## **Sistema MIU: Schema Funzionale con PID e Feedforward per le Soglie Dinamiche**

Questo schema estende il precedente, integrando esplicitamente il concetto di **controllo PID con Feedforward** per la gestione dinamica delle soglie del TaxonomyOrchestrator.

L'obiettivo è mostrare come il sistema non solo genera nuove regole, ma anche **apprende a calibrare i propri parametri di apprendimento** in base all'esperienza.

+---------------------+ +---------------------+  
| Input Utente/ | | IMIUDataManager |  
| Scheduler (Tesi) |<----->| (Persistenza: |  
| - Stringa Iniziale| | MIU\_States, |  
| - Stringa Target | | RegoleMIU, |  
| | | Statistiche, |  
+----------+----------+ | Soglie Dinamiche)|  
 | (Richiesta Esplorazione) +----------+----------+  
 v ^  
+--------------------------------------------------+ (Dati Aggregati, Regole/Soglie Aggiornate)  
| EvolutiveSystem.Engine |  
| (Il Motore di Derivazione - Esecutore della Tesi)|  
| (Il "Processo" Controllato) |  
| +-------------------+ +---------------------+ |  
| | MIUDerivationEngine |<->| RegoleMIUManager | |  
| | (Orchestra Ricerca) | | (Applica Regole, | |  
| | - Carica Regole/Stats | | BFS/DFS Intelligente) | |  
| | - Avvia Esplorazione| | - Usa Euristiche | |  
| +-------------------+ | (CalculatePriority) | |  
| ^ +---------------------+ |  
| | (Eventi: RuleApplied, SolutionFound, NewMiuStringDiscovered)  
| | (Le "Variabili di Processo" - PV)  
+------------|-------------------------------------+  
 |  
 |  
 v  
+--------------------------------------------------+  
| EvolutiveSystem.TaxonomyOrchestration |  
| (Il Conduttore del Circuito di Hegel) |  
| (L' "Attuatore" delle Soglie) |  
| +-------------------+ +---------------------+ |  
| | EventBus |<->| TaxonomyOrchestrator| |  
| | (Canale Eventi) | | (Monitora Eventi, | |  
| +-------------------+ | Trigger Antitesi, | |  
| ^ | APPLICA SOGLIE DINAMICHE)|  
| | (Richiesta Analisi/Generazione) +---------------------+ |  
| v ^  
+--------------------------------------------------+ (Soglie Dinamiche Aggiornate)  
 | |  
 | (Richiesta Analisi/Rilevazione Antitesi - L' "Errore" - E)  
 v |  
+--------------------------------------------------+  
| EvolutiveSystem.Taxonomy |  
| (Il Rivelatore dell'Antitesi) |  
| (Il "Rilevatore di Errore" - E) |  
| +-------------------+ |  
| | Taxonomy | |  
| | (Analizza Dati, | |  
| | - MiuPatternStatistics (Nodi Topologici) |  
| | - Identifica Gap/Inefficienze |  
| | - IDENTIFICA PROBLEMI DI SOGLIE ATTUALI |  
| +-------------------+ |  
+-----------|--------------------------------------+  
 |  
 | (Antitesi Rilevata: Gap/Inefficienze, Problemi Soglie)  
 v  
+--------------------------------------------------+  
| Rule Generator/Learner |  
| (Il Creatore della Sintesi) |  
| (Il "Controller PID" e "Feedforward") |  
| +-------------------+ |  
| | Rule Generator | |  
| | (Formula Ipotesi | |  
| | Nuove Regole) | |  
| | - Test e Validazione |  
| | +-------------------+ |  
| | | Meta-Parameter Learner |<---------------------+ (Contesto Esterno/LLM - Il "Feedforward")  
| | | (Calcola Nuove Soglie) | |  
| | | - Logica PID (P, I, D) | |  
| | +-------------------+ |  
| +-------------------+ |  
+-----------|--------------------------------------+  
 |  
 | (Nuove Regole Validate - Nuova Tesi)  
 | (Nuove Soglie Ottimizzate - Nuova Tesi)  
 +-------------------------------------> (Torna a IMIUDataManager/RegoleMIUManager)

### **Spiegazione delle Aggiunte:**

1. **IMIUDataManager (Aggiornato):**
   * Ora include Soglie Dinamiche come elemento persistente. Questo significa che i valori delle soglie ottimizzate verranno salvati e caricati.
2. **EvolutiveSystem.Engine (Il "Processo" Controllato):**
   * Etichettato come "Il 'Processo' Controllato". Le sue metriche (eventi di RuleApplied, SolutionFound, NewMiuStringDiscovered) sono le "Variabili di Processo (PV)" che il sistema di controllo monitora.
3. **EvolutiveSystem.Taxonomy (Il "Rilevatore di Errore" - E):**
   * Ora include esplicitamente la capacità di **"IDENTIFICARE PROBLEMI DI SOGLIE ATTUALI"**. Questo significa che la logica di IdentifyGaps() e IdentifyInefficiencies() si estenderà per riconoscere non solo problemi nelle regole, ma anche se le soglie del TaxonomyOrchestrator stanno portando a un comportamento subottimale del ciclo di apprendimento (es. troppo frequente/raro, troppo sensibile/insensibile).
   * L'output di Taxonomy (Antithesis Rilevata) include ora anche i problemi relativi alle soglie. Questo è il segnale di "errore" per il nostro controller PID.
4. **EvolutiveSystem.TaxonomyOrchestration (L' "Attuatore" delle Soglie):**
   * Etichettato come "L' 'Attuatore' delle Soglie". Riceve le Soglie Dinamiche Aggiornate dal "Rule Generator/Learner" e le applica internamente per modificare il suo comportamento (es. quanto spesso "scatta" per generare l'Antitesi).
5. **Rule Generator/Learner (Il "Controller PID" e "Feedforward"):**
   * Questo blocco è stato espanso per includere il **Meta-Parameter Learner**.
   * **Meta-Parameter Learner**: Questa è la nuova sottocomponente che implementerà la logica PID.
     + **Input**: Riceverà l'Antithesis Rilevata da Taxonomy (l'errore E).
     + **Logica PID**: Calcolerà i nuovi valori delle soglie basandosi su:
       - **Proporzionale (P):** L'entità dell'errore (quanto è grave il gap/inefficienza o il problema delle soglie).
       - **Integrale (I):** La persistenza dell'errore nel tempo (se il problema non si risolve).
       - **Derivativo (D):** La velocità di cambiamento dell'errore (se il problema sta peggiorando rapidamente).
     + **Output**: Proporrà nuove Soglie Ottimizzate.
   * **Feedforward**: Una nuova freccia etichettata **"(Contesto Esterno/LLM - Il 'Feedforward')"** punta al Meta-Parameter Learner. Questo rappresenta l'input proattivo (es. previsioni, analisi di alto livello da un LLM) che può pre-regolare le soglie prima che un errore si manifesti pienamente.
   * L'output del Rule Generator/Learner ora include non solo Nuove Regole Validate ma anche Nuove Soglie Ottimizzate, che tornano a IMIUDataManager e TaxonomyOrchestrator per chiudere il ciclo di auto-ottimizzazione.

Questo schema ora cattura pienamente la tua visione di un sistema MIU che non solo evolve le sue capacità, ma anche i meccanismi che governano la sua stessa evoluzione, usando i principi del controllo PID e feedforward.