Complete Engineering Guide: Designing and Building a Functional Prosthetic Arm

A Comprehensive Technical Documentation for Beginner and Student Engineers

This documentation provides an in-depth, technically accurate guide for understanding and building a prosthetic arm. It covers everything from fundamental concepts to practical implementation, without oversimplifying the challenges involved. This guide is structured for engineering students who want to learn by building, while being honest about what's achievable in an academic setting versus clinical requirements.

1. High-Level Understanding

What is a Prosthetic Arm in Engineering Terms?

A prosthetic arm is an electromechanical assistive device designed to replace or augment the function of a missing or impaired upper limb. From an engineering perspective, it consists of several integrated subsystems: a mechanical structure that replicates joint movements, an actuation system that provides motion, a control system that interprets user intent, a power system that energizes components, and a human-machine interface (the socket) that physically couples the device to the residual limb.

The fundamental challenge is creating a system that balances multiple competing constraints: weight versus strength, degrees of freedom versus control complexity, power consumption versus battery life, and cost versus functionality. Unlike lower limb prosthetics which primarily support locomotion, upper limb prosthetics must enable manipulation—a far more complex task requiring precision, adaptability, and sensory feedback.

Types of Prosthetic Arms

Prosthetic arms fall into three primary categories, each with distinct engineering characteristics:

Passive Prosthetics are non-actuated devices that provide cosmetic appearance, balance, and limited stabilization. They consist of rigid or semi-rigid structures that can be manually positioned. While they offer no active movement, they're lightweight (typically under 500g), require no power source, and have minimal maintenance requirements. Engineering-wise, these are the simplest—primarily concerned with material selection, weight distribution, and aesthetic finish. They're useful for understanding basic anthropomorphic design without the complexity of actuation.

Body-Powered Prosthetics use mechanical advantage through cable-and-harness systems to translate gross body movements (typically shoulder or upper arm motion) into terminal device action. The user wears a harness that connects to control cables; when they move their shoulder or expand their chest, the cable pulls and actuates a hook or hand. The mechanical advantage ratio is typically 2:1 to 3:1, meaning the user must move their shoulder 2-3 times the distance the terminal device moves.

These systems provide inherent force feedback—the user feels resistance through the cable tension, giving them proprioceptive awareness of grip force. Body-powered devices can generate grip forces of 50-90N depending on user strength and mechanical advantage. They're extremely durable, waterproof, and function-reliable in harsh environments. The engineering challenge lies in optimizing the cable routing, minimizing friction losses (which can consume 30-40% of input force), and designing comfortable harness attachment points that don't create pressure sores during extended use.

Myoelectric Prosthetics are externally powered devices controlled by electromyographic (EMG) signals from residual limb muscles. When a user contracts a muscle, it generates electrical activity (typically 50-500 µV at the skin surface) that can be detected by surface electrodes. These signals are amplified (gain of 1000-10,000x), filtered to remove noise, and processed by a microcontroller to control battery-powered motors.

The human arm has seven degrees of freedom (DOF): three at the shoulder (flexion/extension, abduction/adduction, internal/external rotation), one at the elbow (flexion/extension), two at the forearm (pronation/supination), and two at the wrist (flexion/extension, radial/ulnar deviation). High-end myoelectric prostheses can provide up to 6 controlled DOF: shoulder flexion, humeral rotation, elbow flexion, wrist rotation, wrist flexion, and hand open/close.

These devices can generate grip forces exceeding 100N with minimal user effort, but they're heavy (600-1200g for transradial), expensive ($20,000-$100,000), require battery charging (typically daily), and lack the inherent force feedback of body-powered systems. The sophistication comes at the cost of complexity: more failure modes, sensitivity to electromagnetic interference, and dependence on consistent electrode-skin contact.

Hybrid Prosthetics combine body-powered and myoelectric components. For example, a high-level amputee might use a body-powered elbow with a myoelectric hand, or vice versa. This approach leverages the strengths of each technology—body power for reliable, force-feedback control of one function, and myoelectric for another function that benefits from powered actuation.

Academic Prototypes vs Real-World Prosthetics: A Critical Distinction

This is perhaps the most important section for student engineers to internalize: there's a vast gulf between an academic prototype that demonstrates a concept and a clinically viable prosthetic device.

An academic prototype built on a $300 budget over 3-6 months typically achieves 2-4 DOF, uses hobby-grade servos and 3D-printed PLA components, implements basic switch or threshold-based EMG control, and demonstrates functionality for minutes to hours of intermittent use. Testing is limited to controlled laboratory conditions. This is valuable for learning and proof-of-concept validation.

A clinical prosthetic requires FDA Class II or CE marking approval (in most jurisdictions), costs $50,000-$500,000+ to develop, takes 2-5 years from concept to market, undergoes extensive ISO 22523 testing for mechanical strength and fatigue (>1.2 million cycles), includes custom socket fabrication by certified prosthetists, and must demonstrate 4+ years of daily use reliability. User testing involves actual amputees in longitudinal studies.

The key limitations of academic projects include: inability to create a properly fitted socket (the most critical component for comfort and function), lack of long-term reliability data, use of non-medical-grade materials and components, simplified control systems prone to noise and interference, and ethical constraints that prevent testing on actual users. Most academic prosthetic projects are educational demonstrations, not devices suitable for actual prosthetic use.

Understanding this distinction prevents overpromising capabilities and helps set realistic project goals. A well-executed student project that achieves reliable 3-DOF control and demonstrates basic ADL tasks is far more valuable than an overambitious 18-DOF design that fails during demonstration.

Core Engineering Goals

The fundamental objectives when designing any prosthetic arm are:

Control: The system must reliably translate user intent into device action with minimal latency (less than 200ms from signal to motion is generally acceptable for smooth operation). The control scheme should be intuitive—requiring minimal training and cognitive load. For myoelectric systems, this means achieving greater than 85% classification accuracy for intended movements under real-world conditions with noise and electrode shift.

Comfort: The socket interface is the most critical and most neglected component in student projects. Poor socket fit causes skin breakdown, pain, and device abandonment. Pressure distribution across the residual limb should ideally not exceed 205 kPa in any location, with lower pressures (less than 100 kPa) over bony prominences. Weight distribution must minimize distal loading, and materials must manage heat and perspiration.

Reliability: The device must function consistently across varying environmental conditions (temperature, humidity, electromagnetic interference) and through thousands of actuation cycles. In clinical settings, mean time between failures should exceed 6 months of daily use. For student projects, demonstrating 1000+ cycles without mechanical failure is a reasonable target.

Repeatability: The prosthetic must produce consistent output for the same input. For position control, this means ±5% error. For force control, ±2% of setpoint. Variability in performance erodes user trust and makes the device difficult to use effectively.

2. Problem Definition and User Needs

Defining User Requirements

Before designing any component, you must establish clear, measurable requirements. This process starts with understanding what the user needs to accomplish.

For prosthetic arms, requirements fall into several categories:

Functional Requirements define what tasks the device must enable. These are derived from Activities of Daily Living (ADL) assessments. Basic ADLs include: eating (using utensils, bringing food/drink to mouth), personal hygiene (washing face, brushing teeth), dressing (buttoning, zipping, tying), and toileting. Instrumental ADLs add complexity: meal preparation, managing medications, using communication devices, handling money and documents.

Research on upper limb amputees reveals the most difficult tasks: tying shoelaces (63.7% report difficulty), opening bottles with bottle openers (46.5%), using scissors (42.9%), and buttoning shirts (42.1%). These tasks require precision grip, fine motor control, and often bimanual coordination—capabilities that are extremely challenging to replicate in prosthetics.

Performance Requirements quantify how well the device must perform. For a basic functional prosthetic:

- Grip force: 13.4N minimum for stable grasp of everyday objects; 30-50N for heavier items
- Grip speed: Closing fingers from fully open to closed in 1-2 seconds
- Position accuracy: ±5-10% of commanded position
- Response latency: Less than 200ms from input signal to motion initiation
- Operating time: 4-8 hours between charges for typical use patterns

Physical Requirements constrain the design space:

- Weight: Ideally less than 600g for transradial prosthesis (hand + forearm)
- Size: Approximate hand dimensions—length 175-195mm, width 75-85mm, thickness 30-40mm
- Range of motion: Fingers should flex approximately 90° at PIP joints, 70° at DIP joints
- Degrees of freedom: Minimum 2-3 for basic function (hand open/close + wrist rotation); 4-6 for improved capability

Range of Motion Requirements

The human hand is extraordinarily complex—19 bones, 14 phalangeal joints in the fingers alone, plus the carpometacarpal joints, totaling approximately 20+ degrees of freedom when you count all articulations. Obviously, replicating this fully is impractical.

Understanding which movements matter most for ADLs guides simplification:

The **wrist** provides critical workspace expansion. Wrist flexion/extension (60° each direction) allows the hand to orient properly relative to surfaces at different heights—reaching down to pick something up versus reaching up to a shelf. Wrist radial/ulnar deviation (20°/30°) enables fine positioning adjustments. Wrist pronation/supination (80° each) is often the single most important function for ADLs because it allows the hand to rotate from palm-down to palm-up—essential for tasks like turning doorknobs, using utensils, and keyboard use.

The **fingers** in the human hand have tremendous flexibility, but most ADL grasps can be accomplished with simplified kinematics. Research on grasp taxonomies identifies key patterns:

- **Power grasps** (cylindrical, spherical, hook) use all fingers wrapping around an object with the thumb opposing. These require synchronized finger flexion but not necessarily independent finger control.
- **Precision grasps** (tripod, pinch, lateral) use the thumb opposing one or two fingers with fine force modulation. These require independent thumb control and at least one opposing finger.

For a student project, targeting 3-5 DOF is realistic: (1) wrist rotation, (2) wrist flexion/extension, (3) hand open/close (all fingers together), (4) thumb opposition/adduction, and optionally (5) independent index finger control for pointing gestures.

Everyday Tasks Analysis

Let's analyze specific tasks to understand requirements:

Eating with a spoon: Requires a tripod or lateral grip on the utensil handle (diameter 8-15mm, length 150mm). The wrist must rotate to bring the spoon to the mouth, requiring pronation/supination. Grip force needed is minimal (5-10N) but must be maintained consistently. Success rate with body-powered prosthetics: approximately 90%; with myoelectric: approximately 95%.

Opening a door: Requires cylindrical grip on the handle (diameter 25-40mm), rotation force 15-25N, and often requires the wrist to rotate while maintaining grip. Body-powered success: 85%; myoelectric: 90%. The challenge is simultaneous grip and rotation.

Writing/typing: Requires tripod grip on a pen (diameter 8-12mm) with 3-8N force, plus precise positioning control. This is challenging for prosthetics because it requires fine motor control and sensory feedback about pressure. Success rate with body-powered: 70%; myoelectric: 85%.

Tying shoelaces: Requires bimanual coordination, precision pinch grip with 2-5N force, and the ability to manipulate flexible materials. This is extremely difficult—success rate with body-powered: 20%; myoelectric: 35%. Most prosthetic users adapt by using slip-on shoes or elastic laces.

Prioritizing Design Trade-offs

Every design involves compromises. The three primary dimensions are:

Grip strength vs. speed: Higher grip force typically requires larger, slower motors or higher gear ratios that reduce speed. A finger might close in 0.5 seconds with light force or 2 seconds with maximum force.

Stability vs. adaptability: A rigid, well-constrained mechanism is highly repeatable but can't adapt to object shape variation. Underactuated mechanisms with compliant elements adapt to shapes but have more variable performance.

Aesthetics vs. function: A cosmetically realistic hand requires careful shaping and surface finish, adding weight and complexity. A purely functional design can be lighter and simpler but may affect user acceptance. Research shows that aesthetics matter significantly for social acceptance, particularly for younger users.

For a student project, prioritize: **function > reliability > weight > aesthetics**. Get it working first, make it work consistently, then optimize weight, and finally address appearance if time permits.

3. Anatomy and Biomechanics Breakdown

Human Arm Structure

The human arm consists of several rigid segments connected by joints:

Humerus (upper arm bone) connects to the shoulder via a ball-and-socket joint providing 3 DOF. The shoulder joint center is relatively fixed relative to the scapula.

Radius and ulna (forearm bones) connect to the humerus via the elbow joint. The elbow is actually two joints: the humeroulnar joint provides 1 DOF (flexion/extension, approximately 150° range with full extension at 0°), and the radioulnar joint provides pronation/supination. The elbow's axis of rotation is not perfectly perpendicular to the bones—it has a slight oblique angle and the axis shifts slightly during motion.

Carpals, metacarpals, and phalanges (wrist and hand bones) form an incredibly complex structure. The wrist alone has 8 small bones that collectively provide 2 primary DOF (flexion/extension, radial/ulnar deviation) plus some compound motions. Each finger has 3 phalanges (proximal, middle, distal) connected by joints, plus the metacarpal connection.

Joint degrees of freedom summary:

- Shoulder: 3 DOF (flexion/extension, abduction/adduction, rotation)
- Elbow: 1 DOF (flexion/extension)
- Forearm: 2 DOF (pronation/supination, though this involves both radioulnar joints)
- Wrist: 2 DOF (flexion/extension, radial/ulnar deviation)
- **Total arm: 7 DOF** from shoulder to wrist position/orientation

This is a redundant system—only 6 DOF are needed to position and orient the hand anywhere in 3D space, but having 7 provides flexibility in how you achieve a given hand position. This redundancy is why you can touch your nose with your elbow at different angles.

What Can Realistically Be Replicated

Full replication is impossible with current technology at reasonable cost and weight. Clinical prosthetics typically achieve 3-6 powered DOF maximum. Student projects should target 2-4 DOF.

Focus on the most functionally important movements:

1. **Wrist rotation (pronation/supination)**: This is often cited as the single most valuable DOF for ADLs because it allows the hand to orient properly for tasks. If you can only power one joint, make it this one combined with hand open/close.
2. **Hand open/close**: Essential for any grasp. Can be implemented as all-fingers-together actuation, which dramatically simplifies control.
3. **Wrist flexion/extension**: Provides significant workspace expansion, allowing the hand to reach surfaces at different heights.
4. **Thumb opposition**: Moving the thumb from alongside the fingers to opposing them enables both power and precision grasps.

Higher-level joints (shoulder, elbow) are usually addressed through body-powered mechanisms or passive positioning for student projects because they require very high torque (20-35+ Nm) and consume significant battery power.

What Can Be Simplified

Underactuated mechanisms allow multiple DOF to be driven by a single actuator using clever mechanical coupling. For example, a whiffletree differential or pulley system can drive four fingers with one motor, with the fingers automatically adapting to object shape through passive compliance.

When a finger contacts an object surface, it stops moving while other fingers continue to close, automatically conforming to irregular shapes. This provides "mechanical intelligence"—shape adaptation without complex sensors or control algorithms.

The trade-off is loss of independent finger control. You cannot, for example, point with the index finger while keeping others closed. But for most power grasps (cylindrical, spherical), this limitation is acceptable.

Finger joint coupling can be further simplified. The human finger has 3 joints (MCP, PIP, DIP) with complex coupled motion patterns. For prosthetics, common simplifications include:

- **Two-joint fingers**: Eliminate the DIP joint or mechanically couple it to PIP motion
- **Single-tendon actuation**: One cable/tendon flexes all joints in a finger, with return motion via elastic elements
- **Four-bar linkage mechanisms**: Use mechanical linkages to create coordinated joint motion from a single input

These simplifications reduce the number of actuators, control signals, and mechanical complexity while maintaining reasonable grasp functionality.

4. Mechanical Design

Joint Design and Selection

Each joint in the prosthetic arm must be designed considering: range of motion, load-bearing capacity, friction/efficiency, size and weight, and manufacturing feasibility.

Shoulder joint: For student projects, this is typically a passive or manually positioned joint. If powered, it requires a high-torque servo (30-50 kg·cm) due to the long moment arm of the entire arm assembly. A simple revolute joint providing flexion/extension is most practical. Ball-and-socket joints providing 3 DOF are complex to manufacture and control.

Elbow joint: Similar to shoulder—high torque requirement (20-35 kg·cm for a forearm + hand assembly weighing 500-600g at 15cm moment arm). Can be implemented as:

- Powered revolute joint with a geared servo motor
- Body-powered using cable actuation
- Passive with friction or ratchet locking at discrete angles

The joint must withstand side loads during use. A simple pin-and-bushing design is adequate for low-speed operation, but thrust bearings improve smoothness.

Wrist joints: These require less torque than arm joints because they're moving just the hand assembly (200-300g).

For **wrist rotation (pronation/supination)**: A servo motor can directly drive the hand relative to the forearm. Torque requirement with no load is approximately 0.3-0.4 Nm; with a 300g grasped object, increases to 0.6-0.9 Nm. With a 1.5 safety factor, specify a motor providing at least 0.9 Nm (9.2 kg·cm). An MG996R servo (11 kg·cm) is adequate.

For **wrist flexion/extension**: Similar torque requirements. Can be implemented with a servo motor or, in body-powered designs, with a friction joint that the user positions manually.

Finger joints: Each finger typically has 2-3 joints (metacarpophalangeal MCP, proximal interphalangeal PIP, distal interphalangeal DIP).

For underactuated designs, all joints in a finger can be driven by a single tendon that creates a moment at each joint as it's pulled. The tendon routes through small pulleys at each joint axis. When tension is applied, the finger curls. Elastic bands or springs on the opposite side provide extension force.

For **independently actuated fingers**, each finger needs its own servo motor. Torque requirement for a single finger gripping with 13.4N force at the fingertip (3.75cm moment arm): $\tau = 13.4\text{N} \times 0.0375\text{m} = 0.50$ Nm. With a 1.4 safety factor: 0.70 Nm (7.1 kg·cm). An MG995 servo (11 kg·cm) provides adequate margin.

However, this calculation assumes the motor directly drives the finger joint. If using a tendon transmission system, you must account for:

- **Friction losses**: Tendons sliding through guides can lose 20-40% of input force to friction
- **Mechanical advantage**: The ratio of motor pulley radius to joint moment arm affects force transmission
- **Return force**: Elastic bands providing finger extension consume force that must be overcome

A more conservative specification would be 10-12 kg·cm per finger motor to ensure adequate force margin.

Actuator Selection

Servo motors are the most common choice for student projects:

Advantages: Integrated position feedback, built-in control electronics, easy to interface with microcontrollers (PWM signal), widely available, relatively inexpensive ($3-15 depending on quality).

Disadvantages: Limited rotation range (typically 180-270°, though continuous rotation servos exist), lower efficiency than brushed DC motors with external gearboxes, limited torque-to-weight ratio compared to more expensive options.

Common models for prosthetics:

- **SG90 micro servo**: 1.8 kg·cm, 9g weight, $2-3. Suitable for very small fingers or lightweight prototypes, but often underpowered.
- **MG995/MG996R**: 11 kg·cm, 55g, $8-12. Good all-around choice for fingers and wrists.
- **High-torque digital servos**: 20-35 kg·cm, 60-80g, $15-30. For wrist and elbow joints.

Brushed DC motors with gearboxes provide better efficiency and torque-to-weight but require external motor drivers and position sensors (encoders or potentiometers).

Linear actuators can be used for tendon-driven systems, directly pulling tendons to flex fingers. They provide linear force (measured in Newtons) rather than rotational torque. A typical small linear actuator provides 10-50N force with 20-50mm stroke at $15-30.

Torque Calculations: A Practical Guide

Torque (τ) is the rotational equivalent of force, measured in Newton-meters (Nm) or kilogram-force-centimeters (kg·cm).

Basic torque equation: $\tau = r \times F \times \sin(\theta)$

Where:

- r = distance from rotation axis to where force is applied (moment arm) [meters]
- F = force applied [Newtons]
- θ = angle between force vector and radius vector [degrees]

When force is perpendicular to the moment arm ($\theta = 90°$, most common case): $\boldsymbol{\tau = r \times F}$

Example 1: Single finger grip

A finger needs to apply 13.4N grip force at the fingertip, which is 3.75cm from the MCP joint axis.

$\tau = 0.0375\text{m} \times 13.4\text{N} = 0.50\ \text{Nm} = 5.1\ \text{kg·cm}$

This is the output torque required at the joint. If a servo motor directly drives the joint, you need at least 5.1 kg·cm. Adding a 1.4 safety factor: 7.1 kg·cm.

Example 2: Wrist rotation with grasped object

The hand assembly weighs 300g and its center of mass is 10cm from the wrist rotation axis. You're holding a 300g object with center of mass also at 10cm.

Total weight force: $(0.3\text{kg} + 0.3\text{kg}) \times 9.81\ \text{m/s}^2 = 5.88\text{N}$

When the wrist is horizontal (worst case, arm parallel to ground), this creates a moment about the wrist rotation axis:

$\tau = 0.10\text{m} \times 5.88\text{N} = 0.59\ \text{Nm} = 6.0\ \text{kg·cm}$

With 1.5 safety factor: 9.0 kg·cm.

Note: This is just the static torque to hold the position. If you want to rotate quickly, you must also account for angular acceleration torque.

Example 3: Full hand cylindrical grip

All five fingers gripping a cylindrical object, each contributing 13.4N:

Total grip force = 5 × 13.4N = 67N

If using a single motor with a whiffletree differential, the motor must provide enough torque to drive all fingers simultaneously:

Total required torque ≈ 5 × 0.50 Nm = 2.5 Nm = 25.5 kg·cm

With safety factor: 35.7 kg·cm

This exceeds any standard hobby servo, so you would need either:

- Multiple motors (one per finger or per finger pair)
- A high-torque industrial servo
- Gear reduction on a lower-torque motor

Common mistakes:

- Forgetting to convert units (cm to m, kg to N)
- Ignoring friction losses in transmissions (can be 20-40%)
- Not accounting for worst-case loading angles
- Undersizing motors by using nominal rather than peak torque specs

Material Selection for 3D Printing

PLA (Polylactic Acid):

- **Tensile strength**: 50-60 MPa
- **Density**: 1.24-1.30 g/cm³
- **Print temp**: 190-220°C
- **Pros**: Easy to print, good dimensional accuracy, low cost ($15-25/kg), biocompatible
- **Cons**: Brittle under repeated stress, poor temperature resistance (deforms >50°C), low impact resistance, degrades with moisture over time
- **Best for**: Structural components that don't flex, cosmetic parts, initial prototypes

PETG (Polyethylene Terephthalate Glycol):

- **Tensile strength**: 50-55 MPa
- **Density**: 1.27 g/cm³
- **Print temp**: 220-250°C
- **Pros**: Higher impact resistance than PLA, good layer adhesion, temperature resistant to 70-80°C, moisture resistant
- **Cons**: More stringing during prints, slightly harder to print than PLA
- **Best for**: Load-bearing parts like finger segments, palm structure, components subject to repeated stress

ABS (Acrylonitrile Butadiene Styrene):

- **Tensile strength**: 40-45 MPa
- **Density**: 1.04 g/cm³

- **Print temp**: 220-250°C
- **Pros**: Good impact resistance, temperature resistant to 90-100°C, can be post-processed with acetone vapor
- **Cons**: Warping during printing, requires heated bed and enclosure, produces unpleasant fumes, not biocompatible
- **Best for**: Functional prototypes where post-processing is needed, not recommended for skin contact

TPU (Thermoplastic Polyurethane):

- **Tensile strength**: 26-52 MPa
- **Elongation at break**: 400-700% (highly flexible)
- **Print temp**: 210-230°C
- **Pros**: Extremely flexible, high impact absorption, good chemical resistance
- **Cons**: Difficult to print (slow speeds required), expensive ($25-40/kg)
- **Best for**: Flexible joints, socket liner materials, cosmetic skin layers

Nylon:

- **Tensile strength**: 45-85 MPa
- **Density**: 1.14 g/cm³
- **Print temp**: 240-270°C
- **Pros**: Very high strength and durability, excellent wear resistance, low friction coefficient
- **Cons**: Hygroscopic (absorbs moisture from air, affecting print quality), difficult to print, expensive ($30-50/kg)
- **Best for**: High-stress components like gear teeth, pivot pins, load-bearing structures

Design considerations for 3D printing:

Layer orientation matters critically. Parts printed with layers perpendicular to the primary stress direction are 30-50% weaker than those printed with layers parallel to stress. This is because delamination between layers is a common failure mode.

For a finger segment that experiences bending stress, orient the print so layers run along the length of the finger (longitudinal orientation), not across it.

Infill density affects strength and weight. Common values:

- 20-30%: Cosmetic parts, low-stress components
- 50-70%: Load-bearing structures
- 100%: Critical high-stress components like joint pivots

Higher infill dramatically increases print time and weight, so use judiciously.

Wall thickness (perimeters/shells) often contributes more to strength than infill. Use 3-5 perimeter walls for strength-critical parts.

Weight Distribution and Structural Integrity

Weight is critical in prosthetics. Every 100g of prosthetic weight at the end of the limb feels like significantly more due to the moment arm. A 600g transradial prosthetic can feel like 2-3kg when held at arm's length.

Design strategies to minimize weight:

- Use topology optimization or lattice structures for non-visible internal structures
- Reduce infill density where possible (20-30% often sufficient)
- Select materials by strength-to-weight ratio
- Consolidate parts to eliminate unnecessary fasteners

- Use thin-walled structures with ribbing for stiffness rather than solid sections

Critical stress points:

- Finger joint connections (where phalanges connect to pins)
- MCP joint where fingers connect to palm
- Wrist joint attachment points
- Tendon routing channels (high wear from friction)

Use FEA (finite element analysis) simulation to identify stress concentrations, even in student projects. SolidWorks, Fusion 360, and FreeCAD all have FEA capabilities. For a static load test, mesh the part (aim for 50,000+ elements), apply boundary conditions (fix one end, apply force to the other), and look for regions where von Mises stress exceeds material yield strength (typically 30-40 MPa for PLA/PETG).

Common failure modes:

- **Delamination**: Layers separating under tension perpendicular to print direction
- **Creep**: Plastic deformation over time under constant load (especially PLA)
- **Fatigue**: Crack propagation from repeated loading cycles
- **Wear**: Tendon channels and joint surfaces experience abrasion

Testing should include cyclic loading. ISO 22523 specifies 1.2 million cycles for clinical devices, but even achieving 24,000 cycles (equivalent to 4 years at typical usage rates) in student projects demonstrates reasonable durability.

5. Control Systems

Control Methods Overview

Switch-based control (the simplest):

The user activates a mechanical switch (button, toggle, or pressure sensor) to trigger predefined actions. For example:

- Button 1: Close hand
- Button 2: Open hand
- Button 3: Rotate wrist

Advantages: Extremely reliable, no signal processing required, minimal code complexity, very low latency (less than 50ms).

Disadvantages: Requires the user's other hand or body part to activate switches, not suitable for bilateral amputees, limited to discrete positions (can't achieve proportional control easily).

Implementation: Connect switches to digital input pins on Arduino. Use internal pull-up resistors. Debounce in software (ignore state changes faster than 50ms to prevent switch bounce false triggers).

Joystick-based control:

An analog joystick provides two axes (X, Y), each outputting 0-5V (or 0-3.3V) based on position. Map these values to joint positions or velocities.

Advantages: Proportional control (speed/position varies with joystick deflection), intuitive for users familiar with game controllers, provides 2 DOF control from single input device.

Disadvantages: Still requires an intact hand to operate, can't achieve true simultaneous multi-joint control without multiple joysticks.

Implementation: Read analog input values, map to servo position or motor speed.

Myoelectric control (EMG-based):

Surface electrodes detect electrical activity from muscle contractions (10-500 μV amplitude, 20-500 Hz frequency). These signals are amplified, filtered, and processed to determine user intent.

Basic threshold control (simplest myoelectric approach):

Monitor the EMG signal amplitude from one or two muscles. When amplitude exceeds a threshold, trigger an action.

Example: One electrode on wrist flexor muscles, one on extensor muscles.

- Flexor activation > threshold → close hand
- Extensor activation > threshold → open hand

Advantages: Relatively simple to implement, requires only 2 EMG channels, intuitive mapping (contract muscle → corresponding action).

Disadvantages: Sensitive to signal noise and electrode shift, requires careful threshold calibration for each user, prone to false triggers from muscle fatigue or unintended contractions, limited to sequential control of DOF.

Proportional myoelectric control:

The magnitude of EMG signal determines speed or force of action. Higher muscle contraction = faster/stronger actuation.

Advantages: More natural control, user can modulate force/speed.

Disadvantages: Requires clean EMG signal with good SNR, more complex signal processing, user must learn to control contraction strength precisely.

Pattern recognition-based control:

Use machine learning to classify muscle activation patterns into different intended movements. For example, train a classifier to distinguish:

- Hand open
- Hand close
- Wrist rotate right
- Wrist rotate left
- Pinch grip
- Power grip

Advantages: Can control multiple DOF, potentially more intuitive after training, robust to individual muscle variation.

Disadvantages: Requires training data collection, computationally intensive (may need more powerful microcontroller than Arduino Uno), susceptible to performance degradation with electrode shift or changing conditions, requires user training period.

Implementation complexity: For student projects, pattern recognition is advanced but achievable. Simpler approaches (threshold or proportional control) are recommended for first-time builds.

Microcontroller Selection

Arduino Uno/Nano:

- **Processor**: ATmega328P, 16 MHz, 8-bit
- **Memory**: 32 KB flash, 2 KB RAM
- **Analog inputs**: 6 channels, 10-bit resolution (0-1023)
- **Digital I/O**: 14 pins

- **PWM outputs**: 6 pins (for servo control)
- **Cost**: $3-25 depending on original vs. clone
- **Pros**: Huge community support, extensive libraries, easy programming (Arduino IDE), 5V logic compatible with most hobby servos and sensors
- **Cons**: Limited RAM for complex signal processing, relatively slow for pattern recognition, single-core limits real-time performance

Best for: Basic prosthetic projects with switch, joystick, or simple threshold EMG control, driving 5-8 servos, implementing basic filtering and control logic.

ESP32:

- **Processor**: Dual-core Tensilica Xtensa LX6, 240 MHz, 32-bit
- **Memory**: 4 MB flash, 520 KB RAM
- **Analog inputs**: 18 channels, 12-bit resolution (0-4095)
- **Digital I/O**: 34 pins
- **Wireless**: Built-in WiFi and Bluetooth
- **Cost**: $5-15
- **Pros**: Much more powerful than Arduino, sufficient for pattern recognition, wireless capabilities enable remote monitoring/control, multiple cores allow real-time processing
- **Cons**: 3.3V logic (requires level shifting for 5V sensors/servos), more complex programming, higher power consumption (240mA typical vs. 40mA for Arduino)

Best for: Advanced projects with pattern recognition, wireless control, data logging to smartphone, simultaneous multi-channel EMG processing.

STM32 series (e.g., STM32F103):

- **Processor**: ARM Cortex-M3, 72 MHz, 32-bit
- **Memory**: 64-128 KB flash, 20 KB RAM
- **Cost**: $3-10
- **Pros**: Excellent price-to-performance, low power consumption, many peripherals
- **Cons**: Steeper learning curve, less community support than Arduino

Best for: Production-oriented projects where cost and power efficiency matter, students comfortable with embedded C programming.

PID Control Simplified

PID (Proportional-Integral-Derivative) control is a feedback loop that continuously corrects errors between desired and actual states.

Why do you need it? Servo motors have built-in position control, but for tasks requiring precise force control, velocity control, or controlling motors without built-in feedback, PID is essential.

The three components:

Proportional (P): Correction proportional to current error.

If your desired position is 90° and current position is 70°, error = 20°. Proportional output = K_p × error = K_p × 20°. Larger error → larger correction. This provides fast initial response but can overshoot.

Integral (I): Correction based on accumulated past error.

If there's a small steady-state error (e.g., always 2° off target), the integral term accumulates this error over time and increases correction to eliminate it. Integral output = $K\_i \times \int(\text{error})dt \approx K\_i \times \text{sum\_of\_errors\_over\_time}$. This eliminates steady-state offset but can cause overshoot if tuned too aggressively.

Derivative (D): Correction based on rate of change of error.

If error is decreasing rapidly (you're approaching target fast), derivative term applies braking to prevent overshoot. Derivative output = $K\_d \times d(\text{error})/dt \approx K\_d \times (\text{current\_error} - \text{previous\_error}) / \text{time\_interval}$. This dampens oscillations and improves stability.

Total control output = $K\_p \times \text{error} + K\_i \times \int(\text{error})dt + K\_d \times d(\text{error})/dt$

Tuning PID gains (Kp, Ki, Kd) is an art. Simple tuning procedure:

1. Set Ki = 0, Kd = 0. Increase Kp until system oscillates steadily.
2. Reduce Kp to 60% of oscillation value.
3. Increase Ki slowly until steady-state error is eliminated (but watch for overshoot).
4. Increase Kd to reduce overshoot and oscillations.

For a finger position control with servo motor, typical values might be Kp=2.0, Ki=0.5, Kd=0.1, but these are highly system-dependent.

When do you need PID in prosthetics?

- **Force control**: Gripping with specific force requires feedback from force sensors and PID to maintain setpoint force
- **Velocity control**: Closing fingers at constant speed regardless of load
- **Disturbance rejection**: Maintaining position despite external forces

For basic position control with servos, the servo's internal controller is usually sufficient and external PID isn't needed.

Latency, Precision, and Power Considerations

Latency (delay from input to output) must be minimized for natural control.

Human perception: delays less than 100ms are generally imperceptible; 100-200ms is acceptable; greater than 300ms feels sluggish and disrupts motor coordination.

Sources of latency:

- EMG signal acquisition and filtering: 20-50ms
- Pattern recognition computation: 50-150ms (depends on algorithm)
- Motor response time: 50-200ms (depends on motor and load)

Total system latency target: Less than 200ms

Precision in prosthetic control means repeatability and accuracy.

Position accuracy: For servo-controlled joints, ±5% of full range is acceptable (e.g., ±9° for a 180° servo).

Force accuracy: For force-controlled grasping, ±2% of setpoint force is good. This requires force sensors with adequate resolution and signal processing to reduce noise.

Power consumption directly impacts battery life.

System power calculation example:

- 5 finger servos, average 300mA each at 6V: 9W
- 3 joint servos (wrist, elbow, shoulder), average 500mA at 6V: 9W

- Arduino Uno: 40mA at 5V: 0.2W
- EMG sensors (3x): 60mA total at 5V: 0.3W
- **Total average power**: approximately 18.5W

Peak power during simultaneous actuation can be 2-3× higher.

Battery selection:

For a 7.4V system drawing average 2.5A:

- **2600mAh Li-Po battery**: Theoretical runtime = 2600mAh / 2500mA = 1.04 hours of continuous use
- In practice, factor 0.8 efficiency: approximately 50 minutes continuous

For typical ADL usage (1-3 hours actual hand movement per day with idle periods), a 2600mAh battery provides 6-8 hours between charges.

Commercial prosthetics typically use 800-2600mAh Li-Po batteries at 7.4V, providing 1 day of typical use per charge.

Power optimization strategies:

- Sleep modes when idle (can reduce power by 90%)
- Reduce servo update rate (update position at 50Hz instead of 300Hz)
- Use high-efficiency motors (brushless DC instead of brushed)
- Implement current limiting to prevent battery damage and reduce peak draw

6. Electronics and Wiring

Power System Design

Voltage requirements:

Most hobby servos operate at 4.8-7.4V. Using a **7.4V (2S) Li-Po battery** is common because it's within servo spec and provides good torque.

Microcontrollers typically need 5V (Arduino) or 3.3V (ESP32). Use a **voltage regulator** to step down battery voltage to microcontroller voltage:

- **Linear regulators** (e.g., LM7805 for 5V): Simple, cheap ($0.50), but inefficient (40-60% efficiency), waste energy as heat
- **Buck converters** (switching regulators): More complex ($2-5), but 85-95% efficient, critical for battery-powered devices

Never power servos directly from microcontroller pins—they draw far too much current (500-2500mA vs. maximum 40mA per Arduino pin). Servos must be powered from the battery through a dedicated power rail.

Battery protection:

Li-Po batteries are energy-dense but dangerous if mishandled:

- **Over-discharge**: Damages battery permanently. Don't discharge below 3.0V per cell (6.0V for 2S battery). Use a low-voltage cutoff circuit or monitor voltage in code.
- **Over-current**: Can cause fire. Use a fuse (5-10A rating) in series with battery.
- **Over-charge**: During charging, don't exceed 4.2V per cell. Use a dedicated Li-Po balance charger.
- **Physical damage**: Punctured Li-Po batteries can ignite. Use protective case, avoid crushing.

Recommended battery specs for student project:

- Type: Lithium Polymer (Li-Po) 2S (7.4V nominal, 8.4V fully charged)
- Capacity: 2000-2600mAh
- Discharge rate: 20C or higher (allows 20 × 2.6A = 52A burst current, far exceeding needs)
- Protection: Built-in BMS (Battery Management System) if available
- Cost: $15-30

Wiring Best Practices

Common ground: All components (battery, microcontroller, servos, sensors) must share a common ground connection. This provides a voltage reference for signals.

Star topology: Connect all grounds to a single point (typically the microcontroller ground) rather than daisy-chaining. This reduces noise and ground loops.

Power distribution:

- Battery (+) → Fuse → Power switch → Servos (+)
- Battery (+) → Voltage regulator → Microcontroller (+)
- Battery (-) → Common ground

Wire gauge selection:

Current draw determines required wire thickness (AWG - American Wire Gauge):

- Microcontroller power: 0.5A → 24-26 AWG
- Individual servos: 2A → 20-22 AWG
- Main battery lines: 10A → 16-18 AWG

Using wire that's too thin causes voltage drop and overheating.

Strain relief: Where wires connect to moving parts (e.g., wrist joint), use flexible wire and secure with zip ties or hot glue to prevent fatigue fractures from repeated bending.

Connectors: Use proper connectors (JST, XT60 for battery) rather than soldering wires directly. This enables disassembly for maintenance.

Basic wiring diagram:

The system consists of:

- Battery connected to fuse and switch
- Output from switch feeding servo power rail and voltage regulator
- Voltage regulator output supplying microcontroller
- Microcontroller connected to EMG sensors through analog inputs
- Servo signal lines (PWM) from microcontroller to servo signal pins
- All grounds connected at common point

Signal Noise and Filtering Basics

Noise sources in prosthetics:

1. **Powerline interference (PLI)**: 50/60 Hz from AC mains, plus harmonics (100/120 Hz, 150/180 Hz, etc.)
2. **Motion artifacts**: Low-frequency (less than 20 Hz) noise from cable movement, electrode shift
3. **White Gaussian noise**: High-frequency (greater than 500 Hz) random noise from electronics

4. **EMG cross-talk**: Signals from nearby muscles contaminating target muscle signal
5. **Electromagnetic interference (EMI)**: From WiFi, Bluetooth, servo motors switching

Filtering strategy:

Hardware filtering (at sensor):

- Use shielded cables for EMG electrodes
- Place amplifier close to electrodes to boost signal before noise couples in
- Use differential amplifiers to reject common-mode noise (noise that appears equally on both electrode wires)

Digital filtering (in microcontroller):

High-pass filter (cutoff 20-30 Hz): Removes motion artifacts and DC offset.

Low-pass filter (cutoff 400-500 Hz): Removes high-frequency noise while preserving EMG signal (which is typically 20-500 Hz).

Notch filter (50/60 Hz): Specifically removes powerline interference. Can be implemented as a comb filter that creates notches at 50, 100, 150 Hz, etc., removing the fundamental and all harmonics.

Implementation: Arduino has limited computational power for complex filters, but simple filters are achievable using moving average (sums last N readings and divides by N) or exponential moving average (weighted average giving more weight to recent readings).

For more sophisticated filtering, libraries and pre-built filter chips exist, or use ESP32 with floating-point support for IIR/FIR filters.

Grounding and shielding:

For EMG sensors, proper grounding is critical:

- Use a third electrode as reference/ground, placed on bony prominence with minimal muscle activity
- Keep signal wires short (less than 30cm if possible)
- Route signal wires away from power wires to reduce coupling
- Use twisted-pair wiring for differential signals

SNR (Signal-to-Noise Ratio) is a key metric. For reliable EMG control, aim for SNR greater than 10 (signal amplitude 10× larger than noise amplitude). If SNR is poor:

- Check electrode placement (should be over muscle belly, not tendon)
- Check electrode-skin contact (clean skin, use conductive gel if needed)
- Adjust filter parameters
- Reduce EMI sources (move away from computers, switch to battery power instead of AC adapter)

7. Sensors for Prosthetic Control

EMG Sensors

Electromyography (EMG) measures electrical activity from muscle contractions. When a motor neuron signals a muscle to contract, it generates an action potential that propagates along muscle fibers. This creates an electrical field detectable at the skin surface, typically 10-500 μV amplitude in the 20-500 Hz frequency range.

Surface EMG electrodes are placed on the skin over the target muscle. Common types:

- **Wet electrodes**: Disposable adhesive electrodes with conductive gel. Best signal quality but single-use. Cost: $0.50-2 each.
- **Dry electrodes**: Reusable metal contacts. Slightly lower signal quality but more practical for prosthetics. Common materials: silver/silver-chloride, stainless steel.

Electrode placement is critical:

For residual limb EMG:

- Identify target muscles via palpation (feel muscle contracting while user attempts motion)
- Place electrode over muscle belly (thickest part), not over tendon
- Avoid bony prominences and areas with minimal subcutaneous tissue
- Use ultrasound guidance for precise placement if available (improves signal quality by 15-30%)
- Spacing: Electrodes should be 2-4 cm apart for differential measurement

Common placements for transradial amputees:

- Flexor digitorum superficialis: Anterior forearm, controls finger flexion → use for hand close
- Extensor digitorum: Posterior forearm, controls finger extension → use for hand open
- Flexor carpi radialis: Anterior medial forearm, wrist flexion
- Pronator teres: Anterior proximal forearm, forearm pronation

EMG sensor modules:

For student projects, pre-assembled EMG sensor modules are available that include:

- Amplification (gain 1000-10,000×)
- Filtering (band-pass 20-500 Hz, notch at 50/60 Hz)
- Output voltage: 0-5V proportional to EMG amplitude

Popular modules:

- **MyoWare Muscle Sensor**: $40, easy Arduino interface, good documentation
- **Muscle BioAmp Candy**: $25, open-source, compact
- **Generic EMG amplifier modules**: $5-15 (quality varies, often require additional filtering)

DIY EMG circuit (for advanced students):

Basic stages:

1. **Instrumentation amplifier** (e.g., AD620): Amplify differential signal, reject common-mode noise, gain approximately 100-1000
2. **High-pass filter**: Remove DC offset and motion artifacts, cutoff approximately 20 Hz
3. **Notch filter**: Remove 50/60 Hz powerline interference
4. **Gain stage**: Amplify to 0-5V range for Arduino ADC
5. **Envelope detector**: Rectify and low-pass filter to extract signal amplitude

This requires understanding of op-amp circuits and is error-prone for beginners. Pre-made modules are strongly recommended for first projects.

Force Sensors

Why force sensing? Enables closed-loop force control for tasks requiring specific grip strength—e.g., holding a fragile egg vs. gripping a tool.

Force-Sensitive Resistors (FSR):

FSRs are polymer films whose resistance decreases when pressure is applied.

Characteristics:

- Resistance range: Greater than 10 MΩ (no force) to less than 1 kΩ (maximum force)
- Force range: Typically 0.1-10 kg
- Response time: Less than 5 ms
- Output: Non-linear (resistance proportional to 1/Force^n, where n ≈ 0.7-0.9)
- Cost: $5-15

Limitations for prosthetics:

- Highly non-linear response makes accurate force measurement difficult
- Hysteresis (different reading depending on whether force is increasing or decreasing)
- Drift over time (calibration degrades)
- Sensitive to temperature

Best use: Binary contact detection (object touching vs. not touching) rather than precise force measurement.

Load cells (strain gauge-based):

Load cells use strain gauges bonded to a metal element. When force bends the element, resistance changes proportionally.

Characteristics:

- Linear response (resistance directly proportional to force)
- High accuracy (±0.05-0.5% full scale)
- Force range: 0-100g up to 0-50kg (choose appropriate range)
- Requires signal conditioning (Wheatstone bridge, instrumentation amplifier, often via HX711 module)
- Cost: $3-20 depending on capacity

Implementation:

Small load cells can be embedded in fingertips or in the actuator housing (measuring tendon tension as a proxy for grip force). The HX711 amplifier module provides a digital interface to Arduino, simplifying integration.

Other force sensors:

- **Piezoelectric sensors**: Generate voltage when compressed, good for dynamic forces but not static
- **Capacitive sensors**: Measure force via changes in capacitance, can be flexible
- **Optical sensors**: Measure deformation of compliant material via light intensity changes

Bend Sensors (Flex Sensors)

Flex sensors are variable resistors that change resistance when bent.

Characteristics:

- Resistance: Typically 10-50 kΩ straight, 50-100 kΩ bent 90°
- Bend range: 0-90° or 0-180° depending on type
- Response time: approximately 5 ms
- Durability: 100,000+ bend cycles
- Cost: $8-15 per sensor

Use in prosthetics:

Mounted on a glove worn on the intact hand, flex sensors detect finger positions. As the user bends their fingers, the prosthetic hand mirrors those movements.

Advantages:

- Intuitive control (natural finger movement mapping)
- No muscle signal processing required
- Good for users with bilateral amputation who can wear glove on residual limb

Disadvantages:

- Requires intact hand or sufficient residual limb to wear glove
- Can't detect force, only position
- Glove must be worn continuously during use

Wiring: Flex sensor in voltage divider configuration. As flex sensor bends, resistance increases, causing voltage at ADC pin to increase. Map voltage reading to servo position.

8. Software and Control Architecture

Input → Processing → Output Flow

A prosthetic control system follows a standard embedded control loop:

1. Input Acquisition (10-20% of compute time):

- Read sensor values (EMG, flex sensors, buttons)
- Perform analog-to-digital conversion
- Timestamp data if needed

2. Signal Processing (30-50% of compute time):

- Filter signals (remove noise)
- Extract features (e.g., EMG amplitude, rate of change)
- Buffer data if required for pattern recognition

3. Decision Logic (20-40% of compute time):

- Interpret processed signals to determine intent
- For pattern recognition: classify into movement categories
- For threshold control: compare to thresholds

- For proportional control: map signal amplitude to output magnitude

4. Control Output (10-20% of compute time):

- Generate control signals for actuators
- For servos: calculate PWM pulse width (1-2 ms)
- Apply safety limits, rate limits, and error checking

5. Update Actuators (minimal time):

- Send signals to motors
- Update status LEDs, display, etc.

Loop rate: For responsive control, aim for 50-100 Hz update rate (10-20 ms per loop iteration). Higher rates (200+ Hz) provide smoother control but require faster processors.

Basic Firmware Architecture

Initialization (setup phase):

- Initialize serial communication for debugging
- Configure pin modes
- Attach servos
- Set initial positions (safety: open hand, neutral wrist)
- Initialize filters, variables

Main loop:

- Read inputs
- Process signals
- Decision logic
- Smooth transition (avoid jerky motion)
- Safety check
- Output to actuators
- Optional status reporting
- Maintain loop rate timing

Control Logic

Threshold-based hand control:

Define threshold values. If EMG signal is below open threshold, hand opens. If above close threshold, hand closes. Between thresholds, maintain previous position (hysteresis zone prevents oscillation).

Proportional control with two EMG channels:

Calculate net signal (flexor EMG minus extensor EMG). Map this net signal to hand position, with negative values opening hand and positive values closing.

State machine for multi-function control:

Implement states (idle, hand control, wrist control). A mode button transitions between states. In each state, EMG input controls different joints.

Safety Checks and Fallback States

Critical safety features:

1. Motor current limiting: Prevent motor burnout and battery damage. If motor current exceeds maximum, stop motor and set error state.

2. Position limits: Prevent mechanical damage from over-extension. Constrain servo position to safe range.

3. Watchdog timer: Detect code lockups and automatically reset system.

4. Sensor validation: Detect sensor failures. Check for impossible values (sensor disconnected?).

5. Battery voltage monitoring: Track battery voltage. Warn user when battery is low (flash LED). Emergency shutdown if voltage critical.

6. Graceful degradation: If one component fails, maintain partial functionality rather than complete failure. Example: If wrist rotation servo fails, disable wrist control but keep hand function operational.

9. Gripper and Hand Design

Finger Kinematics

Human fingers have complex kinematics with 3 joints per finger (4 for the thumb including the carpometacarpal joint). Replicating this precisely is unnecessary for most grasps.

Simplified finger models:

1-DOF finger: Single joint (MCP), finger is rigid beyond that point. Extremely simple but very limited—can only achieve hook-like grasps.

2-DOF finger: MCP and PIP joints. DIP is either rigid or passively coupled to PIP motion. This provides adequate shape adaptation for most objects.

Coupled joint motion: In natural fingers, the DIP joint typically flexes about 2/3 as much as the PIP joint. This can be replicated with:

- **Mechanical coupling**: Four-bar linkages that automatically move DIP when PIP moves
- **Passive coupling**: Tendon routing that pulls DIP after PIP has flexed a certain amount

Joint axis design:

Real finger joints have evolving axes (the center of rotation shifts slightly during motion). For prosthetics, a simple pin joint with fixed axis is acceptable.

Design considerations:

- Pin diameter: 2-3mm for PLA/PETG fingers
- Bearing surface: Consider adding PTFE washers or small ball bearings to reduce friction
- Lateral constraint: The pin must prevent sideways motion while allowing rotation

Tendon-Driven Mechanisms

Why tendons instead of direct motor drive?

- **Remote actuation**: Motors can be in the forearm, not in fingers, reducing distal weight
- **Compliance**: Tendon stretch provides some shock absorption
- **Simplicity**: One tendon can actuate multiple joints in a finger

Tendon routing:

The tendon path from motor to fingertip determines mechanical advantage and joint coupling.

Key principle: The tendon should route along the flexor (palm) side of each joint, at a distance from the joint axis. This creates a moment arm—pulling the tendon applies torque to rotate the joint.

Distance from joint axis (moment arm) affects force:

- Larger moment arm (tendon farther from axis) = more torque but more tendon travel required
- Smaller moment arm = less torque but less tendon travel

Typical values: 5-10mm moment arm at MCP, 3-5mm at PIP/DIP.

Tendon materials:

- **Fishing line (monofilament)**: Common, inexpensive ($5 for 100m), 10-20 lb test is adequate. Low stretch, but can abrade over time
- **Braided fishing line**: Higher strength, more flexible, but stretches more
- **UHMWPE thread (Spectra/Dyneema)**: Very high strength-to-diameter, minimal stretch, excellent choice but more expensive ($15-25 for 50m)
- **Steel cable**: Very strong, no stretch, but difficult to work with and can kink

Tendon guides: Small eyelets or channels that constrain tendon path. Can be 3D-printed as part of finger structure or use small metal eyelets. Minimize friction—smooth surfaces, large radii on bends.

Tendon pretension: Some initial tension is needed to eliminate slack, but excessive pretension causes:

- Increased friction
- Difficulty opening fingers (return mechanism must overcome pretension)
- Faster wear

Typical pretension: 0.5-2N per finger.

Return mechanism (finger extension):

- **Elastic bands/springs**: Simple, reliable. Use rubber bands or silicone tubing on the dorsal (back) side of finger
- **Antagonistic tendons**: Second tendon on dorsal side actively pulls finger open. More control but requires twice as many motors
- **Torsion springs at joints**: Provide consistent return torque

Adaptive vs. Rigid Gripping

Rigid gripping: Fingers move to a predetermined position regardless of object presence. If object is smaller than the grip aperture, grasp fails.

Adaptive gripping: Fingers conform to object shape through underactuated mechanisms.

How underactuation works:

A single motor drives multiple fingers (or multiple joints within a finger) through a differential mechanism (whiffletree, seesaw, pulley system, or gear differential).

When any finger/joint contacts an object, it stops moving while the mechanism automatically redistributes force to remaining free fingers/joints. This achieves shape adaptation without sensors or complex control.

Example: Whiffletree differential:

A whiffletree is a tree-like structure of pivoting beams. Input force at the base distributes equally to all output branches. When one branch stalls (finger contacts object), force redistributes to other branches.

Advantages:

- Stable grasps on irregular objects
- Maximizes contact area (more fingers touch object)
- Simple control (single motor)
- Inherently robust

Disadvantages:

- Cannot control individual fingers independently
- More complex mechanism than direct drive
- Some force transmission losses

Design trade-off: For a student project, a simple adaptive mechanism (even just elastic bands allowing fingers to passively conform) provides significant benefit with minimal complexity.

Real-World Issues

Friction:

Tendon-driven systems experience friction at every guide point. Cumulative friction can consume 30-40% of input force.

Mitigation:

- Minimize number of direction changes in tendon path
- Use smooth, low-friction materials (PTFE/Teflon inserts in guides)
- Lubricate tendons (silicone spray)
- Use larger-radius pulleys (reduces bending friction)

Slipping:

Without force feedback, users can't feel when a grasped object is slipping.

Solutions:

- **Slip detection sensors**: Measure vibrations during slip events using accelerometers or microphones
- **Force control**: Maintain consistent grip force using force sensors and PID control
- **Texture on fingertips**: Increase friction coefficient with rubber coatings or textured surfaces

Wear:

Repeated flexing causes wear at:

- Tendon guide channels (abrasion grooves develop)
- Joint pins (holes elongate)
- Tendon itself (fraying, eventual failure)

Mitigation:

- Use harder materials at wear points (nylon or metal inserts in channels)
- Oversize holes slightly and use bushings
- Inspect and replace tendons periodically (check for fraying)

ISO 22523 specifies greater than 1.2 million cycles for clinical devices. Student projects should target at least 24,000 cycles (4 years equivalent at typical usage).

10. User Comfort and Human Factors

Socket Design: The Most Critical Component

The socket is the interface between the residual limb and the prosthetic device. It is simultaneously the most important component for user acceptance and the most difficult to get right.

Poor socket fit causes:

- **Pain and pressure sores**: Can develop in hours of use, preventing device use
- **Instability**: Prosthetic shifts during use, degrading control
- **Skin breakdown**: Abrasion, blisters, ulcers from shear forces
- **Edema (swelling)**: Fluid accumulation from poor circulation
- **Device abandonment**: Up to 75% of myoelectric prosthetic abandonment is due to socket issues

Why socket design is hard:

Every residual limb is unique—different shape, length, muscle mass, bone prominence locations, tissue compliance. The socket must:

- Distribute pressure over maximum surface area (load sharing)
- Avoid excessive pressure on bony prominences (less than 100 kPa) and neurovascular bundles
- Provide stable suspension (keep prosthetic attached)
- Allow muscle contraction for myoelectric control without electrode displacement
- Manage heat and perspiration
- Be comfortable for 8+ hours of daily wear

Socket design approaches:

Total Surface Bearing (TSB): Contact over the entire residual limb surface, distributing pressure broadly. Requires custom molding to limb shape. Typical pressures: 50-100 kPa average, less than 205 kPa maximum.

Patellar Tendon Bearing (PTB): For lower limb, emphasizes loading on patellar tendon and other load-tolerant areas. Upper limb equivalent focuses on specific muscle groups.

Custom socket fabrication (clinical process):

1. **Casting**: Create plaster or alginate cast of residual limb
2. **Positive model**: Fill cast to create positive model (replica of limb)
3. **Modification**: Skilled prosthetist modifies model to add/remove material at specific locations (relieving bony prominences, increasing pressure on muscle bellies)
4. **Thermoforming**: Heat plastic sheet and drape over modified model to create socket
5. **Fitting and refinement**: Try on patient, mark pressure points, iteratively adjust

This process takes 3-8 hours of prosthetist time and requires extensive training (master's degree + certification).

For student projects:

Critical limitation: Without training and equipment, you cannot create a safe, comfortable socket for actual use. Academic projects typically:

- Skip socket entirely, using the hand/arm component only for demonstrations
- Create a simplified attachment (e.g., Velcro strap) for short-term testing only
- Use a mannequin or simple cylindrical form representing a residual limb

If you attempt a simplified socket:

- 3D scan the residual limb (or use a generic model)
- Print in flexible material (TPU) or with mesh/perforated structure for breathability
- Include wide straps to distribute suspension forces
- **Clearly label as research prototype, not for extended wear**
- Test for only short durations (less than 30 minutes) and monitor skin for redness/irritation

Pressure Distribution and Load Bearing

Pressure = Force / Area (measured in kPa or psi).

The human body tolerates different pressure levels at different locations:

- **Muscle tissue**: Can tolerate 100-150 kPa for extended periods
- **Bony prominences**: Should not exceed 50-80 kPa to prevent tissue damage
- **Neurovascular areas**: Minimize pressure (less than 50 kPa) to avoid nerve damage and circulation restriction

Pressure mapping uses arrays of force sensors to visualize pressure distribution. Research devices use 15-50 sensors; for student projects, even 3-6 FSR sensors can provide useful feedback.

Sensor placement (for pressure monitoring in socket):

- Patella tendon (or equivalent load-bearing area)
- Distal end of residual limb (often experiences high pressure)
- Bony prominences (fibula head, ulnar styloid process, etc.)
- Medial and lateral sides

Design goals:

- Coefficient of variation (CV) in pressure across sensors: Lower is better. CV less than 50% indicates reasonably uniform distribution
- Peak pressure less than 205 kPa during gait (or arm movement)
- Average pressure 50-100 kPa

Improving pressure distribution:

- Increase contact area (more surface bearing)
- Add padding at high-pressure points (gel liners, foam inserts)
- Stiffen socket walls to prevent localized deformation that concentrates pressure
- Adjust socket alignment to change load distribution

Heat, Sweat, and Long-Term Wear

Thermoregulation challenges:

Prosthetic sockets trap heat and moisture against the skin. Sweat cannot evaporate, leading to:

- Discomfort (skin maceration, "pruning")
- Reduced electrode-skin contact quality (for myoelectric control)

- Increased friction and risk of skin breakdown
- Odor and hygiene issues

Design strategies:

Ventilation: Perforated socket designs with holes or mesh allow air circulation. Trade-off: reduces structural strength.

Materials:

- Breathable fabrics in liners (moisture-wicking performance textiles)
- Gel liners with channels for moisture escape
- Activated carbon inserts for odor control

User practices:

- Daily socket cleaning (soap and water)
- Wearing thin sock or liner that can be changed
- Removing prosthetic periodically to allow skin to dry
- Dusting with talcum powder to reduce friction

Long-term wear concerns (for clinical devices):

- **Skin breakdown**: Requires weeks to heal, preventing prosthetic use during that time
- **Residual limb volume changes**: Weight loss/gain, muscle atrophy, edema cause socket fit to degrade over time (6-12 months)
- **Material degradation**: Liners compress and lose resilience, requiring replacement
- **Psychological impact**: Discomfort leads to reduced wear time, which limits functional benefit and can lead to abandonment

Adjustability

Why adjustability matters:

Residual limb volume can change by 5-10% throughout the day due to:

- Edema (swelling) when limb is dependent (hanging down)
- Fluid redistribution when limb is elevated
- Activity level (muscle pump during use vs. inactivity)

A socket that fits perfectly in the morning may be loose by afternoon.

Adjustment mechanisms:

Liner thickness: Multiple liner thicknesses (3mm, 6mm, 9mm) allow user to swap as limb volume changes.

Adjustable straps/BOA systems: Mechanical dial that tightens cables for micro-adjustments.

Air bladders: Inflatable chambers in socket that can be pumped up for tighter fit.

Modular socket systems: Outer rigid frame with replaceable inner socket that can be swapped as limb volume changes.

For student projects, simple Velcro straps with multiple attachment points provide basic adjustability.

11. Testing and Validation

Basic Testing Protocols

Even without access to ISO-certified testing labs, student projects should undergo structured validation.

Test categories:

1. Component-level testing: Test individual subsystems before integration.

- **Servo motors**: Verify torque, speed, current draw match specifications. Load test by attaching weights.
- **Sensors**: Calibrate and verify repeatability. Check drift over time.
- **Microcontroller**: Verify code execution time, ensure no memory leaks, test watchdog functionality.

2. Integration testing: Test combined subsystems.

- **Power distribution**: Measure voltage drops under load, verify all components receive proper voltage.
- **Signal integrity**: Measure EMG signal quality with full system running (check for motor noise coupling).
- **Control loop timing**: Verify control loop maintains target frequency under full computational load.

3. Functional testing: Test the complete system performing intended tasks.

- **Grasp tests**: Attempt to grasp objects of varying sizes, shapes, weights.
- **ADL simulation**: Perform realistic tasks (picking up cup, opening door, etc.).
- **User interface**: Test control responsiveness, intuitiveness.

4. Durability testing: Assess reliability and longevity.

- **Fatigue cycling**: Repeatedly open/close hand 1000+ times, check for failures.
- **Wear inspection**: Examine tendons, joint surfaces for wear after testing.

Mechanical Stress Testing

Finger hyperextension test:

Apply lateral force to finger to simulate impact. Should withstand 30N without structural failure.

Method:

- Fix hand in test jig
- Apply measured force perpendicular to finger (use force gauge or weights on lever arm)
- Slowly increase force until failure or 30N reached
- Record failure mode (if occurs): fracture location, delamination, pin failure, etc.

Hand tensile test (ISO 22523 Annex A):

Grasp a 19mm diameter cylinder and pull until the hand fails to maintain grip.

Method:

- Mount prosthetic hand in tensile testing machine (or use hanging weights)
- Actuate hand to close around cylinder
- Apply increasing tensile force (pulling cylinder away from hand)
- Record maximum force before grasp failure

Pass criterion: Greater than 110N for basic functionality.

Finger flexion cycling test:

Repeatedly flex and extend fingers to assess fatigue life.

Method:

- Fully automated: Connect motor control to loop that cycles hand open/close continuously
- Count cycles with sensor or in code
- Run until component failure or target cycle count
- Target: 24,000 cycles (equivalent to approximately 4 years of typical use: 6000 grasps/year × 4 years)

Monitor for:

- Tendon wear/fraying
- Joint looseness (pin holes elongating)
- Motor failure
- 3D-printed part cracks

Failure Mode Analysis

Common failure modes in student prosthetic projects:

Mechanical failures:

- **Tendon breakage**: Usually occurs at crimp points or where tendon bends sharply. Prevention: Use stronger line, avoid sharp bends, inspect regularly.
- **Gear stripping**: Servo gears (especially plastic gears in cheap servos) can strip under overload. Prevention: Don't exceed motor torque ratings, include mechanical hard stops.
- **Layer delamination**: 3D-printed parts fail between layers. Prevention: Orient print layers parallel to stress direction, increase perimeter count, anneal parts.
- **Pin/bearing wear**: Holes elongate, causing slop and misalignment. Prevention: Use metal pins, add bushings, increase pin diameter.

Electrical failures:

- **Battery overdischarge**: Li-Po batteries damaged by discharging below 3.0V/cell. Prevention: Voltage monitoring and cutoff circuit.
- **Servo burnout**: Motor overheating from stall or continuous load. Prevention: Current limiting, timeout logic (disable motor after N seconds at stall).
- **Loose connections**: Vibration causes wires to disconnect. Prevention: Solder connections, use hot glue or zip ties for strain relief.

Control failures:

- **EMG electrode shift**: Electrodes move relative to muscle, changing signal characteristics. Prevention: Secure electrode placement, use multiple electrodes, implement electrode-shift detection algorithms.
- **Signal noise**: EMI from motors corrupts sensor signals. Prevention: Separate power and signal wiring, use filtering, shielding.
- **Software bugs**: Code crashes, memory leaks, infinite loops. Prevention: Extensive testing, watchdog timers, defensive programming.

Failure mitigation:

- **Graceful degradation**: Design system to fail to safe state (e.g., hand opens when power lost, not closes)
- **Redundancy**: Use multiple sensors where critical

- **User feedback**: Provide visual/haptic indication of errors

Improving Precision and Repeatability

Sources of variability:

- **Sensor noise**: ±5-10% variation in EMG readings from same muscle contraction
- **Mechanical backlash**: Gears and joints have small amounts of play (0.5-2mm)
- **Servo deadband**: Servos have a small zone (±1-2°) where they don't respond to small position changes
- **Environmental changes**: Temperature affects motor performance, EMG electrode conductivity

Improvement strategies:

1. Sensor calibration:

- **Baseline calibration**: Record sensor values at rest and at maximum contraction, map full range
- **Dynamic recalibration**: Periodically update calibration during use to adapt to changing conditions
- **Multi-point calibration**: Calibrate at 3-5 different contraction levels for nonlinear sensors

2. Filtering and averaging:

- Apply moving average or exponential smoothing to sensor readings
- Use longer averaging windows for slower-changing signals (position) and shorter for fast signals (impact detection)

3. Hysteresis in thresholds:

- Use different thresholds for increasing vs. decreasing signals
- Example: Open hand when EMG is less than 200, close when EMG is greater than 400. In between, maintain previous state
- This prevents rapid oscillation at threshold boundaries

4. Predictive control:

- Use signal derivatives (rate of change) to anticipate user intent
- If EMG is increasing rapidly, start closing hand before reaching full threshold

5. Mechanical improvements:

- Minimize backlash with spring-loaded anti-backlash gears
- Use higher-quality servos with metal gears
- Tighten tolerances in 3D prints (reduce clearances)

Metrics for precision:

- **Position repeatability**: Command same position 10 times, measure standard deviation. Target: less than 5% of full range
- **Force consistency**: Grip with target force 10 times, measure variation. Target: less than 10% of target force
- **Response time**: Time from input signal to 90% of final position. Target: less than 500ms for natural feel

12. Cost, Practicality, and Realistic Scope

What Can Be Done in a Student Project

Realistic capabilities for $300-400 budget, 3-6 months part-time:

Mechanical:

- 3D-printed hand with 2-4 fingers + thumb
- 2-3 powered DOF (e.g., hand open/close + wrist rotation + thumb opposition)
- Tendon-driven or direct servo actuation
- Simplified socket or no socket (hand component only)

Control:

- Switch-based, joystick-based, or basic threshold EMG control
- Arduino or ESP32 microcontroller
- 2-3 EMG sensors or alternative input (flex sensors, buttons)

Functionality:

- Demonstrate basic cylindrical and pinch grasps
- Hold objects 50-500g
- approximately 1-2 hours battery life
- Function for dozens to hundreds of actuation cycles

Testing:

- Basic functional tests (grasp common objects)
- Mechanical stress testing of components
- Short-duration user trials (simulated, not actual amputees)

Deliverables:

- Working prototype
- CAD files and code (open-sourced)
- Documentation (build guide, test results)
- Demonstration video

What is NOT realistic:

- Clinical-grade reliability (thousands of hours of use)
- Custom-fitted socket for extended wear
- Advanced pattern recognition control (possible but challenging)
- More than 5 powered DOF on this budget
- Simultaneous control of multiple DOF without advanced algorithms
- Sensory feedback (haptic, temperature sensing)
- FDA/CE approval or clinical trials

Budget Breakdown

Baseline functional prototype: approximately $300-400

Materials:

- PLA filament (1.5kg): $30

- PETG filament (0.5kg): $12.50
- Hardware (screws, pins, springs): $15
  Subtotal: $57.50

Electronics:

- MG995 servos (5x): $50
- MG996R servos (3x): $30
- Arduino Uno: $15
- EMG sensor modules (3x): $75
- Li-Po battery 2600mAh 2S (2x): $30
- Wiring, breadboard, connectors: $18
  Subtotal: $218

Miscellaneous:

- Adhesives, tools, consumables: $25
  Subtotal: $25

Total: approximately $300

Low-cost alternative (approximately $100-150):

- Salvage filament from recycling or donors
- Use SG90 micro servos ($3 each) instead of MG995/996R
- Arduino clone board ($8)
- DIY EMG amplifier circuit ($5 components) instead of modules
- Salvage battery from old laptop or power tool
- Makerspace/university lab access for tools and materials

Trade-off: Lower-quality servos have less torque and durability, DIY EMG requires more electronics knowledge, but proves concept at 1/3 cost.

High-capability student project (approximately $800-1200):

- Multiple high-torque digital servos
- ESP32 with pattern recognition
- Force sensors, accelerometers, additional I/O
- Higher-capacity batteries
- Better materials (nylon, carbon fiber inserts)
- Possible custom PCB instead of breadboard

Approaches commercial prosthetic capability in some respects but still lacks socket design and long-term reliability.

Trade-offs: Accuracy vs. Simplicity

Design decisions invariably involve compromise:

Number of DOF:

- More DOF = better function, more versatility, closer to natural hand
- Fewer DOF = simpler control, lower cost, higher reliability, easier for users to learn
  Student choice: Start with 2-3 DOF, validate functionality, then add more if time permits

Control complexity:

- Advanced control (pattern recognition) = more natural, can control multiple DOF
- Simple control (threshold/switch) = robust, low latency, easier to debug
  Student choice: Threshold control for first prototype, explore pattern recognition if interested and time allows

Materials:

- High-strength materials (nylon, carbon fiber) = durable, professional
- PLA = easy to print, cheap, sufficient for proof-of-concept
  Student choice: PLA for most parts, PETG for high-stress components like fingers

Actuation:

- Independent motors per finger = full control, ability for complex gestures
- Underactuated mechanisms = simpler, lighter, adaptive grasping
  Student choice: Underactuated for first build, demonstrates mechanical intelligence

Testing thoroughness:

- Extensive testing (ISO 22523) = confidence in reliability, publishable data
- Basic functional testing = validates concept, sufficient for academic project
  Student choice: Core functional tests plus 1-2 stress tests, document thoroughly

Budgeting Parts and Time

Parts procurement strategy:

Month 1: Design phase

- Don't buy anything yet
- Finalize CAD design
- Create detailed bill of materials (BOM) with part numbers, suppliers, costs
- Identify alternatives for each component

Month 2: Initial purchase

- Buy 1 set of electronics to test control system (breadboard prototype)
- Buy minimal filament for test prints (0.5kg)
- Validate electrical design works before committing to full build

Month 3: Full build

- Once design is validated, purchase remaining components
- Buy slightly more filament than calculated (15-20% extra for failed prints)

Time budget (typical 20-week timeline):

- Weeks 1-3: Research, requirements, learning CAD/electronics (30 hours)
- Weeks 4-7: Design and analysis (40 hours)
- Weeks 8-12: Manufacturing and assembly (50 hours)
- Weeks 13-16: Control system implementation and tuning (40 hours)
- Weeks 17-20: Testing, refinement, documentation (30 hours)
  Total: approximately 190 hours (approximately 10 hours/week)

Common time underestimates:

- 3D printing time: Can take 50+ hours of print time for full hand
- Failed prints: Budget 20-30% extra time for reprints

- Debugging control issues: Often takes 2-3× longer than anticipated
- Documentation: Takes longer than you think

Team structure (if group project):

- Mechanical lead: CAD design, manufacturing, assembly
- Electrical lead: Circuit design, wiring, power system
- Software lead: Control algorithms, testing
- Integration lead: System-level testing, documentation

Overlap and cross-training are essential—everyone should understand all subsystems.

13. Ethical and Accessibility Considerations

Designing for Inclusivity

User-centered design is critical:

Prosthetic devices must serve the needs of actual users, not just satisfy engineering curiosity. This means:

Involve users early and often:

- Consult with amputees during design phase (if ethically possible and approved)
- Review literature on user needs and priorities
- Understand that functional priorities vary: some users prioritize cosmesis, others prioritize function

Prioritize user needs over technical novelty:

Research shows users care most about:

1. Pain reduction and comfort
2. Ability to perform ADLs
3. Reliability (device works when needed)
4. Social acceptance/aesthetics
5. Cost/accessibility

Advanced features (e.g., 15 DOF, AI control) are lower priority than these fundamentals.

Design for diverse users:

- **Size variation**: People have different hand sizes. Parameterize CAD models to scale for different users.
- **Bilateral amputees**: Some control methods (flex sensors on intact hand) don't work for bilateral users.
- **Cultural considerations**: Aesthetic preferences, functional needs vary across cultures.
- **Economic accessibility**: Design should be achievable at low cost for resource-limited regions.

Open-source approach:

Publishing designs, code, and documentation openly enables:

- Global collaboration and improvement
- Adaptation to local needs and resources
- Educational use
- Reduced cost (no proprietary licensing)

Many student prosthetic projects contribute to open-source repositories.

Real Human Use vs. Educational Prototypes

Critical ethical distinction:

Educational prototypes are NOT medical devices. They lack:

- Clinical validation and safety testing
- Regulatory approval (FDA, CE)
- Professional fitting and customization
- Liability insurance
- Long-term support and maintenance

Student projects should:

- Clearly label prototypes as "research/educational use only, not for medical use"
- Never claim medical efficacy or therapeutic benefit
- If demonstrating on a person, use informed consent and limit to brief, supervised sessions
- Not promise users a functional prosthetic—manage expectations
- Acknowledge limitations explicitly in documentation

If transitioning to real use (beyond academic scope):

Requires:

- Partnership with certified prosthetists and clinicians
- IRB approval for human subjects research
- Compliance with medical device regulations (ISO 13485, FDA 21CFR820)
- Clinical trials demonstrating safety and efficacy
- Product liability considerations
- Sustainable maintenance and support model

This is a multi-year, multi-million-dollar endeavor not achievable in a student project timeline.

Safety Responsibilities

Engineering ethics principles:

From IEEE Code of Ethics:

- "Hold paramount the safety, health, and welfare of the public"
- "Disclose promptly factors that might endanger the public or the environment"

Applied to prosthetics:

Design for safety:

- Fail-safe defaults (hand opens on power loss, not closes and traps)
- No sharp edges or pinch points
- Current limiting to prevent burns
- Overload protection to prevent mechanical damage

Testing for safety:

- Verify no electrical shock hazards (proper insulation, grounding)
- Check maximum forces/pressures don't exceed safe limits

- Test emergency stop functionality
- Simulate failure modes (what happens if sensor fails? If battery dies?)

User safety:

- Provide clear instructions and safety warnings
- Never use a device on a person without their informed consent
- Stop immediately if any adverse event (pain, injury, equipment damage)
- Have emergency plan (how to quickly remove device)

Do no harm: If your prototype might cause injury, don't test on people. Use mannequins, robotic test rigs, or carefully controlled simulations instead.

14. Final Summary and Roadmap

Key Principles for Success

1. Start simple, iterate to complex: A working 2-DOF hand is infinitely better than a non-functional 18-DOF hand. Get basic functionality working, then add features.

2. Test early and often: Don't wait until the end to test. Validate each subsystem before integration. Find problems early when they're easy to fix.

3. Document everything: Keep lab notebook, photograph build process, save all code versions. Future you will thank present you. Also helps others learn from your work.

4. Embrace failure: Every failed print, burned-out servo, and debugging session teaches you something. The goal is learning, not perfection.

5. Focus on fundamentals: Don't chase novel features before mastering basics. Solid mechanical design + reliable control beats fancy algorithms with poor execution.

6. Respect the human aspect: This technology serves people with limb differences. Approach with humility, empathy, and awareness of their lived experience.

Do's and Don'ts

DO:

- ✓ Calculate torque requirements before selecting motors
- ✓ Use safety factors (1.4-1.5) in all calculations
- ✓ Filter EMG signals thoroughly (high-pass, low-pass, notch)
- ✓ Test components individually before system integration
- ✓ Include current limiting and voltage monitoring
- ✓ Print test pieces to validate design before full print
- ✓ Version control your code (Git)
- ✓ Calibrate sensors at start of each session
- ✓ Build in adjustability where possible
- ✓ Document assumptions and design decisions

DON'T:

- ✗ Undersize motors—calculate properly or oversize

- ✗ Ignore weight—every gram matters at the end of limb
- ✗ Skip mechanical stress testing—know failure modes
- ✗ Power servos from microcontroller pins—use external power
- ✗ Overdischarge Li-Po batteries—monitor voltage
- ✗ Print fingers perpendicular to stress—layer orientation matters
- ✗ Attempt custom socket without training—it's the hardest part
- ✗ Promise functionality you can't deliver—manage expectations
- ✗ Test on people without consent and safety review
- ✗ Ignore signal noise—it will ruin control

Development Roadmap for Beginners

If you're starting today (20-week timeline):

Weeks 1-3: Foundation

- Read this documentation thoroughly
- Study existing open-source designs
- Learn CAD software (Fusion 360, SolidWorks, FreeCAD)
- Learn Arduino basics (blink LED, read sensors, control servo)
- Define requirements for your project

Weeks 4-7: Design

- CAD model of hand/arm mechanism
- Torque calculations → motor selection
- Circuit schematic
- Create bill of materials (BOM)
- Order initial electronics for testing

Weeks 8-12: Build

- 3D print components (start printing early—takes time)
- Assemble mechanical structure
- Build control circuit on breadboard
- Test individual components

Weeks 13-16: Control

- Write control code (start simple: manual control)
- Add sensor input (EMG or alternative)
- Tune control parameters (thresholds, PID gains)
- Integrate all systems

Weeks 17-20: Test & Document

- Functional testing (grasp objects, simulate ADLs)
- Mechanical stress testing (cycling, load tests)
- Refine based on results
- Create final documentation and demonstration

Beyond week 20: If continuing, focus on:

- Improving reliability (more testing, failure analysis)
- Adding features (additional DOF, force control)
- Publishing results (paper, presentation, open-source release)

You now have the foundation to design and build a prosthetic arm. The path forward is challenging—you'll encounter setbacks, design flaws, and technical hurdles. That's engineering. Every problem you solve builds competence. Every iteration improves the design. Every test reveals something new.

The comprehensive knowledge contained in this guide covers biomechanics, mechanical design, actuation systems, control theory, electronics, signal processing, sensors, firmware architecture, materials science, testing protocols, and human factors. Use this documentation as your reference throughout your project. Return to specific sections as needed, validate calculations, and always prioritize safety and realistic scoping.

This is not just a technical endeavor—you're working to create tools that could improve quality of life for people with limb differences. Approach the work with technical rigor, genuine empathy, and recognition of what's achievable within your constraints. A well-executed, honest prototype that clearly demonstrates one principle is worth infinitely more than an overcomplicated design that fails and teaches nothing.

Begin with clear eyes about what you can accomplish, test thoroughly, document honestly, and learn deeply. That is how you become an engineer.

References

Sources used in developing this comprehensive technical documentation include research from peer-reviewed journals on prosthetic design, biomechanics, and control systems; clinical literature on prosthetic assessment and user requirements; open-source prosthetic projects and their documentation; IEEE standards on robotic systems; ISO 22523 medical prosthetics standards; commercial prosthetic manufacturer technical specifications; control systems and signal processing textbooks; materials science literature on 3D printing polymers; microcontroller manufacturer datasheets; and established practices in rehabilitation engineering. This documentation synthesizes practical engineering knowledge with clinical context to provide students with actionable guidance for building functional prosthetic systems.