per\_pixel} = \text{known\_distance} / \text{pixel\_distance}$

### Common Vision Tasks

**Part Localization** 1. Acquire image 2. Find part features (edges, blobs) 3. Calculate centroid and rotation 4. Transform to robot coordinates 5. Update pick position

**Presence Detection** - Simple threshold or blob count - Verify part before pick attempt - Confirm placement after release

**Orientation Verification** - Detect asymmetric features - Ensure correct part rotation - Reject or correct misaligned parts

**Quality Inspection** - Measure dimensions - Detect defects (scratches, dents) - Color verification - Label/markings presence

## 2D Vision System Example

**PCB Pick-and-Place** - Camera: 1MP CMOS, USB 3.0 - Lens: 16mm, f/2.8 - Lighting: White LED ring light - Field of view: 100×100 mm - Resolution: 0.1 mm/pixel - Processing: OpenCV on Raspberry Pi - Cycle time: 200ms per image

**Processing Steps** 1. Capture image when part enters FOV 2. Convert to grayscale 3. Threshold to binary image 4. Find largest blob (PCB outline) 5. Calculate centroid and orientation 6. Transform to robot coordinates 7. Send offset to motion controller

## 3D Vision

### Techniques

**Laser Triangulation:** - Laser line projected on part - Camera observes line deflection - Calculates height profile

**Stereo Vision:** - Two cameras at known separation - Matches features between images - Calculates depth from disparity

**Structured Light:** - Projects pattern (stripes, dots) - Analyzes pattern distortion - Generates 3D point cloud

**Time-of-Flight:** - Measures light travel time - Direct depth measurement - Lower resolution than other methods

**Applications** - Bin picking (random part orientation) - Height measurement - Volume calculation - Complex part recognition

## Proximity Sensors

**Inductive Sensors** - Detect ferrous and non-ferrous metals - Sensing distance: 1-20mm - Fast response (<1ms) - Use: Metal part presence

**Capacitive Sensors** - Detect any material (different sensitivity) - Sensing distance: 5-30mm - Use: Plastic, liquid, powder detection

### Photoelectric Sensors

**Through-Beam:** - Separate emitter and receiver - Long range (up to 30m) - Most reliable - Use: Part presence, counting

**Retro-Reflective:** - Reflector returns light to sensor - Medium range (up to 5m) - Single mounting location

**Diffuse:** - Detects object reflection - Short range (<1m) - Simple installation

**Ultrasonic Sensors** - Measures distance via sound echo - Range: 50mm to 5m - Immune to color/transparency - Use: Level sensing, distance measurement

**Laser Distance Sensors** - Precise distance measurement - Range: 50mm to 300m - High accuracy (+/-1mm or better) - Use: Height profiling, positioning

## Force and Torque Sensors

**Load Cells** - Strain gauge based - Measure grip force - Prevent part damage - Verify secure grip

**6-Axis F/T Sensors** - Measure force (Fx, Fy, Fz) - Measure torque (Tx, Ty, Tz) - Enable compliant assembly - High cost (\$2k-\$10k+)

**Applications** - Peg-in-hole insertion - Contact detection - Quality assurance (press fits)

## Sensor Integration

**Digital I/O** - Simple presence/absence sensors - 24V industrial standard - Direct connection to PLC or controller

**Analog Input** - Distance, force, pressure sensors - 0-10V or 4-20mA signals - Requires ADC on controller

**Network Communication** - Vision systems via Ethernet - Modbus TCP, EtherCAT, PROFINET - Higher bandwidth, more data

## Vision Software

**Open Source** - OpenCV: Comprehensive library, C++/Python - SimpleCV: Python wrapper for OpenCV - Scikit-image: Python image processing

**Commercial** - Cognex VisionPro: Industry standard, expensive - National Instruments Vision: LabVIEW integration - Halcon: Powerful, MVTec software

**DIY Implementation** Python + OpenCV example:

```
import cv2
import numpy as np

# Capture image
cap = cv2.VideoCapture(0)
ret, frame = cap.read()

# Convert to grayscale
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

# Threshold
_, binary = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)

# Find contours
contours, _ = cv2.findContours(binary, cv2.RETR_EXTERNAL,
                                cv2.CHAIN_APPROX_SIMPLE)

# Get largest contour
largest = max(contours, key=cv2.contourArea)

# Calculate centroid
```

```
M = cv2.moments(largest)
cx = int(M['m10']/M['m00'])
cy = int(M['m01']/M['m00'])
```

```
# Send to robot controller
```

## Calibration and Accuracy

**Hand-Eye Calibration** - Determines transformation between camera and robot - Move robot to known positions - Capture images of calibration target - Solve transformation matrix

**Performance Metrics** - Repeatability: +/-0.1-0.5mm typical for 2D vision - Accuracy: Depends on calibration quality - Cycle time: 100-500ms per image

---

**Next:** 9.5 Motion Control and Path Planning

---

## References

### 1. Machine Vision Software

- Bradski, G. & Kaehler, A. (2008). *Learning OpenCV*. O'Reilly
- OpenCV Documentation - [opencv.org](http://opencv.org)
- Halcon Machine Vision - MVTec Software GmbH

### 2. Camera and Optics

- FLIR (Basler) Industrial Camera Selection Guide
- Edmund Optics - Imaging Optics Technical Library
- GigE Vision and Camera Link Standards

### 3. Image Processing

- Gonzalez, R.C. & Woods, R.E. (2018). *Digital Image Processing*. Pearson
- Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer

### 4. Calibration

- Zhang, Z. (2000). "Camera Calibration Technique". *IEEE Trans. PAMI*
- 

## Module 9 - Pick & Place Robot

Motion control determines how the robot moves between positions efficiently, accurately, and safely.

## Motion Controllers

### Controller Types

Dedicated Robot Controller: - Specialized for robot kinematics - Built-in trajectory planning - Examples: KUKA KRC, ABB IRC5, Universal Robots - Commercial systems, proprietary interfaces

CNC Controller (LinuxCNC, Mach3): - General-purpose motion control - G-code programming - Open source options available - Requires kinematics configuration

PLC + Motion Modules: - Allen-Bradley, Siemens, Omron - Integrated with factory automation - Higher cost

Standalone Motion Controller: - Galil, Delta Tau, ACS - Flexible programming - Multi-axis coordination

**DIY Options** - LinuxCNC with custom kinematics component - GRBL (limited to 3 axes, Cartesian only) - Marlin (3D printer firmware, adapted) - Custom Arduino/Raspberry Pi solutions

## Coordinate Systems

**Machine Coordinates** - Fixed reference frame - Motor positions or joint angles - Controller operates in this space

**World Coordinates** - User-defined reference frame - Typically part fixture or table origin - Programming done in this space

**Tool Coordinates** - Origin at tool center point (TCP) - Accounts for end effector geometry - Critical for accurate positioning

## Transformations

$\text{Position\_machine} = \text{Transform\_world\_to\_machine} \times \text{Position\_world}$

$\text{Position\_tool} = \text{Transform\_tool\_offset} \times \text{Position\_machine}$

## Kinematics Algorithms

**Forward Kinematics** Given: Joint positions (motor angles/positions) Find: Tool position in world space

Cartesian:

$X = X_{\text{motor}}$

$Y = Y_{\text{motor}}$

$Z = Z_{\text{motor}}$

SCARA:

$X = L1 \times \cos(\theta_1) + L2 \times \cos(\theta_1 + \theta_2)$

$Y = L1 \times \sin(\theta_1) + L2 \times \sin(\theta_1 + \theta_2)$

$Z = Z_{\text{motor}}$

**Inverse Kinematics** Given: Desired tool position Find: Required joint positions

Cartesian: Trivial (1:1 mapping)

SCARA: Analytical solution (see Section 9.2)

Delta: Complex analytical or numerical solution

**Singularities** Configurations where small tool motions require infinite joint velocities: - SCARA: Fully extended or folded - Delta: Near workspace boundaries - Solution: Avoid via path planning or joint limits

## Trajectory Planning

**Point-to-Point (PTP) Motion** - Moves from start to end position - Path is not controlled (joint space interpolation) - Fastest for simple pick-and-place - Each joint accelerates/decelerates independently

**Linear (Cartesian) Motion** - Tool moves in straight line - Requires inverse kinematics at each timestep - Slower than PTP - Used when path matters (avoiding obstacles)

**Circular/Spline Motion** - Tool follows arc or smooth curve - Complex planning - Uncommon in pick-and-place

## Velocity Profiling

**Trapezoidal Profile** - Constant acceleration, constant velocity, constant deceleration - Simple to implement - Discontinuous acceleration (jerk)

Phase 1: Accel  $v = a \times t$

Phase 2: Constant  $v = v_{\max}$

Phase 3: Decel  $v = v_{\max} - a \times t$

**S-Curve Profile** - Smooth acceleration (limited jerk) - Reduces vibration and wear - More complex calculation - Better for high-speed systems

## Motion Time Calculation

Trapezoidal profile:

If distance allows reaching  $v_{\max}$ :

$$t_{\text{total}} = d/v_{\max} + v_{\max}/a$$

If too short (triangular profile):

$$t_{\text{total}} = 2 \times \sqrt{d/a}$$

## Multi-Axis Coordination

**Synchronized Motion** - All axes reach end position simultaneously - Scale velocities to slowest axis - Smooth coordinated motion

**Blending** - Round corners between move segments - Reduces cycle time (no complete stop) - Trade-off: path accuracy vs. speed

**Look-Ahead** - Controller examines upcoming moves - Optimizes velocity through corners - Prevents unnecessary deceleration

## Control Loops

**Position Control** - PID feedback loop - Typical update rate: 1-2 kHz - Tuning parameters:  $K_p$ ,  $K_i$ ,  $K_d$

**Velocity Control** - Inner loop for servo drives - Faster update rate (10-20 kHz) - Reduces following error

**Torque/Current Control** - Innermost loop in servo drive - Limits motor stress - Enables force control

## Path Planning Strategies

### Pick-and-Place Sequence

Standard sequence: 1. Move to approach position (above part) 2. Lower to pick position 3. Activate gripper/vacuum 4. Raise to clearance height 5. Move to place approach position 6. Lower to place position 7. Release gripper/vacuum 8. Raise to clearance height 9. Return to home or next pick

**Clearance Heights** - Safe Z height to avoid collisions - Typically 50-100mm above highest obstacle - Balance safety vs. cycle time

### Approach Strategies

Z-First: - Move Z to clearance - Move XY to target - Lower Z to final position - Safe, slower

XYZ-Simultaneous: - Interpolate all axes together - Faster - Requires collision-free path

**Obstacle Avoidance** - Define keep-out zones in workspace - Path planner routes around obstacles - Simple: Use via points - Advanced: A\* or RRT path planning algorithms

## Homing and Calibration

**Homing Sequence** 1. Move axes to home switches at slow speed 2. Back off and re-approach at creep speed 3. Set machine zero position 4. Move to known calibration position

**Absolute Encoders** - Eliminate need for homing - Position known on power-up - Higher cost

### Calibration Procedures

Tool Center Point (TCP): - Jog tool to known reference point - Measure offset from wrist to tool tip - Enter offset in controller

Vision Calibration: - Place calibration grid in camera FOV - Robot moves to grid points - Calculate camera-to-robot transformation

**Repeatability Testing** - Move to position 100+ times - Measure variation with indicator or CMM - Should be <0.05mm for Cartesian, <0.02mm for SCARA

## Motion Programming

### G-Code (CNC-Style)

```
G0 X100 Y50 Z50      ; Rapid to approach
G1 Z10 F500           ; Feed to pick height
M42 P1                ; Activate vacuum
G4 P0.5               ; Dwell 0.5 sec
G0 Z50                ; Retract
G0 X200 Y150          ; Move to place
G1 Z15                ; Lower to place
```

```
M43 P1          ; Release vacuum
G0 Z50          ; Retract
```

**Teach Pendant** - Jog robot to positions - Record positions to program - Playback sequence - Common on commercial robots

### Scripting (Python, JavaScript)

```
robot.move_to(x=100, y=50, z=50, speed=1000)
robot.move_to(z=10, speed=100)
robot.gripper_on()
time.sleep(0.5)
robot.move_to(z=50)
robot.move_to(x=200, y=150)
```

**Graphical Programming** - Drag-and-drop blocks (LabVIEW, Blockly) - Visual sequence design - Easier for non-programmers

### Performance Optimization

**Minimize Air Time** - Reduce unnecessary Z moves - Optimize travel paths - Use blending/rounding

**Maximize Acceleration** - Reduce moving mass - Stiffen structure - Tune control loops

**Parallel Operations** - Gripper activation during motion - Vision processing during previous cycle - Multi-robot coordination

**Cycle Time Analysis** Break down into components: - Move time: 60-70% - Gripper actuation: 10-15% - Vision processing: 10-20% - Settling time: 5-10%

Focus optimization on largest contributors.

### Real-Time Control

**Deterministic Timing** - Real-time operating system (RTOS or Linux PREEMPT\_RT) - Guaranteed response times - Critical for servo control

**Trajectory Generation** - Pre-compute paths - Stream points to controller - Buffer to prevent starvation

**Communication Latency** - Minimize overhead in control loops - Direct memory access (DMA) for high-speed I/O - Avoid polling delays

---

**Next:** 9.6 CNC Integration

---

### References

#### 1. Motion Control Theory

- Craig, J.J. (2017). *Introduction to Robotics: Mechanics and Control*. Pearson



- Spong, M.W. et al. (2005). *Robot Modeling and Control*. Wiley
- LaValle, S.M. (2006). *Planning Algorithms*. Cambridge University Press

## 2. Trajectory Planning

- Lynch, K.M. & Park, F.C. (2017). *Modern Robotics*. Cambridge University Press
- Siciliano, B. et al. (2009). *Robotics: Modelling, Planning and Control*. Springer

## 3. Real-Time Control

- LinuxCNC Documentation - Real-time motion control
- PREEMPT\_RT Linux Kernel Documentation
- EtherCAT Technology Group - Real-time Ethernet

## 4. Control Systems

- Åström, K.J. & Murray, R.M. (2008). *Feedback Systems*. Princeton University Press

# Module 9 - Pick & Place Robot

Integrating pick and place robots with CNC machines creates automated manufacturing cells that reduce labor, increase throughput, and enable lights-out operation.

## Integration Architectures

**Standalone Robot Controller** - Separate controller for robot - Communication via digital I/O or network - CNC and robot coordinate via handshaking signals - Most flexible, higher complexity

**Integrated CNC Control** - Robot axes added to CNC controller - Unified programming (G-code or similar) - Simpler coordination - Requires controller support for additional axes

**PLC Coordination** - PLC orchestrates both CNC and robot - Ladder logic or structured text programming - Common in industrial automation - Good for complex multi-machine cells

## Communication Protocols

### Digital I/O (Handshaking)

Simple signals exchanged between CNC and robot:

CNC outputs: - Cycle complete - Door open - Ready for load/unload - Part present in fixture

Robot outputs: - Part loaded - Part unloaded - Robot busy - Error/fault

Typical sequence: 1. CNC finishes part, signals "cycle complete" 2. Robot waits for "door open" signal 3. Robot signals "robot busy", enters machine 4. Robot unloads part, loads new stock 5. Robot exits, signals "part loaded" 6. CNC closes door, starts cycle

**Serial Communication (RS-232)** - Text-based commands - Simple protocol (ASCII strings) - Limited to 15m distance - Low cost, easy debugging

Example protocol:

CNC → Robot: "READY\r\n"

Robot → CNC: "LOADING\r\n"

Robot → CNC: "COMPLETE\r\n"

CNC → Robot: "START\r\n"

### **Ethernet Protocols**

Modbus TCP: - Industrial standard - Read/write registers and coils - Many PLCs and controllers support - 100 Mbps, reliable

EtherCAT: - Real-time deterministic communication - Microsecond synchronization - Requires compatible hardware - Excellent for coordinated motion

Ethernet/IP: - Used in Allen-Bradley systems - CIP (Common Industrial Protocol) - Good integration with PLCs

**Fieldbus (Legacy)** - PROFIBUS, DeviceNet, CANopen - Being replaced by Ethernet solutions - May be present in older equipment

### **Machine Door Control**

**Pneumatic Doors** - Robot triggers solenoid valve - Opens/closes door automatically - Interlock prevents robot entry when closed

**Manual Doors with Interlocks** - Operator opens door manually - Limit switch signals robot "door open" - Safety rated interlock (guard switch)

**Electronic Locks** - Solenoid or electromagnetic lock - Released only when safe (CNC stopped, robot clear) - Meets safety standards (ISO 14119)

### **Fixture and Part Handling**

#### **Part Holding in CNC**

Vise: - Manual or pneumatic - Robot places part, triggers vise close - Repeatability critical for part location

Chuck (Lathe): - Pneumatic or hydraulic actuation - Robot loads bar stock or individual parts - Air chuck signal controlled by CNC

Fixture Plate: - Custom fixture for part geometry - Pneumatic clamps or vacuum - Multiple parts per fixture (family parts)

**Part Orientation** - Robot may need to flip part between OP1 and OP2 - Fixture design must accommodate robot gripper - Clearance for gripper access

### **Workpiece Identification**

**Barcode/QR Code** - Vision system reads code on part - Tracks part serial number - Selects appropriate CNC program - Quality traceability

**RFID Tags** - Embedded tag in fixture or pallet - Identifies part family or program number - Automatic program selection

**Vision Inspection** - Camera verifies part type - Dimensional check before machining - Wrong part detection

## Multi-Machine Cells

**Single Robot, Multiple CNCs** - Robot services 2-4 machines - Optimizes machine utilization - Requires cycle time balancing

## Scheduling Algorithms

Fixed Sequence: - Robot visits machines in order - Simple, predictable - May have idle time

Priority-Based: - Service machine with longest wait first - Dynamic scheduling - Better utilization

**Buffer Stations** - Queue of raw stock - Queue of finished parts - Intermediate storage between operations

## LinuxCNC Integration

**Custom Kinematics Component** LinuxCNC supports adding custom robot kinematics as HAL component.

Example for SCARA (C code):

```
// Inverse kinematics function
int kinematicsInverse(const EmcPose *pos, double *joints) {
    double x = pos->tran.x;
    double y = pos->tran.y;
    double z = pos->tran.z;

    // Calculate joint angles
    double c2 = (x*x + y*y - L1*L1 - L2*L2) / (2*L1*L2);
    double theta2 = atan2(-sqrt(1-c2*c2), c2); // Elbow down
    double theta1 = atan2(y,x) - atan2(L2*sin(theta2), L1+L2*cos(theta2));

    joints[0] = theta1;
    joints[1] = theta2;
    joints[2] = z;

    return 0;
}
```

## HAL Configuration

```
# Load kinematics component
loadrt trivkins coordinates=XYZ
# Or custom: loadrt scara_kins

# Connect motion controller to drives
net j0-pos-cmd joint.0.motor-pos-cmd => servo.0.position-cmd
net j1-pos-cmd joint.1.motor-pos-cmd => servo.1.position-cmd
net j2-pos-cmd joint.2.motor-pos-cmd => servo.2.position-cmd

# Connect I/O for gripper
net gripper-on motion.digital-out-00 => parport.0.pin-01-out
```

## G-Code Programming

```
G0 X100 Y50 Z50      ; Move to pick position
G1 Z10 F500           ; Lower to part
M64 P0                ; Turn on gripper (digital out 0)
G4 P0.5               ; Wait 0.5 sec
G0 Z50                ; Raise
G0 X200 Y150 Z50      ; Move to place
G1 Z15                ; Lower
M65 P0                ; Turn off gripper
G0 Z50                ; Raise
```

## Safety Integration

**Interlocks** - Robot cannot enter while CNC spindle running - CNC cannot start while robot inside  
- Emergency stop halts both systems

**Safety-Rated Signals** - Use safety PLCs or relays - Redundant monitoring - Meets Category 3 or 4 per ISO 13849

**Light Curtains** - Detects operator entry - Stops robot and CNC - Type 4 sensors per IEC 61496

## Conveyor Integration

**Part Input** - Conveyor brings raw stock to robot - Sensor detects part presence - Robot picks from moving conveyor (tracking) or stopped

**Part Output** - Robot places finished parts on conveyor - Accumulation area for batch pickup - Automatic boxing/palletizing

**Synchronization** - Encoder on conveyor belt - Robot tracks moving part position - Pick on the fly (advanced)

## Automated Tool Changing

**CNC Tool Changer** - Standard on many CNC machines - Robot does not need to handle tools - CNC controls tool selection

**Robot Tool Changer** - Robot swaps end effectors - Handle different part sizes/types - Quick-change interface

## Cell Controller Software

**Supervisory Control** - PC or industrial controller - Coordinates CNC, robot, conveyors - Production scheduling - Data logging and reporting

**Communication** - OPC UA (modern standard) - MTConnect (machine tool data) - MQTT (IoT messaging)

**User Interface** - HMI (Human-Machine Interface) - Production status display - Manual override controls - Error diagnostics

## Example Cell Configurations

**Single CNC + Robot** - Robot loads raw stock from pallet - Unloads finished part to conveyor - Simple handshaking via I/O - Typical cycle: CNC 3 min, robot 20 sec

**Two CNCs + Robot** - Robot alternates between machines - OP1 on CNC1, OP2 on CNC2 - Robot flips part between operations - Double throughput vs. single machine

**CNC + Inspection + Robot** - Robot loads CNC - After machining, robot moves part to CMM - Inspection results logged - Accept/reject decision - Accepted parts to conveyor, rejected to bin

## Cycle Time Optimization

**Overlap Operations** - Start robot motion while CNC finishing - Trigger robot early (before “cycle complete”) - Reduce idle time

**Predictive Scheduling** - Estimate CNC completion time - Robot arrives just in time - Minimizes wait time

**Parallel Processing** - While CNC machines part, robot services other machines - Vision processing during robot motion - Gripper activation during approach

## Data Collection and Traceability

**Part Tracking** - Log serial numbers, timestamps - Track which machine, which program - Quality data association

**Machine Utilization** - Spindle time vs. idle time - Downtime reasons (waiting for parts, tool changes) - OEE (Overall Equipment Effectiveness)

**Quality Data** - In-process inspection results - SPC (Statistical Process Control) charting - Alarm/reject logs

---

**Next:** 9.7 Programming and Path Planning

---

## References

### 1. Communication Protocols

- Modbus Protocol Specification - Modbus.org
- OPC UA Specification - OPC Foundation
- MTConnect Standard - MTConnect Institute
- MQTT Protocol - OASIS Standard

### 2. Industrial Networking

- EtherNet/IP Specification - ODVA
- PROFINET IO Specification - PROFIBUS International
- EtherCAT Technology Group Documentation

### 3. CNC Integration

- LinuxCNC HAL (Hardware Abstraction Layer) Documentation

- Mach3/Mach4 Plugin Development Guide
- FANUC FOCAS Library Documentation

#### 4. MES/ERP Systems

- ISA-95 Enterprise-Control System Integration Standard
  - Groover, M.P. (2014). *Automation and Production Systems*. Pearson
- 

## Module 9 - Pick & Place Robot

Robot programming defines the sequence of motions, gripper actions, and decision logic for automated operation.

### Programming Methods

**Teach Pendant** - Handheld controller with jog buttons - Move robot to desired positions - Record positions to program memory - Common on commercial robots - Intuitive for non-programmers

Advantages: - No CAD or simulation needed - Immediate feedback - Easy to adjust positions

Disadvantages: - Requires physical robot access - Time-consuming for complex sequences - Difficult to duplicate programs

### Text-Based Programming

G-Code (CNC-style):

```
01000 (Pick and Place Program)
G0 X100 Y50 Z100          ; Move to safe position
G1 Z10 F500               ; Lower to pick height
M64 P0                    ; Gripper on
G4 P0.5                   ; Dwell 500ms
G0 Z100                   ; Retract
G0 X200 Y150              ; Move to place location
G1 Z15                    ; Lower to place
M65 P0                    ; Gripper off
G0 Z100                   ; Retract
M30                       ; Program end
```

Robot-Specific Language (example):

```
MOVE P1 SPEED=100
GRIP ON
WAIT 0.5
MOVE P2 SPEED=80
GRIP OFF
```

Python/Scripting:

```
robot.move(x=100, y=50, z=100, speed=1000)
robot.move(z=10, speed=500)
robot.gripper_on()
```

```

time.sleep(0.5)
robot.move(z=100)
robot.move(x=200, y=150)
robot.move(z=15, speed=500)
robot.gripper_off()
robot.move(z=100)

```

**Graphical Programming** - Drag-and-drop blocks (Blockly, LabVIEW) - Visual flowcharts - Easier for complex logic - Good for non-programmers

**Offline Programming** - Simulate robot in CAD/CAM software - Generate programs without robot access - Verify collisions and reach - Examples: RoboDK, ABB RobotStudio, KUKA Sim

## Program Structure

### Main Sequence

#### INIT:

- Home robot
- Initialize gripper
- Reset counters

#### LOOP:

- Call vision routine
- Calculate pick position
- Execute pick sequence
- Execute place sequence
- Increment counter
- Check for errors
- Repeat

#### ERROR\_HANDLER:

- Stop motion
- Release gripper
- Move to safe position
- Signal fault

**Subroutines** Modular programming improves maintainability:

```

01000 (Main program)
M98 P1001 (Call pick routine)
M98 P1002 (Call place routine)
M30

```

```

01001 (Pick routine)
G0 X#100 Y#101 Z50
G1 Z#102 F500
M64 P0
G4 P0.5
G0 Z50

```

M99

```
01002 (Place routine)
G0 X#103 Y#104 Z50
G1 Z#105
M65 P0
G0 Z50
M99
```

## Position Management

**Absolute Positioning** - Coordinates specified relative to machine origin - Consistent, repeatable  
- Programs not portable between robots

**Relative Positioning** - Offsets from current position - Useful for incremental moves - Accumulates errors if not managed

**Named Positions** Define positions as variables:

```
PICK_APPROACH = (100, 50, 100)
PICK_POSITION = (100, 50, 10)
PLACE_APPROACH = (200, 150, 100)
PLACE_POSITION = (200, 150, 15)
```

**Position Arrays** Handle multiple locations:

```
pick_positions = [
    (100, 50, 10),
    (120, 50, 10),
    (140, 50, 10),
]

for pos in pick_positions:
    robot.move(*pos)
    robot.pick()
```

## Vision-Guided Programming

**Vision Offset Correction** 1. Move to nominal pick position 2. Trigger camera capture 3. Vision system calculates offset ( $\Delta X$ ,  $\Delta Y$ ,  $\Delta \theta$ ) 4. Adjust pick position by offset 5. Execute pick

```
def vision_pick(nominal_x, nominal_y, z):
    # Move to vision position
    robot.move(nominal_x, nominal_y, z + 50)

    # Get vision offset
    offset = vision_system.get_offset()

    # Apply offset
    actual_x = nominal_x + offset.x
    actual_y = nominal_y + offset.y
```



```

actual_rotation = offset.rotation

# Adjust gripper rotation if available
robot.set_rotation(actual_rotation)

# Execute pick
robot.move(actual_x, actual_y, z)
robot.gripper_on()
robot.move(z=z+50)

```

**Dynamic Part Tracking** For moving conveyors:

```

def conveyor_pick():
    # Wait for part detection
    while not conveyor_sensor.part_detected():
        time.sleep(0.01)

    # Get current part position
    part_pos = conveyor.get_part_position()

    # Calculate intercept point
    intercept = calculate_intercept(part_pos, conveyor_speed, robot_speed)

    # Track and pick
    robot.track_conveyor(intercept, conveyor_speed)
    robot.gripper_on()
    robot.stop_tracking()

```

## Error Handling and Recovery

**Common Errors** - Gripper failed to grip (no vacuum, jaws blocked) - Part not found by vision - Motion limit exceeded - Communication timeout - Safety stop triggered

### Error Detection

```

def safe_pick(x, y, z):
    robot.move(x, y, z)
    robot.gripper_on()
    time.sleep(0.3)

    # Check if part gripped
    if not gripper.has_part():
        # No part detected
        robot.gripper_off()
        robot.move(z=z+50)
        raise PickError("Failed to grip part")

    # Success
    robot.move(z=z+50)

```

## Recovery Strategies

Retry:

```
max_retries = 3
for attempt in range(max_retries):
    try:
        safe_pick(x, y, z)
        break # Success
    except PickError:
        if attempt == max_retries - 1:
            signal_fault()
            stop_system()
```

Skip and Continue:

```
try:
    safe_pick(x, y, z)
except PickError:
    log_error("Missed pick at position X{} Y{}".format(x, y))
    increment_miss_counter()
    continue # Move to next part
```

Operator Intervention:

```
except PickError:
    pause_system()
    display_message("Pick failed. Please check and press continue.")
    wait_for_operator_confirmation()
    resume_system()
```

## Advanced Programming Techniques

**Array Processing** Pick from grid of parts:

```
for row in range(num_rows):
    for col in range(num_cols):
        x = start_x + col * spacing_x
        y = start_y + row * spacing_y
        pick_and_place(x, y, pick_z, place_x, place_y, place_z)
```

**Palletizing** Stack parts in layers:

```
def palletize(part_num):
    layer = part_num // parts_per_layer
    position_in_layer = part_num % parts_per_layer

    row = position_in_layer // parts_per_row
    col = position_in_layer % parts_per_row

    x = pallet_x + col * part_spacing_x
    y = pallet_y + row * part_spacing_y
```

```

z = pallet_z + layer * part_height

robot.place(x, y, z)

```

### Conditional Logic

```

if part_weight > 2.0:
    use_large_gripper()
else:
    use_small_gripper()

if vision.part_type == "TYPE_A":
    place_position = bin_a
elif vision.part_type == "TYPE_B":
    place_position = bin_b
else:
    place_position = reject_bin

```

**State Machines** Manage complex sequences:

```

state = "IDLE"

while True:
    if state == "IDLE":
        if part_available():
            state = "PICK_APPROACH"

    elif state == "PICK_APPROACH":
        move_to_pick_approach()
        state = "PICK"

    elif state == "PICK":
        execute_pick()
        if gripper.has_part():
            state = "PLACE_APPROACH"
        else:
            state = "ERROR"

    # ... additional states

```

### Simulation and Testing

**Offline Simulation** - Test programs before running on real robot - Verify reach and avoid collisions  
 - Optimize cycle times - Train operators

**Dry Run Mode** - Execute program at slow speed - No gripper activation - Verify motion paths -  
 Check clearances

**Single-Step Mode** - Execute one command at a time - Verify each position - Debug logic - Safe  
 for initial testing

## Program Optimization

**Reduce Cycle Time** - Minimize Z-axis travel - Use blended moves (round corners) - Overlap gripper actuation with motion - Optimize acceleration profiles

**Reduce Code Complexity** - Use subroutines for repeated sequences - Parameterize positions (variables vs. hard-coded) - Comment code clearly - Version control for programs

### Example: Before Optimization

```
G0 X100 Y50 Z100
G0 Z10
M64 P0
G4 P0.5
G0 Z100
G0 X200 Y150
G0 Z15
M65 P0
G0 Z100
```

### After Optimization

```
G0 X100 Y50 Z100
G1 Z10 F1000
M64 P0 ; Start gripper activation
G0 Z50 ; Raise while gripper activates
G4 P0.2 ; Brief dwell (reduced)
G0 X200 Y150 Z50 ; Move to place approach
G1 Z15 F1000
M65 P0 ; Release gripper
G0 Z50 (not Z100) ; Lower clearance height
```

Cycle time reduced by ~30%.

## Documentation

### Program Header

```
(Program: PICK_PLACE_PCB_V2)
(Date: 2025-11-04)
(Author: Engineer Name)
(Description: Pick PCBs from tray, place in fixture)
(Pick location: Tray at X0 Y0)
(Place location: Fixture at X300 Y200)
(Gripper: Vacuum, 2x 15mm cups)
```

### Inline Comments

```
# Calculate pick position from tray index
x = tray_origin_x + (part_index % tray_cols) * part_spacing
y = tray_origin_y + (part_index // tray_cols) * part_spacing
```

```
# Vision offset correction applied here
offset = vision.get_offset()
x += offset.x
y += offset.y
```

---

**Next:** 9.8 Safety Systems

---

## References

### 1. Robot Programming Languages

- ABB RAPID Programming Manual
- FANUC Karel and TP Programming Reference
- KUKA Robot Language (KRL) Manual
- Universal Robots URScript Manual

### 2. Software Development

- ROS (Robot Operating System) Documentation - [ros.org](http://ros.org)
- MoveIt Motion Planning Framework - [moveit.ros.org](http://moveit.ros.org)
- Python Robotics Toolbox

### 3. Path Planning

- LaValle, S.M. (2006). *Planning Algorithms*. Cambridge University Press
- Latombe, J.C. (1991). *Robot Motion Planning*. Kluwer Academic

### 4. Error Handling

- IEC 61508 - Functional Safety of Electrical Systems
  - Software exception handling best practices
- 

## Module 9 - Pick & Place Robot

Pick and place robots present serious hazards including crushing, impact, and entanglement. Proper safety design is mandatory for legal compliance and personnel protection.

### Applicable Standards

#### International Standards

ISO 12100: Safety of machinery - General principles - Risk assessment methodology - Hazard identification and mitigation - Design inherently safe first, then guard, then warn

ISO 10218-1/2: Robots and robotic devices - Safety requirements - Part 1: Industrial robots - Part 2: Robot systems and integration - Mandatory for robot systems in most countries

ISO 13849-1: Safety of machinery - Safety-related parts of control systems - Performance levels (PLa through PLe) - Category architecture (B, 1, 2, 3, 4) - Reliability calculations

ISO 13850: Emergency stop - Principles for design - Category 0 (immediate power removal) or Category 1 (controlled stop then power removal) - Red mushroom button on yellow background - Mechanical latching (requires manual reset)

### **North American Standards**

ANSI/RIA R15.06: Industrial robots and robot systems - Safety requirements - Risk assessment - Safeguarding methods - Collaborative robot requirements (R15.06-2012 Annex G, superseded by ISO/TS 15066)

OSHA Regulations: - 29 CFR 1910.212: Machine guarding - 29 CFR 1910.147: Lockout/tagout

### **European Directives**

Machinery Directive 2006/42/EC: - CE marking requirement - Essential health and safety requirements - Technical construction file

## **Risk Assessment**

### **Hazard Identification**

Mechanical Hazards: - Crushing between robot and fixed objects - Impact from moving robot - Shearing at joints or between gripper jaws - Entanglement in moving cables or linkages

Electrical Hazards: - Contact with live parts (motors, drives) - Arc flash from high-power systems

Other Hazards: - Pneumatic system pressure release - Flying parts from gripper failures - Noise from pneumatic systems

### **Risk Evaluation**

Severity × Probability = Risk Level

Severity: - S1: Slight injury (bruise, minor cut) - S2: Serious injury (fracture, amputation) - S3: Fatal injury

Probability: - P1: Low (remote possibility) - P2: Medium (occasional) - P3: High (frequent or continuous exposure)

**Risk Reduction Hierarchy** 1. Eliminate hazard (inherently safe design) 2. Guard against hazard (physical barriers, interlocks) 3. Warn about hazard (lights, alarms, signs) 4. Train and supervise personnel (PPE, procedures)

## **Safeguarding Methods**

### **Physical Guards**

Fixed Guards: - Permanent panels or fencing - No moving parts - Requires tools to remove - Lowest cost, highest reliability - Examples: Wire mesh panels, polycarbonate enclosures

Interlocked Guards: - Opens for access - Safety switch (guard lock) - Robot stops when guard open - ISO 14119 compliant switches - Two-channel redundant monitoring

Access Gates: - Pedestrian or material access - Trapped key interlocks (access only when robot stopped) - Time delay (prevent re-entry after closing)

## Perimeter Guarding

Fence Specifications: - Height: Minimum 1800mm (6 ft) - Mesh opening: <50mm to prevent reach-through - Distance to hazard: Minimum 100mm (guard to robot when fully extended)

Gates and Openings: - Self-closing and interlocked - Trapped key systems for maintenance access - Lockout/tagout provisions

## Presence-Sensing Devices

Light Curtains: - Type 4 per IEC 61496 - Protected height: 300-1800mm typical - Resolution: 14mm, 30mm, or 50mm - Response time: <20ms - Safety category 4, PLe per ISO 13849

Safety Laser Scanners: - 2D or 3D area monitoring - Configurable warning and protective zones - Automatic zone switching (different areas for different modes) - Examples: SICK S3000, Keyence SZ-V

Safety Mats: - Pressure-sensitive floor mats - Detect personnel entry to zone - Lower reliability than optical sensors - Useful for large floor areas

## Safety Relays and Controllers

Safety PLC: - Redundant processors - Self-checking architecture - Examples: Pilz PNOZ, Allen-Bradley GuardLogix, Siemens S7-1200F

Dual-Channel Safety: - Two independent safety circuits - Discrepancy detection - Fail-safe design (power off = safe state)

## Emergency Stop (E-Stop)

Requirements: - Red mushroom button on yellow background - Diameter: 40mm minimum - Direct-opening action (mechanically linked contacts) - Latching (manual reset required) - Category 0 (immediate stop) or Category 1 (controlled stop)

Placement: - Accessible from all operator positions - At control pendant - On robot body if accessible - On external control panels

E-Stop Circuit:

[E-Stop 1] --□ □ -- [E-Stop 2] --□ □ -- [Safety Relay] --> [Motor Power Contactors]

Series connection: Any e-stop opens circuit.

## Safety Functions

**Safe Torque Off (STO)** - Removes power to motor drives - Robot coasts to stop (Category 0) - Does not control deceleration - Basic safety function (PLd or PLe)

**Safe Stop 1 (SS1)** - Controlled deceleration - Power removed after stop (Category 1) - Prevents uncontrolled motion - Requires safe motion monitoring

**Safe Stop 2 (SS2)** - Controlled stop, power maintained - Holds position - Used for temporary stops - Requires continuous position monitoring

**Safely-Limited Speed (SLS)** - Monitors robot speed - Stops if exceeds limit - Enables teaching/maintenance modes - Typical limit: 250 mm/s for hand-guiding

**Safe Reduced Speed (SRS)** - Reduces maximum speed in certain zones or modes - Enables closer operator proximity

**Safe Operating Stop (SOS)** - Monitors that robot is stationary - Power maintained - Allows safe operations near stopped robot

## Operating Modes

**Automatic Mode** - Full speed operation - All guards closed and interlocked - No personnel in hazard zone - Production mode

**Manual Mode (Teach/Setup)** - Reduced speed (<250 mm/s) - Three-position enabling switch (dead-man switch) - Operator inside guarded area - Requires mode selector key switch

**Maintenance Mode** - Lockout/tagout required - All energy sources isolated - Physical locks on power disconnects - Permits inside control panel

## Collaborative Operation

### ISO/TS 15066 Requirements

For robots working with humans without guarding:

Power and Force Limiting: - Contact force <150N (body) - Contact pressure depends on body region - Transient (<0.5s) vs. quasi-static contact - Built-in sensors or motor current monitoring

Speed and Separation Monitoring: - Minimum separation distance maintained - Robot slows or stops as operator approaches - Safety-rated position sensors - Requires 3D monitoring (laser scanners)

Hand Guiding: - Operator physically guides robot - Force sensor on end effector - Reduced speed - Three-position enabling switch

### Applications for Pick and Place

Limited collaborative operation: - Robot operates in automatic mode when separated - Slows or stops when operator enters shared zone - Typical for machine tending with occasional operator access

Not typically fully collaborative due to high speeds.

## Safety Circuit Design

**Category 3 Architecture** - Dual-channel monitoring - Single fault safe - Fault detection via cross-monitoring - Suitable for PLd

Example circuit:

[Guard 1 Contact A] -----+-----> [Safety Relay Input A]  
[Guard 2 Contact A] -----+

[Guard 1 Contact B] -----+-----> [Safety Relay Input B]  
[Guard 2 Contact B] -----+



Safety Relay monitors channel discrepancy.  
If  $A \neq B$ , fault detected, outputs disabled.

**Category 4 Architecture** - Similar to Category 3 - Additional redundancy and fault exclusion - Required for PLe applications

## **Warning Devices**

### **Visual Indicators**

Stack Lights: - Red: Fault/E-stop - Yellow: Warning/reduced speed - Green: Running automatic  
- Blue: Manual/teach mode (optional)

Strobes/Beacons: - Flash before robot motion starts - Warn of automatic operation

**Audible Alarms** - Horn or siren before automatic start - 3-second delay after guard close before motion - Distinctive sounds for different conditions

**Floor Markings** - Yellow striped tape marks hazard zone - Red zone: Do not enter during operation - ANSI/ISO safety colors

## **Validation and Testing**

### **Commissioning Checks**

E-Stop Function: - Press each e-stop button - Verify immediate robot stop - Verify power removal  
- Test reset function

Guard Interlock: - Open each guard - Verify robot stop - Attempt to start with guard open (should fail) - Close guard and verify restart

Light Curtain: - Break beam with test piece - Verify robot stop - Measure stopping time and distance - Verify muting functions (if applicable)

### **Periodic Testing**

Daily: - Visual inspection of guards - E-stop function test - Light curtain beam check

Monthly: - Full interlock function test - Safety relay diagnostics - Light curtain alignment

Annually: - Complete risk assessment review - Safety function response time measurement - Documentation update

## **Documentation**

### **Required Documentation**

Risk Assessment: - Hazard identification - Risk evaluation - Mitigation measures - Residual risk

Safety Validation: - Circuit diagrams - Component specifications (safety ratings) - Performance level calculation - Test results

Operating Procedures: - Startup/shutdown procedures - Lockout/tagout procedures - Emergency response - Authorized personnel list

Training Records: - Personnel trained - Training content - Dates and signatures - Competency verification

## **Common Violations and Mistakes**

**Inadequate Guarding** - Gaps in fence allowing reach-in - Guards easily removable without tools  
- Insufficient height

**Bypassing Safety Devices** - Taping guard switches closed - Jumper wires around safety circuits  
- Disabling light curtains

**Improper E-Stop Circuit** - Single-channel only - No mechanical latching - Insufficient number/placement

**Lack of Training** - Operators unaware of hazards - Maintenance without lockout/tagout - No emergency procedures

**Non-Safety-Rated Components** - Standard relays instead of safety relays - Non-rated sensors  
- Inadequate performance level

## **Best Practices**

- Design inherently safe first (limit force, speed, energy)
- Use safety-rated components throughout
- Redundant monitoring for critical functions
- Regular testing and maintenance
- Comprehensive operator training
- Clear documentation
- Risk assessment review after any modifications

---

**Next:** 9.9 Performance Optimization

---

## **References**

### **1. Safety Standards**

- ISO 10218-1:2011 - Robots - Safety requirements - Part 1: Robots
- ISO 10218-2:2011 - Robots - Safety requirements - Part 2: Robot systems and integration
- ANSI/RIA R15.06-2012 - Industrial Robots and Robot Systems Safety Requirements
- ISO 12100:2010 - Safety of machinery - General principles for design

### **2. Functional Safety**

- ISO 13849-1:2015 - Safety of machinery - Safety-related parts of control systems
- IEC 62061:2005 - Safety of machinery - Functional safety of safety-related control systems
- IEC 61508 - Functional safety of electrical/electronic/programmable systems

### **3. Safety Components**

- Pilz - Safe Automation Solutions Technical Documentation

- Schmersal - Safety Sensors and Systems Catalog
- SICK - Safety Technology for Machine Applications

#### 4. Risk Assessment

- ISO 14121-1:2007 - Safety of machinery - Risk assessment
  - Machinery Directive 2006/42/EC - EU Safety Requirements
- 

## Module 9 - Pick & Place Robot

Optimizing pick and place performance increases throughput, reduces cost per part, and improves return on investment.

### Cycle Time Analysis

#### Break Down the Cycle

Typical pick-and-place cycle: 1. Move to pick approach (20-30%) 2. Lower to pick position (5-10%) 3. Activate gripper (10-15%) 4. Raise from pick (5-10%) 5. Move to place approach (20-30%) 6. Lower to place position (5-10%) 7. Release gripper (5-10%) 8. Raise from place (5-10%)

Focus optimization on largest time consumers.

**Measurement Tools** - Stopwatch for manual timing - Controller logs (motion time stamps) - Sensor signals (oscilloscope or data logger) - Video analysis (slow-motion playback)

### Motion Optimization

#### Reduce Travel Distance

Minimize Z-Axis Travel: - Lower clearance heights (balance safety) - Pick/place from higher positions (less Z motion) - Before: Z+100mm clearance, After: Z+50mm (save 0.3s)

Optimize Layout: - Minimize distance between pick and place - Arrange multiple place locations in compact pattern - Consider robot reach envelope

#### Increase Velocity and Acceleration

Structural Improvements: - Stiffen frame (reduce deflection) - Reduce moving mass (lighter materials) - Better bearings and guides (lower friction)

Tuning: - Increase PID gains (careful: can cause instability) - Optimize acceleration limits (test structural limits) - Adjust S-curve jerk limiting (balance speed vs. smoothness)

Typical improvements: - Acceleration: 2 m/s<sup>2</sup> □ 5 m/s<sup>2</sup> (check structural limits) - Velocity: 1000 mm/s □ 2000 mm/s - Cycle time reduction: 20-30%

#### Motion Blending

Before (Sharp Corners):

```
G0 X100 Y50 Z50      ; Stop at endpoint
G0 X200 Y150         ; Stop at endpoint
```

After (Blended):

```
G0 X100 Y50 Z50      ; Blend through
G64 P5               ; Path tolerance 5mm
G0 X200 Y150
```

Robot rounds corners instead of stopping. Saves time but reduces positioning accuracy at intermediate points.

**Multi-Axis Coordination** - Simultaneous XYZ motion vs. sequential - Before: Move Z, then XY (safe but slow) - After: Move XYZ simultaneously (faster, requires collision checking)

## Gripper Optimization

### Faster Actuation

Vacuum Systems: - Higher flow venturi (faster grip and release) - Larger diameter tubing (lower restriction) - Shorter tube length (less volume to evacuate) - Typical improvement: 0.5s  $\rightarrow$  0.2s

Pneumatic Grippers: - Higher pressure (4 bar  $\rightarrow$  6 bar, check gripper rating) - Larger valve orifice - Speed controllers adjusted - Typical improvement: 0.3s  $\rightarrow$  0.15s

### Overlap Gripper with Motion

Before (Sequential):

```
robot.move(z=pick_height)
robot.wait_motion_complete()
robot.gripper_on()
time.sleep(0.5)
robot.move(z=clearance_height)
```

After (Overlapped):

```
robot.move(z=pick_height)
robot.wait_motion_complete()
robot.gripper_on()           # Start gripper
robot.move(z=clearance_height) # Move immediately
time.sleep(0.2)              # Brief dwell while moving
```

Gripper activates during upward motion. Verify part secure before horizontal motion.

## Vision Processing Optimization

### Reduce Image Acquisition Time

Camera Settings: - Lower resolution (if acceptable for task) - Region of interest (ROI) cropping - Faster frame rate camera - Example: 1280×1024  $\rightarrow$  640×480, saves 50ms

### Optimize Processing Algorithms

**Simplify:** - Reduce image preprocessing steps - Use faster algorithms (template matching  $\square$  blob detection) - Downsample image before processing

**Parallel Processing:** - GPU acceleration (CUDA, OpenCL) - Multi-threading for independent operations - Example: OpenCV with CUDA can be 5-10 $\times$  faster

### **Pipeline Vision with Motion**

Before (Sequential):

Wait for part  $\rightarrow$  Capture image  $\rightarrow$  Process  $\rightarrow$  Move to pick

After (Pipelined):

Capture image while robot finishing previous place

Process image while robot moving to pick approach

Pick offset already calculated when arrival

Saves entire vision processing time from critical path.

### **Controller and Software Optimization**

**Look-Ahead and Trajectory Planning** - Controller examines upcoming moves - Optimizes velocity profile through sequences - Prevents unnecessary deceleration/acceleration

**Reduce Communication Latency** - Direct I/O instead of networked (Modbus, etc.) - Higher baud rates for serial communication - Ethernet vs. slower protocols

**Optimize Control Loop** - Higher servo update rate (1 kHz  $\square$  2 kHz) - Faster computer/controller (if CPU limited) - Real-time OS for deterministic timing

### **Multi-Robot and Parallel Operations**

**Multiple Robots** - Two robots share workspace - Coordinate to avoid collisions - Double throughput (with overhead for coordination)

**Parallel Grippers** - Pick multiple parts simultaneously - Gripper with 2, 4, or more pickup points - Requires precise part spacing

**Multi-Station Buffering** - Queue parts between operations - Smooth variations in cycle time - Increase overall throughput

### **Part Presentation Optimization**

#### **Oriented Part Feed**

Vibratory Bowl Feeders: - Parts arrive in consistent orientation - Eliminates vision processing for orientation - Faster and more reliable picking

Tray/Magazine Systems: - Parts in fixed grid - Known positions (no vision needed) - Simple array programming

**Conveyor Optimization** - Match conveyor speed to robot cycle time - Avoid start/stop (continuous flow) - Part spacing allows robot access

## Energy Efficiency

### Reduce Power Consumption

Mechanical: - Counterbalance Z-axis (spring or pneumatic) - Reduce friction (better bearings, lubrication) - Lighter moving components

Electrical: - Regenerative braking on servo drives - Idle mode (reduced current when not moving) - Turn off vacuum when not gripping

Pneumatic: - Reduce air pressure to minimum required - Fix leaks (cost savings and reliability) - Use vacuum pumps instead of venturi (lower energy)

**Idle Modes** - Reduce motor current during dwell - Sleep mode when no parts available - Significant savings for non-continuous operation

## Overall Equipment Effectiveness (OEE)

### OEE Calculation

$OEE = Availability \times Performance \times Quality$

$Availability = Operating\ Time / Planned\ Production\ Time$

$Performance = Actual\ Output / Maximum\ Possible\ Output$

$Quality = Good\ Parts / Total\ Parts$

**Improve Availability** - Reduce downtime (maintenance, breakdowns) - Faster changeovers (quick-change tooling) - Minimize setup time

**Improve Performance** - Reduce cycle time (all above optimizations) - Eliminate minor stoppages (jams, sensor faults) - Consistent part supply

**Improve Quality** - Reduce pick failures (better grippers, vision) - Prevent dropped parts - Accurate placement (calibration, stiffness)

## Case Study: Optimization Example

**Initial System** - Cartesian robot, 600×400mm workspace - Cycle time: 3.5 seconds - Throughput: 1029 parts/hour

**Optimizations Applied** 1. Reduced clearance height: 100mm → 60mm (save 0.4s) 2. Increased velocity: 1000 mm/s → 1500 mm/s (save 0.3s) 3. Blended motion through intermediate points (save 0.2s) 4. Overlapped gripper actuation with motion (save 0.3s) 5. Pipelined vision processing (save 0.6s)

**Optimized System** - Cycle time: 1.7 seconds (51% reduction) - Throughput: 2118 parts/hour (106% increase)

**Cost/Benefit** - Optimization labor: 20 hours @ \$75/hr = \$1500 - Increased output: 1089 parts/hour - Value at \$0.50 profit/part and 2000 hours/year: \$1,089,000/year - ROI: Immediate (payback in <1 day of operation)

## Benchmarking and Testing

### Performance Metrics

Cycle Time: - Measure over 100+ cycles - Average, minimum, maximum - Standard deviation (consistency)

Throughput: - Parts per hour - Account for downtime and faults

Accuracy: - Measure placement variation - CMM or vision system - +/-X/Y/rotation

Reliability: - Mean time between failures (MTBF) - Success rate (picks/attempts) - Uptime percentage

### Testing Procedures

Endurance Testing: - Run continuous operation (8-24 hours) - Monitor for degradation or failures  
- Thermal effects on accuracy

Stress Testing: - Maximum speed and acceleration - Heaviest payload - Continuous operation - Find limits and margins

## Continuous Improvement

**Data Collection** - Log cycle times, faults, downtime - Identify trends and patterns - Focus on biggest opportunities

**Iterative Optimization** - Change one parameter at a time - Measure impact - Document results - Repeat

**Operator Feedback** - Operators identify inefficiencies - Suggest improvements - Involve in testing

---

**Next:** 9.10 Maintenance

---

## References

### 1. Performance Analysis

- Groover, M.P. (2014). *Automation, Production Systems, and Computer-Integrated Manufacturing*. Pearson
- Overall Equipment Effectiveness (OEE) - Industry Standard Metrics
- Lean Manufacturing Principles - Toyota Production System

### 2. Motion Optimization

- Lynch, K.M. & Park, F.C. (2017). *Modern Robotics*. Cambridge University Press
- Trajectory optimization algorithms - Academic research papers
- Time-optimal motion planning techniques

### 3. Vision System Performance

- Bradski, G. & Kaehler, A. (2008). *Learning OpenCV*. O'Reilly
- Real-time image processing optimization techniques
- GPU acceleration for vision processing - CUDA/OpenCL documentation

### 4. System Tuning

- PID tuning methods - Ziegler-Nichols and modern techniques
  - Motor and drive optimization - Manufacturer technical guides
-