Architectural Analysis of Advanced CNC Panel Bending Control Systems: A Reverse-Engineering Perspective on the Salvagnini P4/P2 Platform

1. Executive Summary and System Definition

The domain of automated sheet metal forming has evolved from rigid, hard-coded NC (Numerical Control) systems to flexible, adaptive Cyber-Physical Systems (CPS). At the forefront of this evolution is the automatic panel bender, typified by the Salvagnini P4 and P2 series. Unlike press brakes, which rely on a simple vertical ram movement into a V-die, panel benders utilize a complex interpolated kinematic linkage to fold metal sheets clamped by a blankholder. This process requires a control architecture capable of synchronizing multi-axis trajectories with sub-millisecond precision while simultaneously compensating for material variability in real-time.

For a Senior Control Systems Engineer tasked with developing a proprietary controller for such a machine, understanding the incumbent architecture is critical. The Salvagnini system represents a paradigm shift from proprietary ASIC-based CNCs (such as legacy Fanuc or Siemens hardware) toward a **PC-Based Control architecture**. This architecture leverages standard industrial PCs (IPCs), real-time operating system (RTOS) extensions, and high-speed deterministic fieldbuses—specifically **EtherCAT**—to achieve the necessary bandwidth for advanced algorithmic control [1, 2, 3].

The core intellectual property of these machines lies not in the commodity hardware components (motors, drives, PCs), but in two specific software-defined capabilities:

1. **The "Bending Formula"**: A kinematic solver that generates non-linear, rolling-contact trajectories for the bending blades to minimize material marking and optimize fold geometry [4].
2. **MAC 2.0 (Material Adaptive Correction)**: A real-time adaptive control loop that infers material properties (tensile strength, yield point) from motor torque feedback during the elastic deformation phase, adjusting the final axis position to compensate for springback without external measuring devices [5, 6, 7].

This report deconstructs the hardware topology, software stack, kinematic mathematics, and sensor feedback loops of the P4 (Hybrid Hydraulic/Electric) and P2 (All-Electric) platforms. It is intended as a foundational technical reference for replicating and improving upon these control strategies using modern tools such as C++, Python, and the EtherCAT ecosystem.

2. Hardware Topology and Fieldbus Architecture

The hardware topology of a modern high-performance panel bender is defined by the requirement for centralized computation with distributed actuation. The controller must coordinate high-power hydraulic servo valves (in P4 models) or high-torque electric motors (in P2 models) with the precision of a CNC kernel.

2.1 The Central Processing Unit: PC-Based Control

The shift from dedicated NC hardware to PC-based control is a defining characteristic of Salvagnini's architecture [3, 8]. The proprietary "SIX" controller or its successors are built upon standard Industrial PC hardware, likely supplied by **Beckhoff Automation** given the pervasive use of the EtherCAT protocol and TwinCAT software references [9, 10, 11].

2.1.1 Industrial PC (IPC) Specifications

The likely hardware core is a **Beckhoff C69xx or C60xx series IPC** [11, 12]. These units are cabinet-mounted, fanless industrial computers designed to withstand the vibration and thermal environments of a factory floor (0-55°C operating range).

- **Processor Architecture:** Intel Core i7 or Xeon multi-core processors are standard. Multi-core is essential for the **CPU Isolation** strategy used in real-time control.
- **Core Isolation:** In a quad-core system, Windows 10 IoT Enterprise (hosting the HMI and CAM) is restricted to Cores 0 and 1. The Real-Time Kernel (TwinCAT Runtime) is pinned to Cores 2 and 3, running in kernel mode (Ring 0). This ensures that heavy HMI graphics processing or database queries do not induce jitter in the motion control loops [13].
- **Interfaces:** The IPC acts as the EtherCAT Master, utilizing a standard Intel Ethernet chipset (e.g., i210) dedicated solely to the fieldbus.

2.1.2 Why PC-Based?

The decision to use an IPC over a traditional PLC is driven by the computational intensity of the **MAC 2.0** algorithms. Calculating the stress-strain curve of a material in real-time, filtering signal noise, and solving high-order kinematic polynomials for trajectory generation requires floating-point performance that exceeds standard PLC capabilities [8, 13].

2.2 The Nervous System: EtherCAT Fieldbus

The backbone of the machine is **EtherCAT (Ethernet for Control Automation Technology)** [1, 2, 14]. Unlike polling-based buses (e.g., Modbus TCP) or token-passing buses (e.g., Profibus), EtherCAT uses a "processing on the fly" mechanism.

2.2.1 Criticality of Distributed Clocks (DC)

For a panel bender, the synchronization of the Upper Blade (A) and Lower Blade (D) is

paramount. These axes effectively perform a "handshake" or coordinated "rolling" motion. If they are out of sync by even 100 microseconds, the blade may skid on the material surface or apply uneven pressure.

EtherCAT's **Distributed Clocks (DC)** mechanism synchronizes the local time of every servo drive and I/O terminal to within << 1 µs of the master clock [2]. This allows the motion controller to define a precise "Action Time" for the setpoints of all axes, ensuring that the hydraulic valves of the P4 (which have different response times than electric motors) actuate in perfect unison with the electric manipulator.

2.2.2 Bus Cycle Time

The typical bus cycle time for high-dynamic bending is **250 µs to 1 ms** [13].

- **1 ms Cycle:** Sufficient for the Manipulator (X/Y) handling the sheet.
- **250 µs Cycle:** Required for the Bending Unit axes (A, D, B) during the active bend phase to capture the high-frequency torque ripple data needed for the MAC 2.0 adaptive control [15].

2.3 Drive and Actuation Systems

The P4 and P2 represents two different approaches to actuation, necessitating a controller that can abstract the physical drive layer.

2.3.1 Hybrid Actuation (P4 Series)

The P4 is a "Hybrid" machine. The bending force (up to 250 tons) is hydraulic, while sheet manipulation is electric [16, 17].

- **Hydraulic Servo Loop:** The controller does not just open/close valves; it manages **Servo-Hydraulic Valves**. These are connected via EtherCAT analog I/O terminals (±10V control) or direct fieldbus interfaces.
    - *Control Challenge:* Hydraulic systems are inherently non-linear due to fluid compressibility (bulk modulus) and temperature changes. The controller must implement advanced linearization tables and temperature compensation algorithms [17].
- **Electric Axes:** The manipulator and rotator use Brushless AC Servos.

2.3.2 All-Electric Actuation (P2 / P4 Lean)

The P2 is fully electric, using **Direct Drive** technology for the bending blades [4, 18].

- **Direct Drive Motors:** High-torque, low-RPM motors connect directly to the load without gearboxes.
    - *Significance:* This eliminates mechanical backlash and compliance. The current (torque) draw of the motor becomes a pure, high-fidelity signal of the bending force, which is the physical basis for the sensitivity of the **MAC 2.0** system. A gearbox

would introduce friction and noise, masking the subtle yield point signals [14].
- **Energy Recovery (KERS):** The drives share a common DC bus. When the heavy bending frame decelerates, the regenerative energy is stored in capacitor banks or fed to other axes (e.g., the manipulator accelerating) rather than being dissipated as heat in braking resistors [5, 19].

2.3.3 Servo Drive Vendors

Research indicates collaboration with **Kollmorgen (S700 Series)** and **Moog (DE2020/DR2020)** [20, 21]. These drives are chosen for their deep integration with EtherCAT, supporting modes like **CSP (Cyclic Synchronous Position)** and **CST (Cyclic Synchronous Torque)** which are essential for the hybrid control schemes used.

---

3. Axis Configuration and Criticality Analysis

Developing a controller requires a precise mapping of the machine's degrees of freedom. A standard panel bender is not a Cartesian robot; it is a collection of interacting kinematic chains.

3.1 The Manipulator Group (Sheet Handling)

These axes are responsible for presenting the sheet to the bending line.

- **X-Axis (Feeder):**
  - *Type:* Electric Servo (Rack and Pinion or Ball Screw).
  - *Function:* Longitudinal positioning of the sheet.
  - *Criticality:* High accuracy required for flange length precision.
- **Y-Axis (Centering):**
  - *Type:* Electric Servo.
  - *Function:* Transverse movement to square the sheet against referencing stops [22].
- **Z-Axis (P-Robot/Loader):**
  - *Type:* Electric/Pneumatic.
  - *Function:* Loading/Unloading. Often integrated with a Cartesian robot or storage tower [20, 23].
- **C-Axis (Rotator):**
  - *Type:* Electric Servo (Direct Drive or Gear).
  - *Function:* Rotates the sheet 0/90/180/270 degrees.
  - *Dynamics:* Highly variable inertia. A large sheet has massive rotational inertia compared to a small part. The controller must adapt the velocity/acceleration loop gains based on the estimated part inertia (Mass properties from CAD) [14].

3.2 The Bending Unit (The Core Mechanism)

This is the proprietary heart of the machine [17].

- **Axis A (Upper Blade):**
  - *Type:* Hydraulic (P4) or Electric (P2).
  - *Function:* Performs negative (down) bends.
  - *Trajectory:* Interpolated curvilinear path.
- **Axis D (Lower Blade):**
  - *Type:* Hydraulic (P4) or Electric (P2).
  - *Function:* Performs positive (up) bends.
  - *Trajectory:* Interpolated curvilinear path.
- **Axis B (Blankholder):**
  - *Type:* Hydraulic/Electric.
  - *Function:* Clamps the sheet against the Counterblade (C).
  - *Control:* Force control is critical here. It must hold the sheet firmly enough to prevent slipping but not so hard as to crush surface finishes or delicate forms.
- **Axis C (Counterblade):**
  - *Type:* Mechanical Reference.
  - *Function:* The anvil against which the bend is formed.

3.3 Auxiliary Automation Axes

- **ABA (Automatic Blankholder Adjustment):**
  - *Type:* Electric Servo [24, 25].
  - *Function:* Adjusts the width of the blankholder tool segments to match the part length. This eliminates manual setup time.
  - *Control:* Position control with torque limiting to detect "hard stops" when segments are fully expanded/contracted.
- **CLA (Auxiliary Blades):**
  - *Type:* Pneumatic/Electric.
  - *Function:* Modular blade segments that engage/disengage for interrupted bends (e.g., bending between tabs) [26, 27].

3.4 Axis Summary Table

Axis Designation	Common Name	Actuation (P4)	Actuation (P2)	Control Mode (EtherCAT)	Critical Dynamic Factor
X	Manipulator Long.	Servo Electric	Servo Electric	CSP	Positioning Accuracy
Y	Manipulator Trans.	Servo Electric	Servo Electric	CSP	Centering Logic

C	Rotator	Servo Electric	Servo Electric	CSP	Variable Inertia Handling
A	Upper Blade	Servo Hydraulic	Direct Drive Electric	CSP / CST	Interpolation / Force Feedback
D	Lower Blade	Servo Hydraulic	Direct Drive Electric	CSP / CST	Interpolation / Force Feedback
B	Blankholder	Servo Hydraulic	Servo Electric	Force / Position	Clamping Pressure Control
ABA	Tool Adjuster	Servo Electric	Servo Electric	CSP	Setup Speed

---

4. Software Stack Architecture

The software architecture follows a strict separation of concerns, typical of safety-critical industrial automation. This is a "Split-Architecture" design [3].

4.1 Layer 1: The Real-Time Motion Kernel (The "Brain")

This layer runs on the RTOS (e.g., **TwinCAT 3 Runtime**) [9, 10]. It has the highest priority and direct access to the EtherCAT hardware.

- **Execution Context:** Kernel Mode (Ring 0).
- **Programming Languages:** IEC 61131-3 (Structured Text) for state machines; C++ (TwinCAT C++ Module) for complex kinematic algorithms [10].
- **Responsibilities:**
  - **Trajectory Generation:** Converting abstract "Bend Angle" commands into 1ms setpoints for Axes A, D, X, Y.
  - **MAC 2.0 Execution:** Running the observer models that monitor torque and adjust trajectories.
  - **Safety Logic:** Handling E-Stops and light curtains (via TwinSAFE).
  - **Servo Loops:** Closing the Position Loop (and potentially Velocity loop) if the drive is in Torque mode.

4.2 Layer 2: The Application and HMI (The "Face")

This layer runs on **Windows 10 IoT**. It is not real-time deterministic.

- **HMI Software: FACE** or **JOB.CONSOLE** [28, 29]. Likely built on **.NET (C# / WPF)**.
- **Responsibilities:**
  - **Job Management:** Loading lists of parts (kits) [28].
  - **Visualization:** 3D rendering of the bending process for operator verification.
  - **Database:** SQL LocalDB or similar for storing tool libraries, material tables, and machine parameters.
  - **Diagnostics:** Displaying errors, maintenance logs [28].

4.3 Layer 3: Inter-Process Communication (IPC)

The bridge between the Windows HMI and the Real-Time Kernel is critical. In the Beckhoff ecosystem, this is handled by **ADS (Automation Device Specification)** [9].

- **Mechanism:** The HMI sends an asynchronous ADS Write command (e.g., "Load Job #123").
- **Response:** The RT Kernel accepts the command, buffers the motion vectors, and reports status back via ADS Notifications.
- **Design Note:** A proprietary controller must implement a similar shared-memory or mailbox architecture to prevent HMI lag from stalling the machine.

4.4 Layer 4: Offline CAM (STREAMBEND)

The intelligence of the machine begins before the operator touches it. **STREAMBEND** [27, 30, 31] is the offline programming suite.

- **Function:** It takes a 3D CAD file (STEP/IGES), unfolds it, and determines the **Bend Sequence**.
- **Collision Detection:** The software simulates the part flipping and rotating to ensure it doesn't hit the machine frame or the manipulator.
- **Code Generation:** Instead of standard ISO G-code (which is limited for this topology), it generates a proprietary XML or binary job file containing high-level parametric instructions (e.g., BEND_UP, ANGLE=90, RADIUS=2, SIDE=3).

---

5. Kinematics and The "Bending Formula"

The "Bending Formula" [4, 5, 26] is the mathematical engine that drives the interpolated movement of the blades. Standard circular interpolation (G02/G03) is insufficient for high-quality panel bending.

5.1 The Kinematics of Folding

In a V-die press brake, the punch moves in a straight line (Y-axis). In a panel bender, the blade must "roll" around the material to fold it without sliding. Sliding causes scratches ("galling") and damages the protective coating of the sheet [32].

5.1.1 Trajectory Geometry

The blade tip coordinates (y_b, z_b) are a function of the bend angle α.

To maintain rolling contact at a bend radius R, the kinematics can be modeled as:

$$y_b(\alpha) = R \cdot (1 - \cos(\alpha)) + T \cdot \sin(\alpha)$$

$$z_b(\alpha) = R \cdot \sin(\alpha) + T \cdot (1 - \cos(\alpha))$$

Where:

- α = Instantaneous bend angle.
- R = Desired internal radius.
- T = Material Thickness.

Note: This is a simplified model. The actual Salvagnini "Bending Formula" likely includes higher-order polynomial terms to account for the **Instantaneous Center of Rotation (ICR)** of the blade mechanism, which moves as the blade linkage extends [33].

5.2 Oscillating Blade Mechanism

The P4 uses an **oscillating blade** mechanism [17, 33]. This mechanical linkage allows the blade to change its angle of attack relative to the sheet.

- **Control Implication:** The controller must solve the **Inverse Kinematics** of this linkage.

  Given a desired blade tip position (y, z) and angle ϕ, what are the required positions of the linear actuators driving the linkage?
- **Equation Solver:** This requires a non-linear solver running in the real-time context, updating setpoints every cycle (e.g., 250 µs).

5.3 Dynamics and Jerk Control

Rapidly reversing the heavy bending frame (from Down-Bend to Up-Bend) induces massive jerk.

- **S-Curve Profiles:** The motion profile must use 3rd-order (cubic) or 5th-order (quintic) polynomials to ensure smooth acceleration and jerk limiting. This prevents hydraulic

hammer in P4s and mechanical oscillation in P2s.

---

6. Sensor Feedback and MAC 2.0 (Adaptive Control)

The **MAC 2.0 (Material Adaptive Correction)** system [5, 6, 7] is the most critical differentiator. It allows the machine to bend "Batch One" (single parts) correctly without trial-and-error.

6.1 The Problem: Springback and Material Variance

Metal is elastic. If you bend it to 90°, it springs back to 88°. The amount of springback depends on the **Yield Strength** and **Young's Modulus**.

- *Issue:* In a single batch of steel, yield strength can vary by 10-20%. A fixed program will produce inconsistent angles.

6.2 The MAC 2.0 Solution: In-Cycle Identification

Instead of using external angle measurement sensors (lasers/cameras) which are slow (requires stopping the bend to measure), MAC 2.0 uses **Internal Model Control** based on drive feedback [15].

6.2.1 Phase 1: Sensing (Elastic Region)

As the blade contacts the sheet and begins to push, the material resists. This resistance is proportional to its stiffness.

- **Data Source:** The controller monitors the **Torque Current (I_q)** of the bending axis motor (Direct Drive) or the pressure curve of the hydraulic valve [32].
- **Calculation:** It calculates the slope of the Stress-Strain curve ($\frac{d\sigma}{d\epsilon}$) in the elastic region (the first few degrees of movement).
- **Inference:** From this slope, it estimates the actual Yield Strength of the specific sheet being bent [7, 15].

6.2.2 Phase 2: Adaptation (Plastic Region)

Once the material yields, the controller uses the estimated properties to predict the Springback ($\Delta\alpha$).

- **Real-Time Offset:** The trajectory generator adds an offset to the target position *while the blade is moving*.
  - *Nominal Target:* 90°
  - *Calculated Springback:* 2.4°

    - *New Target:* 92.4°
- **Latency:** This calculation happens within milliseconds [15].

6.3 Angle Measurement System (AMS)

For extreme precision, Salvagnini also employs **AMS**, a laser-based system [34, 35].

- **Function:** Measures the angle after the bend.
- **Closed Loop:** If the angle is out of tolerance, the machine performs a "re-hit" correction.
- **Difference from MAC 2.0:** AMS is reactive (measure-then-correct); MAC 2.0 is predictive (measure-during-bend).

---

7. Safety and Ancillary Systems

A proprietary controller must strictly adhere to safety standards (ISO 13849).

7.1 Functional Safety (TwinSAFE)

The P4/P2 likely uses **FSoE (FailSafe over EtherCAT)** or **TwinSAFE** [21].

- **Architecture:** Safety logic (E-Stops, Light Curtains, Interlocks) is transmitted over the *same* EtherCAT cable as the motion data, but in a dedicated, redundant frame container (Black Channel approach).
- **Safe Motion:** The drives support STO (Safe Torque Off) and SLS (Safely Limited Speed). When an operator approaches the loading area, the machine doesn't shut down; it switches to a slow, safe speed, improving cycle restart times.

7.2 Energy Management (KERS)

The **KERS (Kinetic Energy Recovery System)** [19] requires a shared DC Bus topology on the servo drives.

- **Control Logic:** The controller must sequence axes so that braking axes dump energy into the DC bus exactly when other axes are accelerating. This requires "Energy-Aware Trajectory Planning."

---

8. Engineering Roadmap for Proprietary Controller Development

Based on the analysis of the Salvagnini architecture, the following roadmap is proposed for developing a competitive proprietary controller.

8.1 Phase 1: Hardware Selection & Abstraction

- **IPC:** Select a Quad-Core IPC (e.g., Beckhoff CX20x0 or C69xx).
- **Fieldbus:** Mandate EtherCAT. It is the only open protocol with the requisite synchronization features.
- **Drives:** Source EtherCAT drives with high-fidelity torque reporting (e.g., Kollmorgen AKD2G or similar).
- **HAL:** Write a Hardware Abstraction Layer in C++ that interfaces with the EtherCAT Master (e.g., Acontis or TwinCAT) so the high-level code is hardware-agnostic.

8.2 Phase 2: The Soft-Motion Kernel

Do not try to write a trajectory planner from scratch in C#. Use an established Real-Time Motion Library or RTOS framework.

- **Option A (Buy):** Beckhoff TwinCAT NC I.
- **Option B (Build):** Develop a C++ kinematics solver running on a Real-Time Linux (PREEMPT_RT) or IntervalZero RTX kernel.
- **Key Task:** Implement the "Rolling" kinematic solver. You must derive the equations of motion for your specific mechanical linkage.

8.3 Phase 3: The MAC Clone (Adaptive Control)

This is the most challenging phase.

1. **Data Acquisition:** Create a high-speed ring buffer logging Axis_Position and Motor_Torque at 4kHz.
2. **Modelling:** Conduct empirical tests. Bend 100 samples of known material. Record the torque curves. Correlate "Torque Slope" to "Measured Springback."
3. **Algorithm:** Implement a **Kalman Filter** to estimate the material state vector $[E, \sigma_y]$ (Modulus, Yield) in real-time.
4. **Feed-Forward:** Use this estimate to drive a feed-forward term in the position loop.

8.4 Phase 4: HMI and CAM Integration

- **HMI:** Develop in WPF (C#) or Qt (C++). Focus on the "Job List" workflow.
- **3D Visualization:** Integrate a lightweight 3D engine (e.g., Helix Toolkit or Hoops) to visualize the bend sequence from the imported CAD data.
- **Importer:** Write a parser for STEP files that extracts flange geometry and maps it to the parametric macros defined in Phase 2.

9. Conclusion

The Salvagnini P4/P2 architecture proves that the modern competitive advantage in machine tools is software-defined. The hardware is high-quality but standard (IPCs, EtherCAT, Servos).

The proprietary value lies in the **Real-Time Kinematic Solver** (simulating complex rolling motions) and the **MAC 2.0 Adaptive Observer** (treating the drive current as a sensor).

To replicate this, one must move beyond the "G-Code" mindset. The controller acts less like a playback device for pre-recorded paths and more like a robotic physicist, constantly solving equations of motion and material deformation in a 1-millisecond loop. The successful development of a proprietary controller rests on the seamless integration of a hard real-time kernel with a physics-based material model.