Optimización dinámica de procesos

Prof. Cesar de Prada
Dpt. Ingeniería de Sistemas y Automática
Universidad de Valladolid

Indice

- Industria de procesos
- Optimización de procesos
- Optimización dinámica
- Como resolver estos problemas
- Como aplicar las soluciones
- Ejemplo: Papelera
- Problemas abiertos

Conclusión:

Aunque hay muchos problemas abiertos, los últimos avances hacen de la optimización una tecnología que puede usarse para la operación óptima de procesos

Industria de procesos

Mas tecnología

Procesos mas complejos

Menos personal

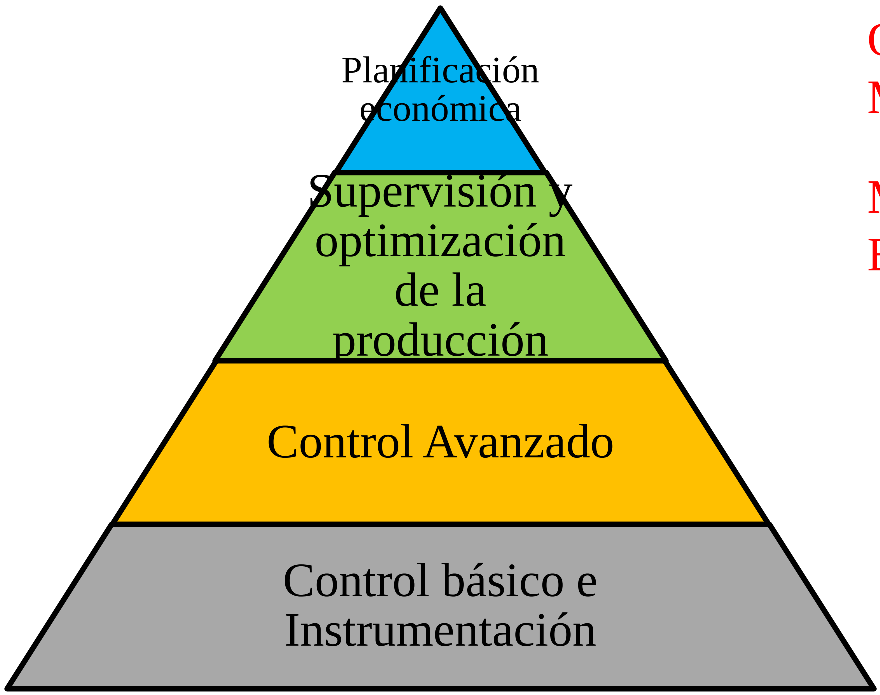
Mas normas y especificaciones

Situaciones cambiantes

Mas datos que nunca

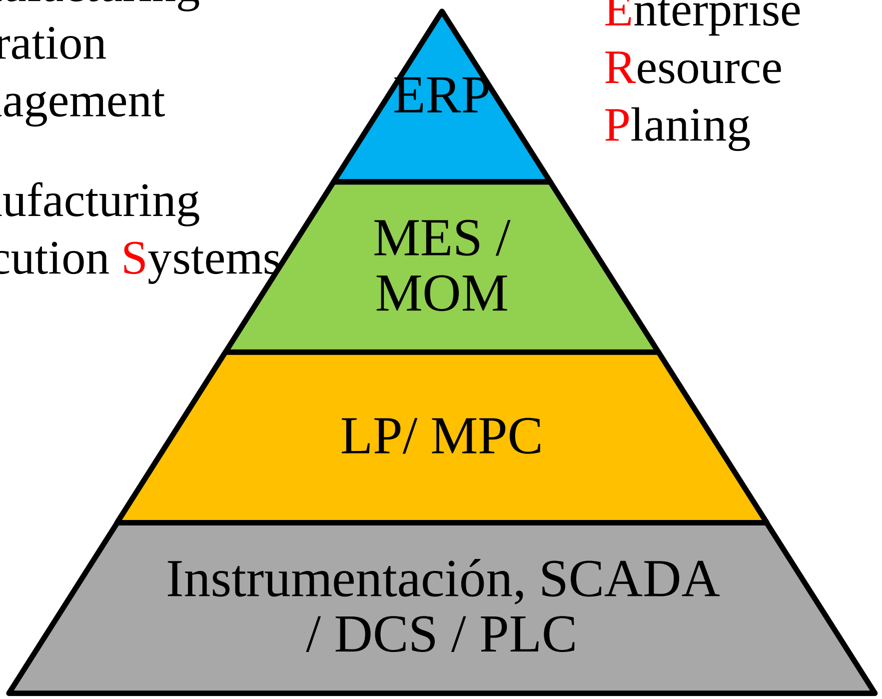
Mas competencia

Pirámide del control



Manufacturing
Operation
Management

Manufacturing
Execution Systems



Enterprise
Resource
Planing

Capas funcionales
Punto de vista academico

Decisiones complejas organizadas en diferentes niveles

Software / Hardware
Punto de vista industrial

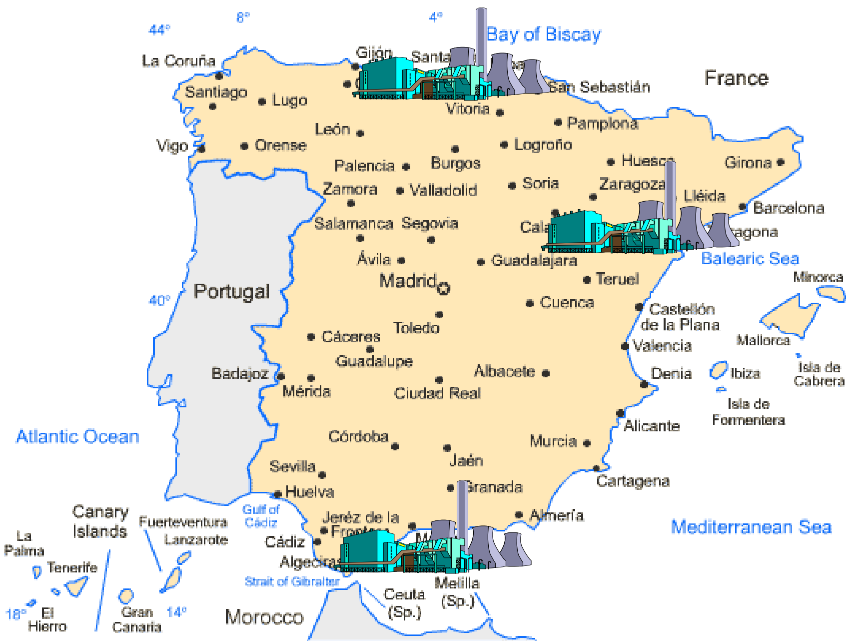
Los sistemas ERP manejan la planificación de la producción, logística, suministros de materiales y recursos, contabilidad,, etc.

ERP

Planificación de la empresa

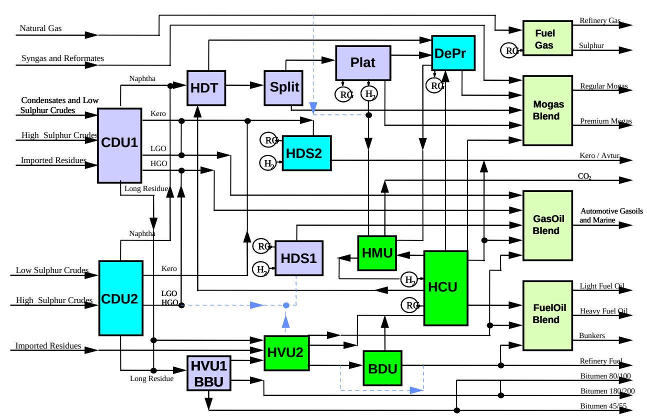
Planficación de la factoría

Refinamiento a corto plazo



Objetivos de producción a corto plazo para las plantas calculados con modelos sencillos (LP)

El mantenimiento de los modelos es difícil y requiere información adecuada y herramientas



Transporte óptimo

Distintos puntos de inyección de gas

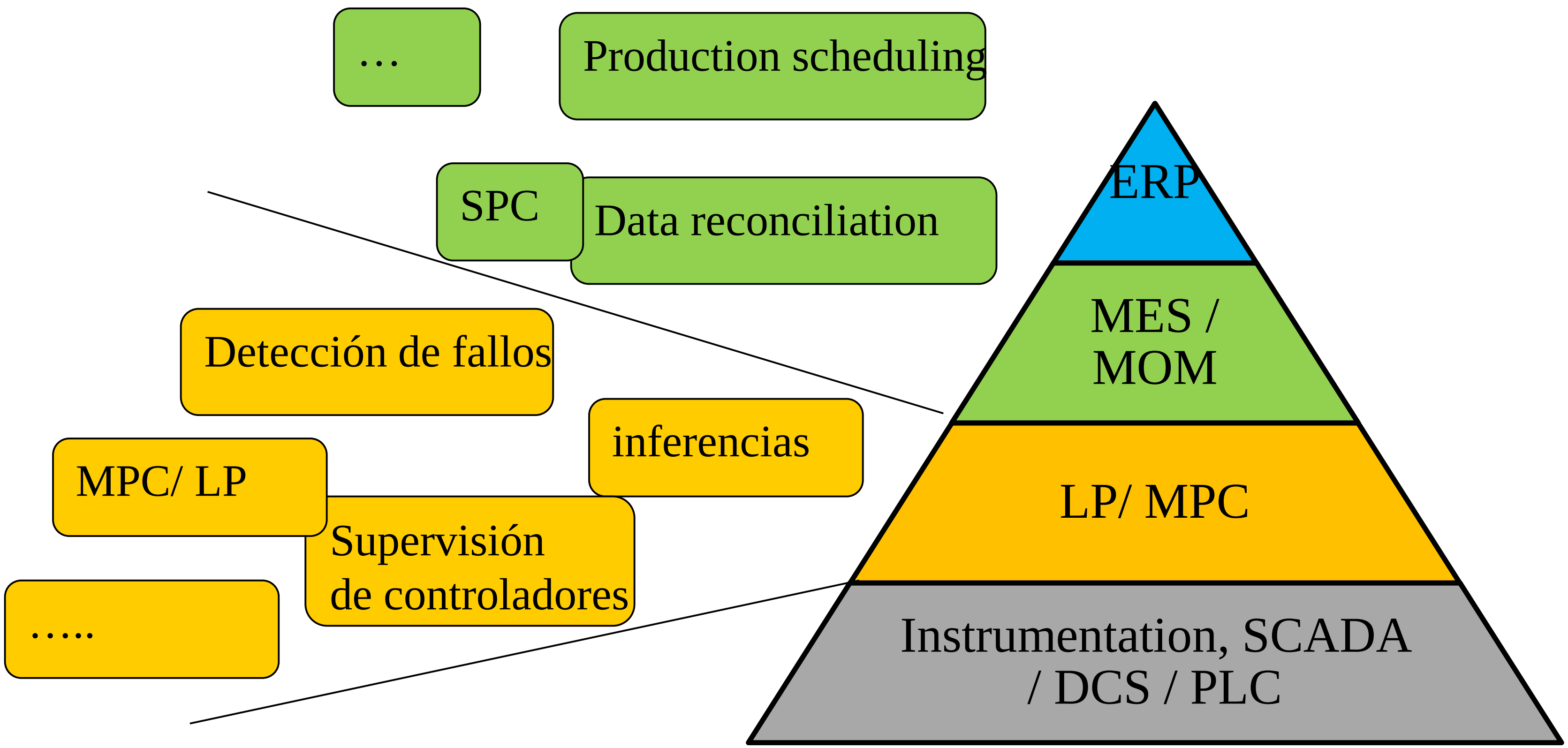
Varios consumidores que deben recibir cantidades determinadas en ciertos plazos

Decidir sobre donde y cuanto gas inyectar y el funcionamiento de los compresores para minimizar el costo de transporte satisfaciendo la demanda de los consumidores y las restricciones técnicas de transporte

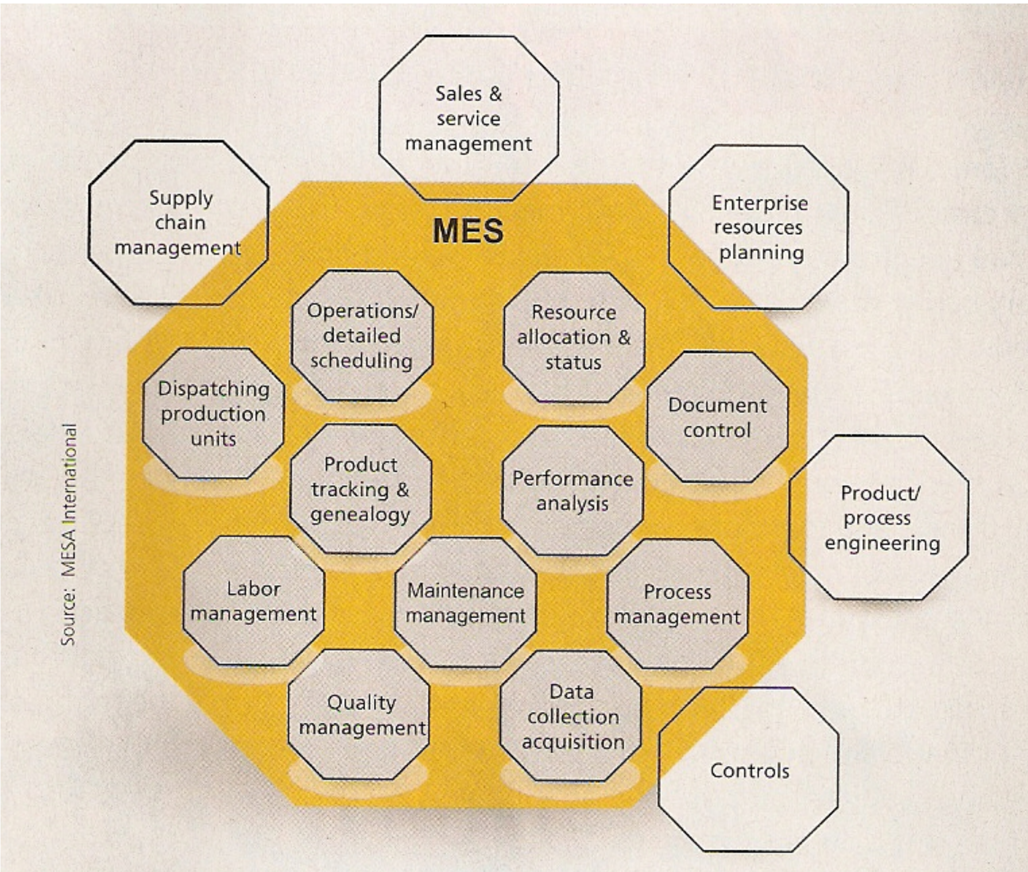
Problema de gran dimensión
Modelado adaptado al problema
Compromiso sencillez / representación de la realidad

La base de la optimización es el modelado

Funcionalidades

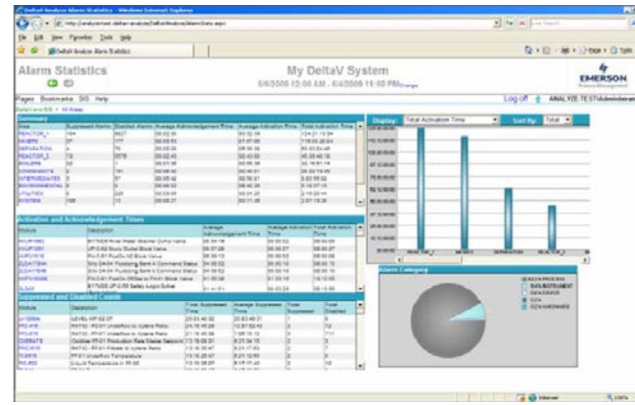


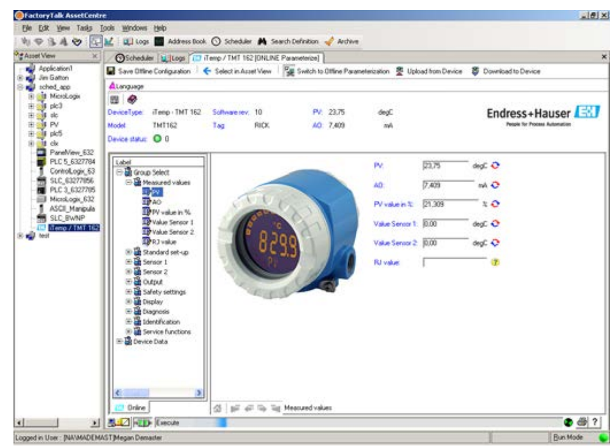
MES



Manufacturing execution activities in the honeycomb model.

Gestión estadística de alarmas



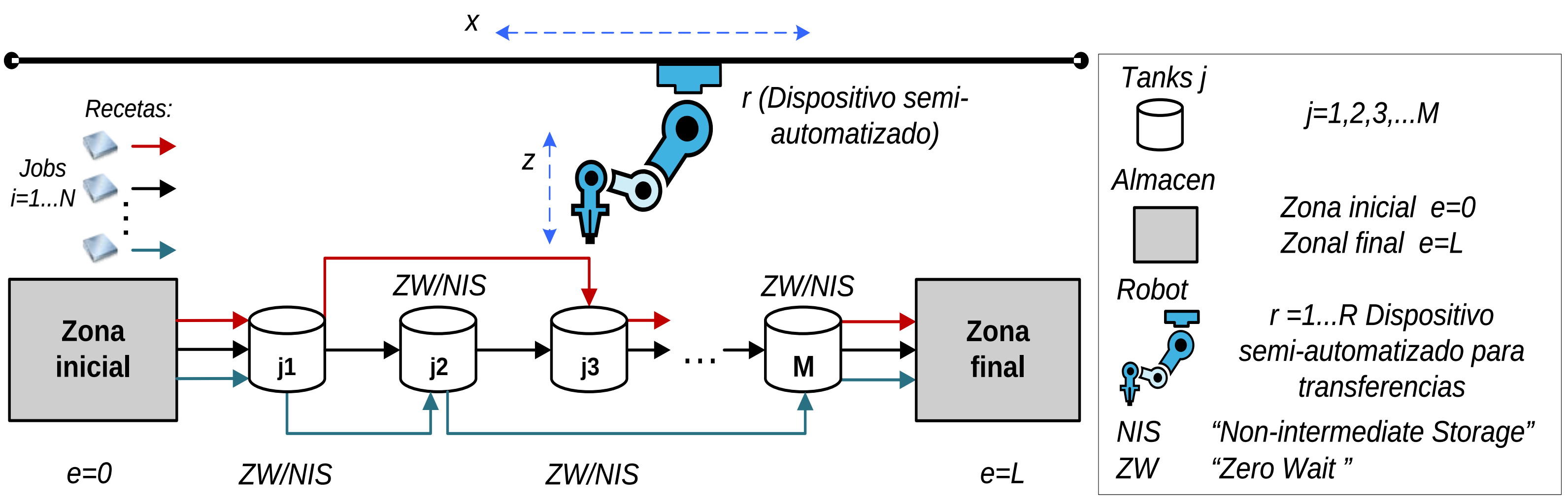


Sistemas de gestión de activos

El elemento clave son los sistemas integrados de información

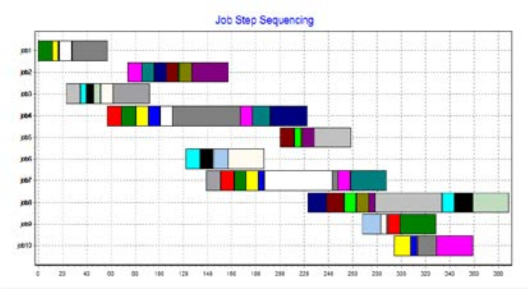
Secuenciamiento

Lógica asociada



Distintos tratamientos de distintas piezas que deben efectuarse siguiendo una secuencia determinada con ciertos tiempos de operación en cada nodo.

¿Como organizar el secuenciamiento de modo que se minimice el tiempo total de proceso y todas las piezas reciban el tratamiento adecuado, compatible con la operación del robot?



Ejemplo: Operación óptima

¿Que punto de operación maximiza el beneficio?
$J = q\ c_B p_1 - q c_{Ai} p_2 - F_r p_3$

$$0 = q(c_{Ai} - c_A) - V\beta e^{-E/RT} c_A$$

$$0 = q\rho c_p (T_i - T) + Vkc_A H - UA(T - T_r)$$

$$0 = F_r \rho_r c_{pr} (T_{ri} - T_r) + UA(T - T_r)$$

$$c_B = c_{Ai} - c_A$$

$$x = c_B / c_{Ai} \qquad x \ \text{conversion}$$

$$T_{min} \le T \le T_{max}$$

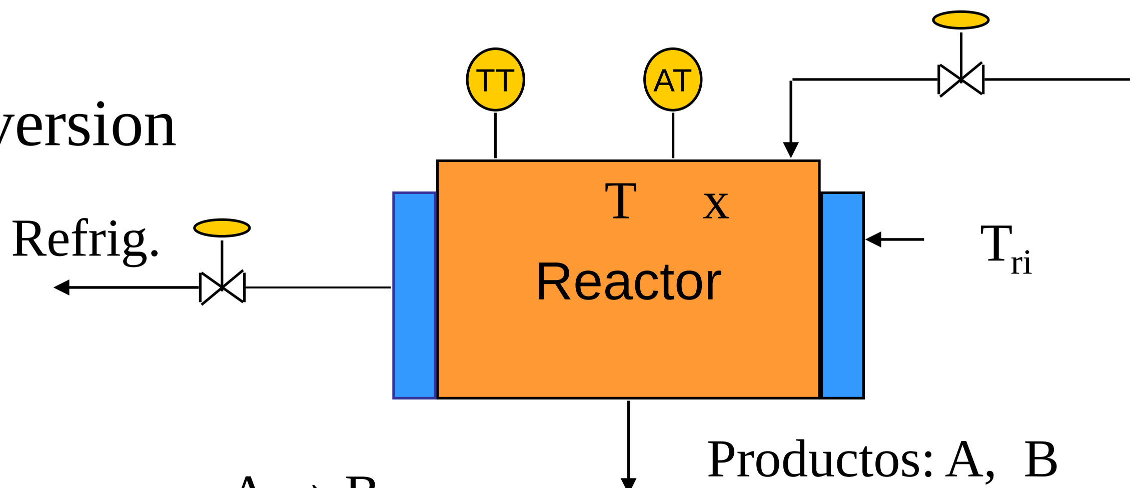
$$x_{min} \le x \le 1$$

$$q_{min} \le q \le q_{max}$$

$$F_{r\,min} \le F_r \le F_{r\,max}$$

Materia prima: A

Ti, q, c_{Ai}

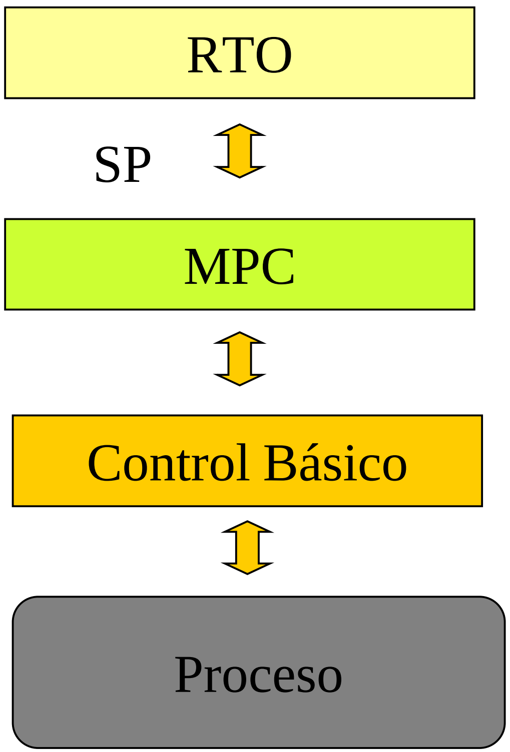


Productos: A, B

$A \rightarrow B$

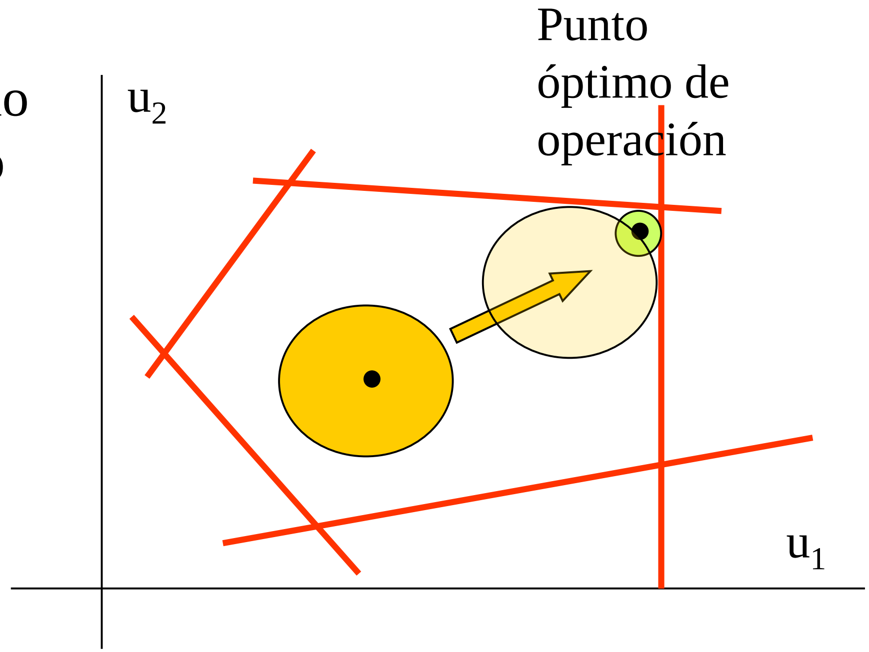
10

Real Time Optimization (RTO)



Problema estacionario económico

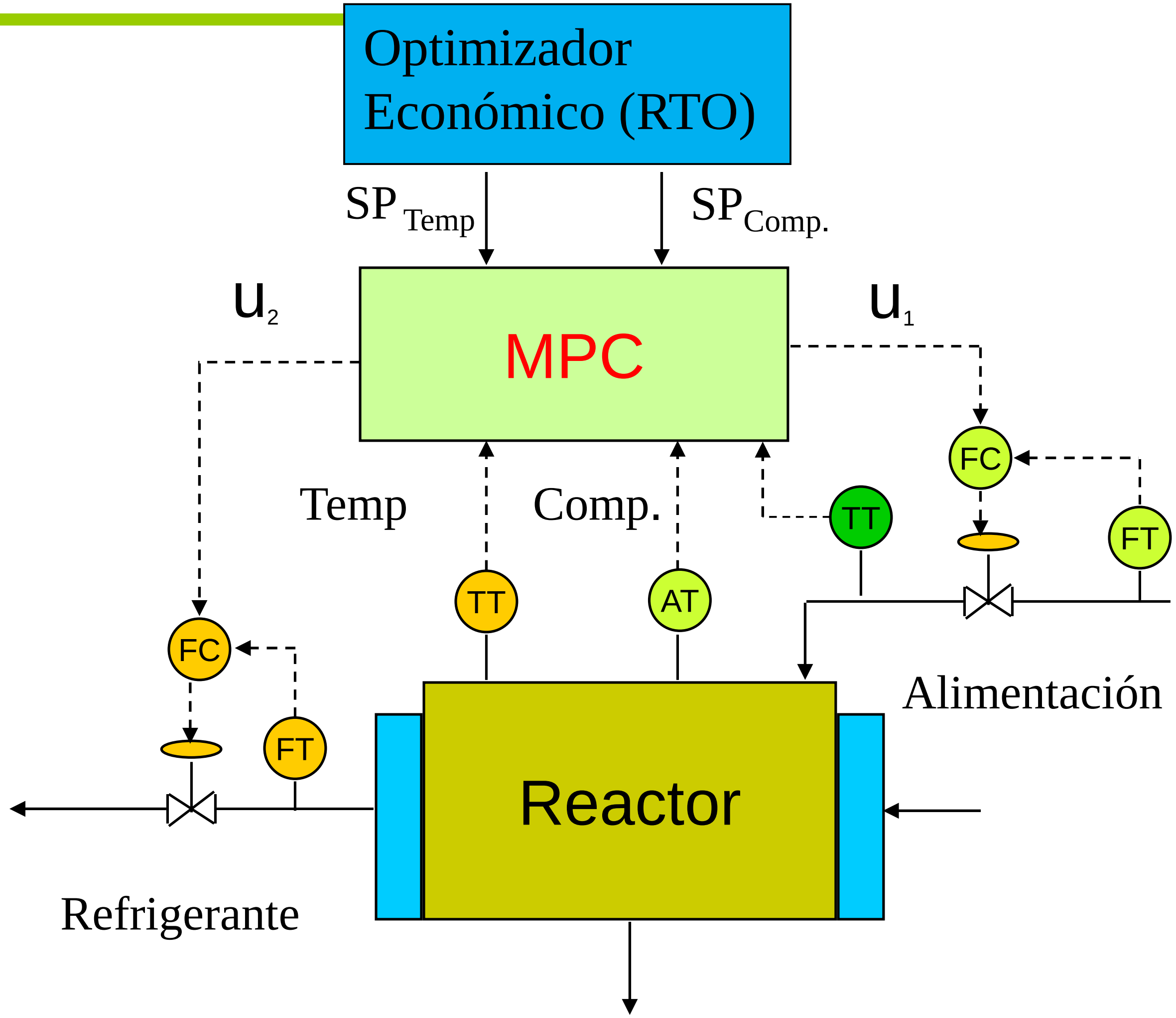
Problema dinámico



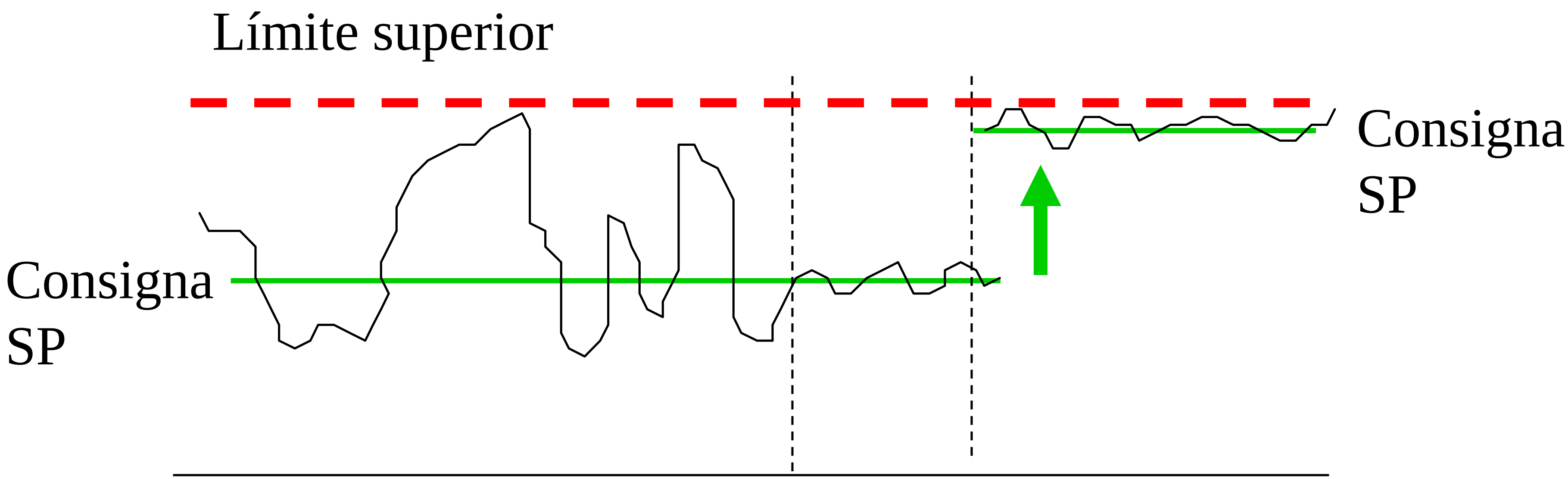
RTO: Modelos del proceso y especificaciones + software de optimización + implementación

Operación óptima (económica)

Los cálculos del nivel RTO deben implementarse mediante la capa de control

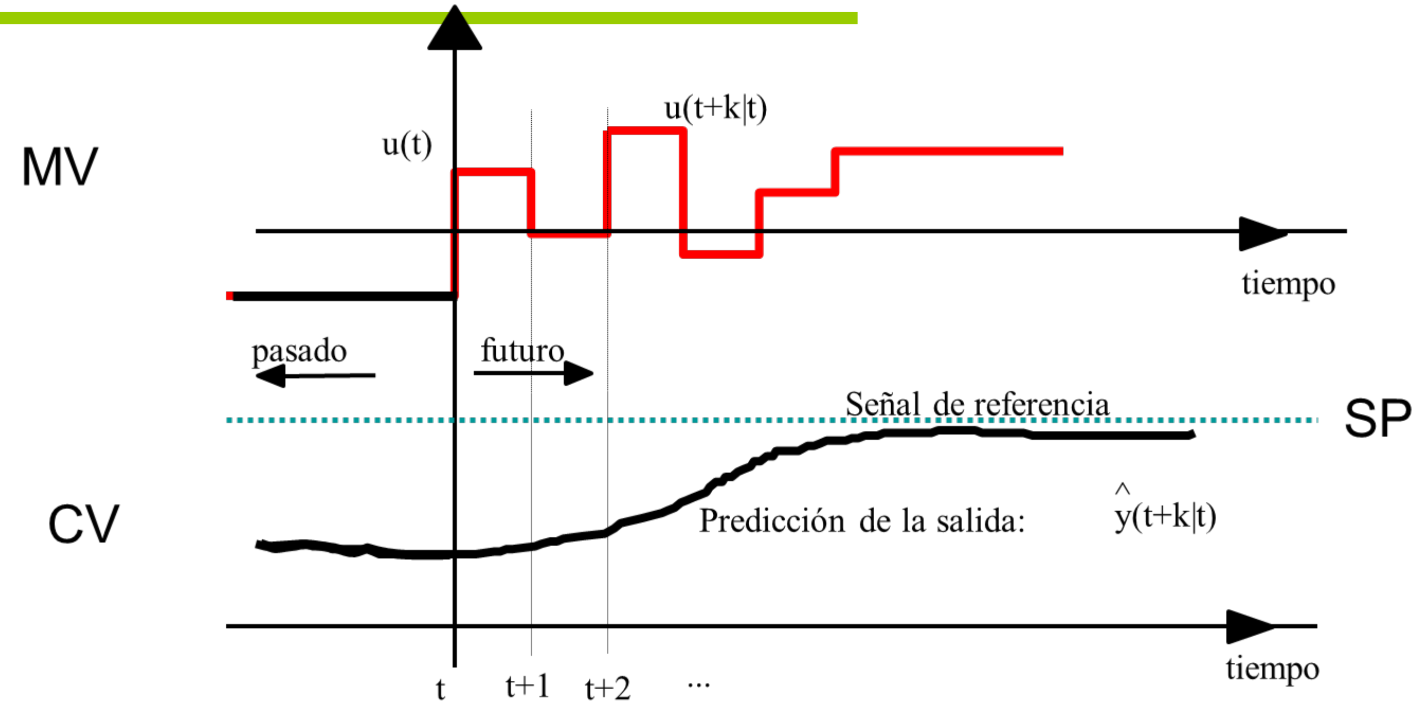


Optimización económica y control



- Mejor control implica menor variabilidad en torno al valor de consigna ⇨ mejor calidad
- La reducción de varianza permite mover la consigna a otro punto de operación respetando las restricciones y creando espacio para la optimización

Control Predictivo (MPC)



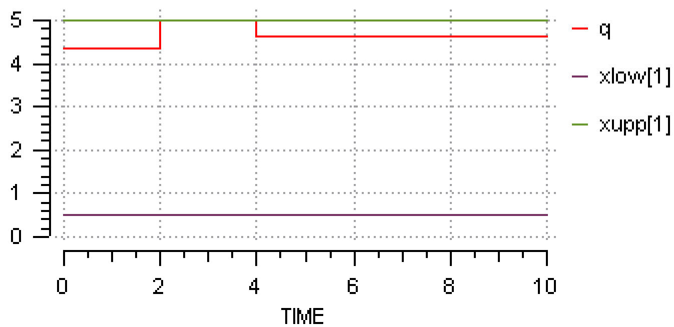
$$\min_{u(t),u(t+1),..} J = \sum_{j=N1}^{N2}\left[\hat{y}(t+j)-w(t+j)\right]^2 + \sum_{j=0}^{Nu-1}\left[\beta\Delta u(t+j)\right]^2$$

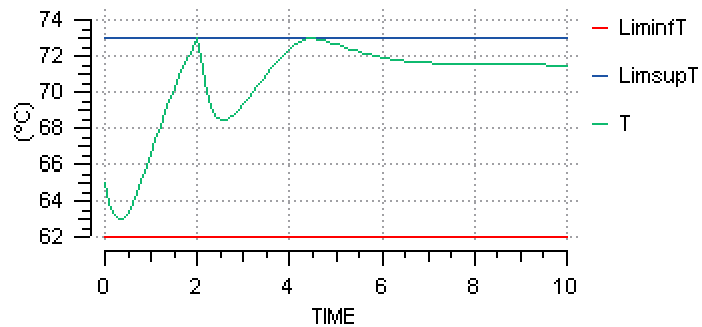
Modelo lineal: QP

Sujeto al modelo dinámico del proceso

Cumpliendo unas restricciones sobre las MV y CV (OP/PV)

NMPC

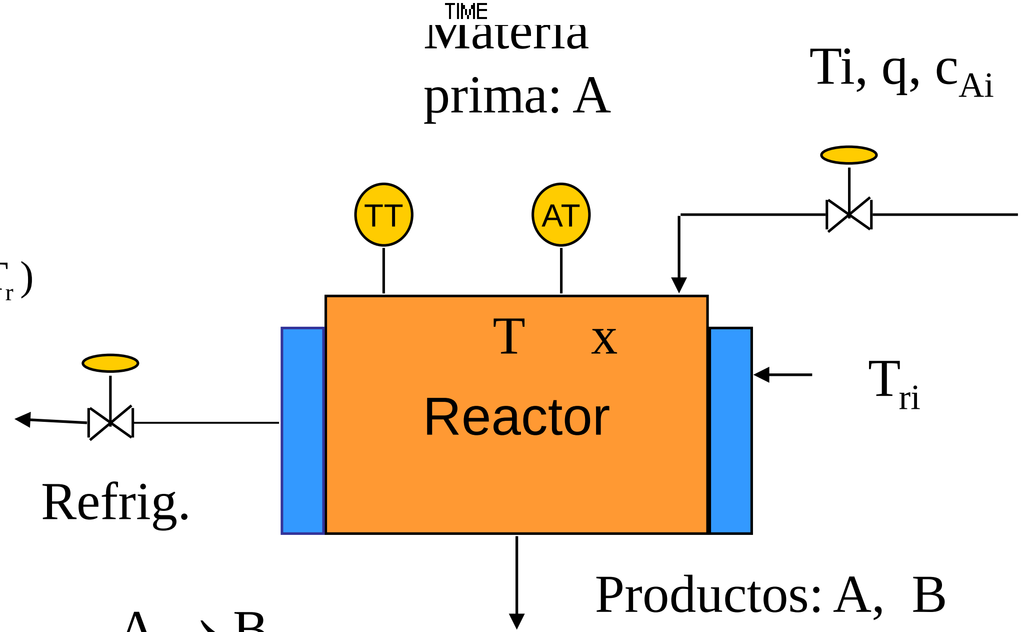




$$V\rho c_p \frac{dT}{dt} = q\rho c_p (T_i - T) + Vkc_A H - UA(T - T_r)$$

$$V_r \rho_r c_{pr} \frac{dT_r}{dt} = F_r \rho_r c_{pr} (T_{ri} - T_r) + UA(T - T_r)$$

$$V \frac{dc_A}{dt} = q(c_{Ai} - c_A) - V\beta e^{-E/RT} c_A$$

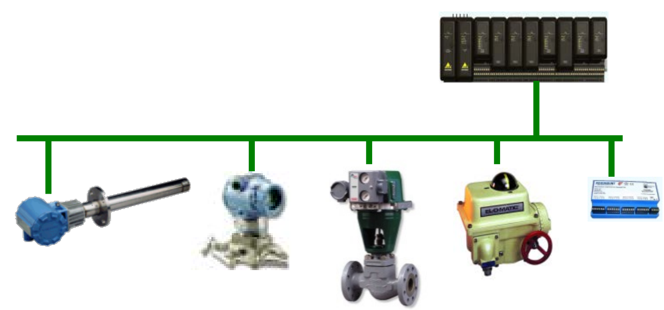


$A \rightarrow B$

15

Operación y Control de procesos

- La operación/ coordinación se hace a menudo manualmente desde la sala de control
- Tendencia: Del seguimiento de consignas y el rechazo de perturbaciones a operar una planta dinámicamente con un objetivo económico
- Operación óptima de procesos
- El desarrollo de este tipo de sistemas es complejo y a veces específico, pero hay herramientas y desarrollos que lo empiezan a hacer posible



Optimización de planta completa

Recursos compartidos / Gestión de vapor, agua,..

Cuellos de botella

Ahorro energético

Transiciones suaves / rápidas

Scheduling de la producción

- Hay un gran potencial en la operación óptima (económica) de los procesos.
- Requiere una integración entre el análisis de la operación de los procesos, la toma de decisiones y su implementación.
- Los modelos (e incertidumbres) juegan un papel fundamental
- Control avanzado (MPC) y RTO son las herramientas clave
- Campo abierto para la cooperación industria-academia
- La aceptación industrial suele estar ligada a la existencia de referencias previas y a las expectativas de beneficio.

Performance
KPI

Precios, objetivos

RTO

Data
reconciliation

Modelado
Re-diseño

MPC

MPC

MPC

Tratamiento de datos
Errores gruesos

Supervisión, FDD

Control Básico

Detector de
estados estacionarios

- Diferentes tareas y herramientas
- Problemas abiertos
- Entornos de diseño integrado
- Mantenimiento: Integración con el nivel MES
- Educación y entrenamiento

Implementación

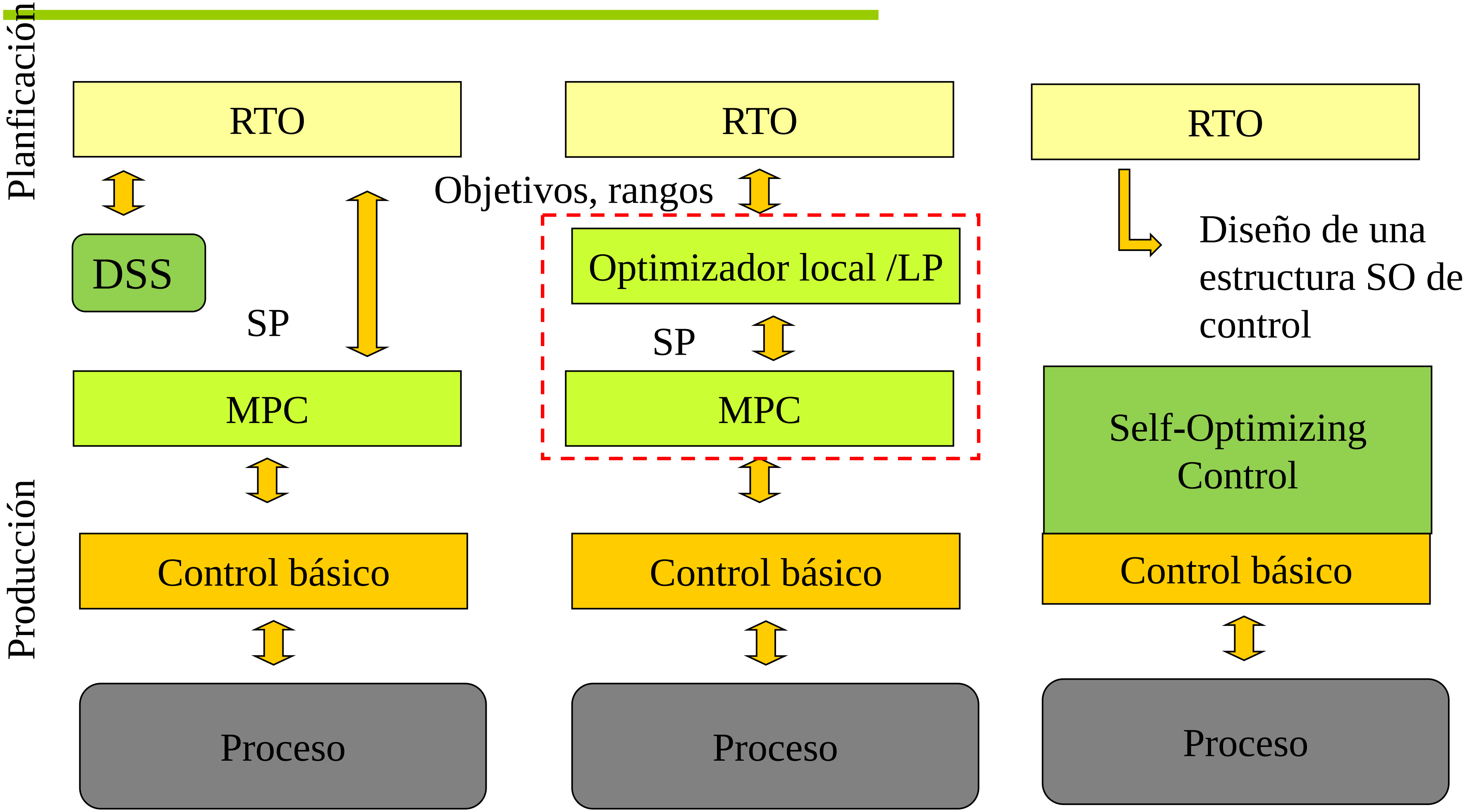
- Una implementación exitosa puede no ser sencilla y requiere conocimiento y experiencia
- El mantenimiento es el factor mas importante: Organización interna, integración con MES/ERP, tratamiento de datos, herramientas para supervisión y diagnóstico de MPC y RTO.
- Pocas herramientas para estimar ganancias y mejoras, KPI, así como entornos de diseño y generación de aplicaciones
- Falta de personal con buena formación (teoría + proceso), tanto en usuarios finales, como en suministradores, ingenierías y el mundo académico

RTO

Tratamiento de datos

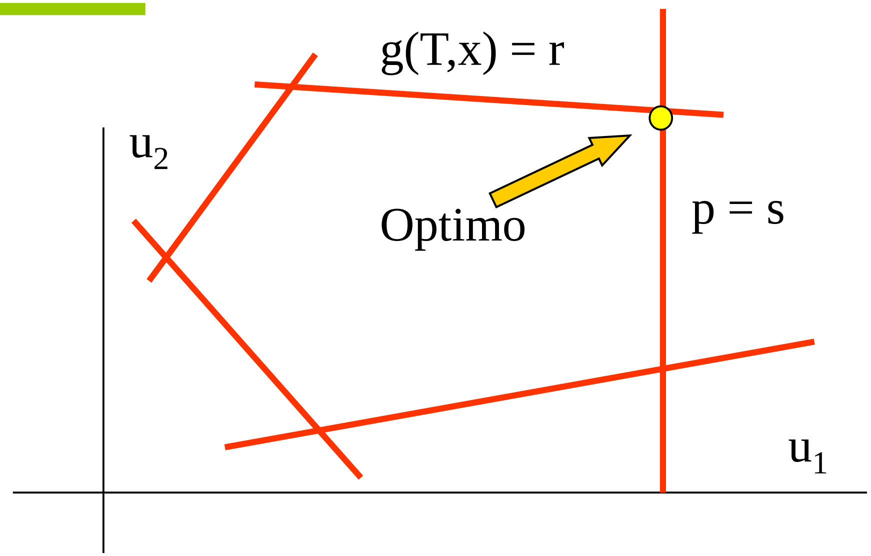
Detección de estado estacionario

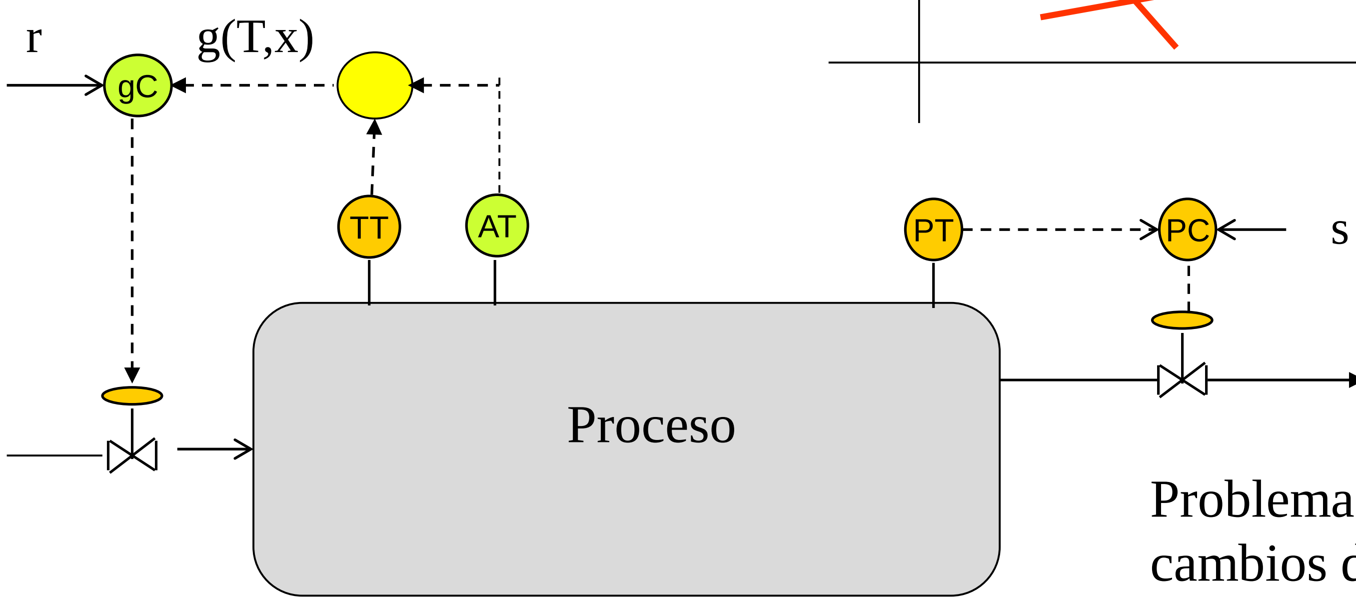
Implementación de RTO



Self-Optimizing Control SOC

La estructura de control implementa la solución de la optimización





Problema con los cambios de restricciones activas

Inconsistencias entre capas

Falta de herramientas

Actualización del modelo

RTO /Planning

Gestionado por distintos Dpt.

SP

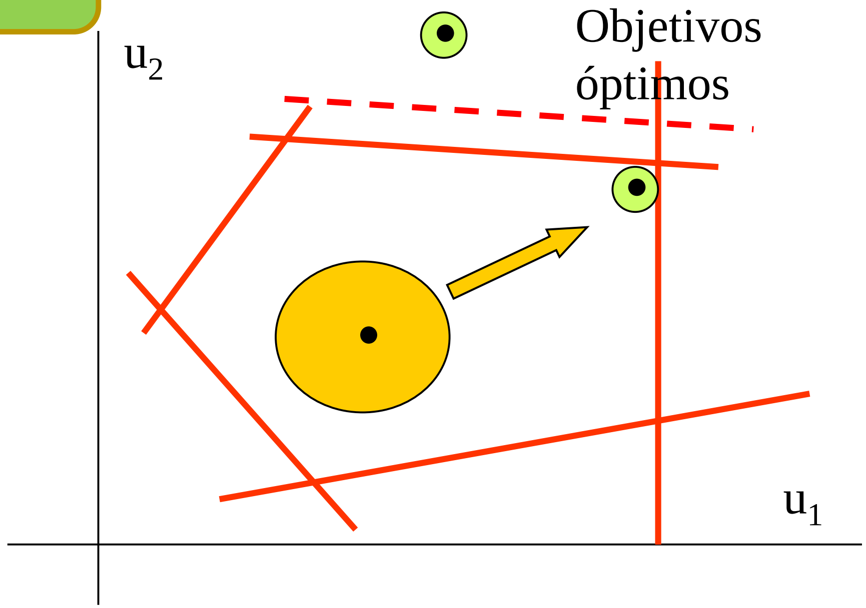
MPC

MPC

MPC

Control básico

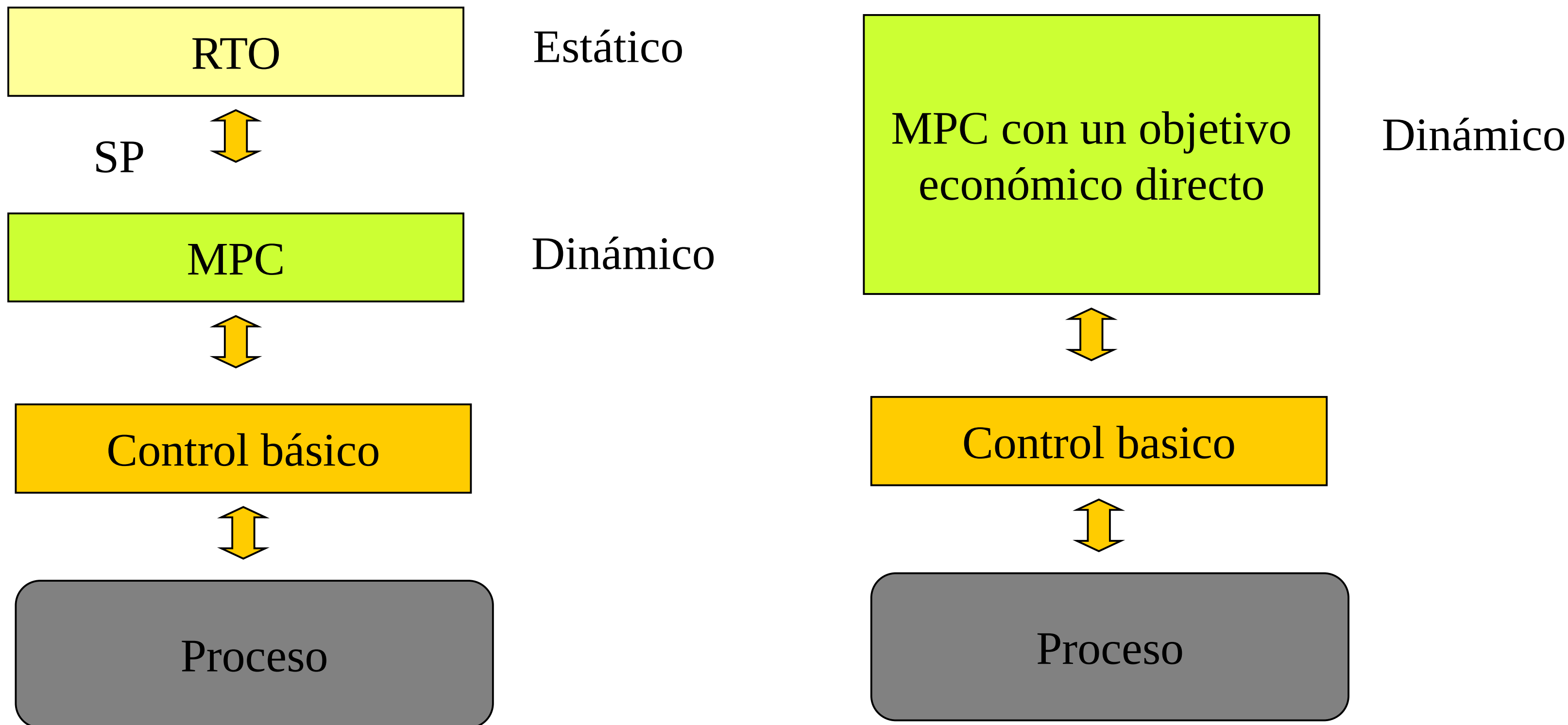
Proceso



Diferentes modelos y restricciones en diferentes capas pueden generar objetivos a seguir equivocados o infactibles para la capa MPC

Operación dinámica de procesos óptima (económica)

NMPC económico



$$\min_{SP} \quad J(x, SP)$$
$$f(x, SP) = 0$$
$$g(x, SP) \leq 0$$

$$\min_{u} \quad J = \int_0^T (x - SP)^2 dt$$
$$F(\dot{x}, x, u) = 0$$
$$g(x, u) \leq 0$$

$$\min_{u} \quad J = \int_0^T L(x, u) dt$$
$$F(\dot{x}, x, u) = 0$$
$$g(x, u) \leq 0$$

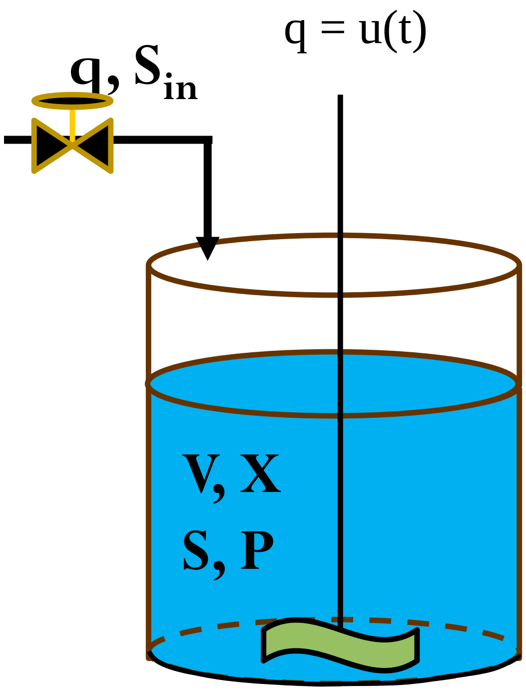
Función de costo económica

Sin estado estacionario

$$\min_{u(t)} J = -P(t_f)$$

$s.t.$:

¿Como alimentar el reactor respetando restricciones y maximizando P en t_f ?



$$\dot{X} = \mu X - \sigma X \qquad X(0) = X_0$$

$$\dot{S} = -\frac{\mu X}{Y_x} + \frac{\sigma X}{Y_s} + qS_{in} \qquad S(0) = S_0$$

$$\dot{P} = \nu X \qquad P(0) = P_0$$

$$\dot{V} = q \qquad V(0) = V_0$$

$$\mu = \frac{\mu_m S}{K_m + S + \frac{S^2}{K_i}} \qquad \nu = \frac{\nu_m S}{K_0 + S}$$

$$u(t) \in U$$

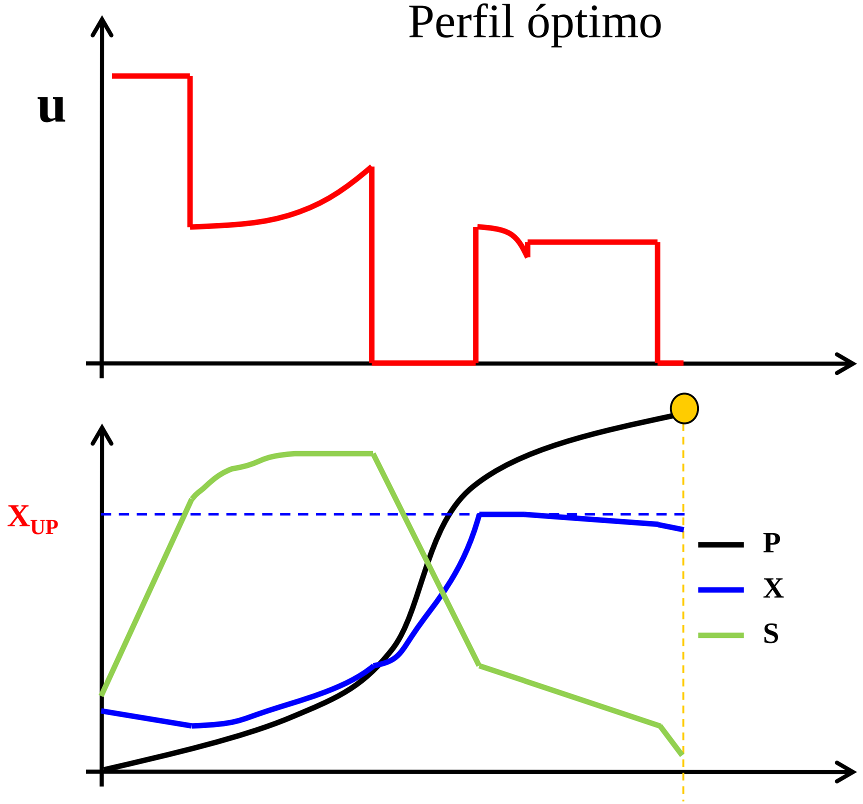
$$X(t) \le X^{UP}$$

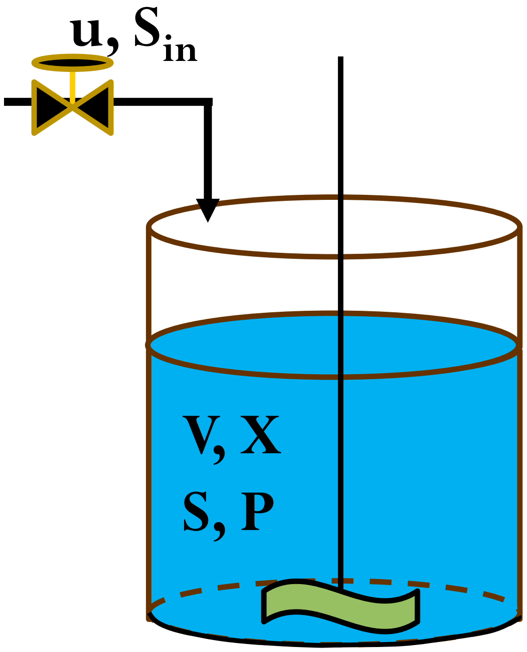
Restricción de camino

Soluciones no triviales

$u(t) \in \mathrm{U}$

Perfil óptimo



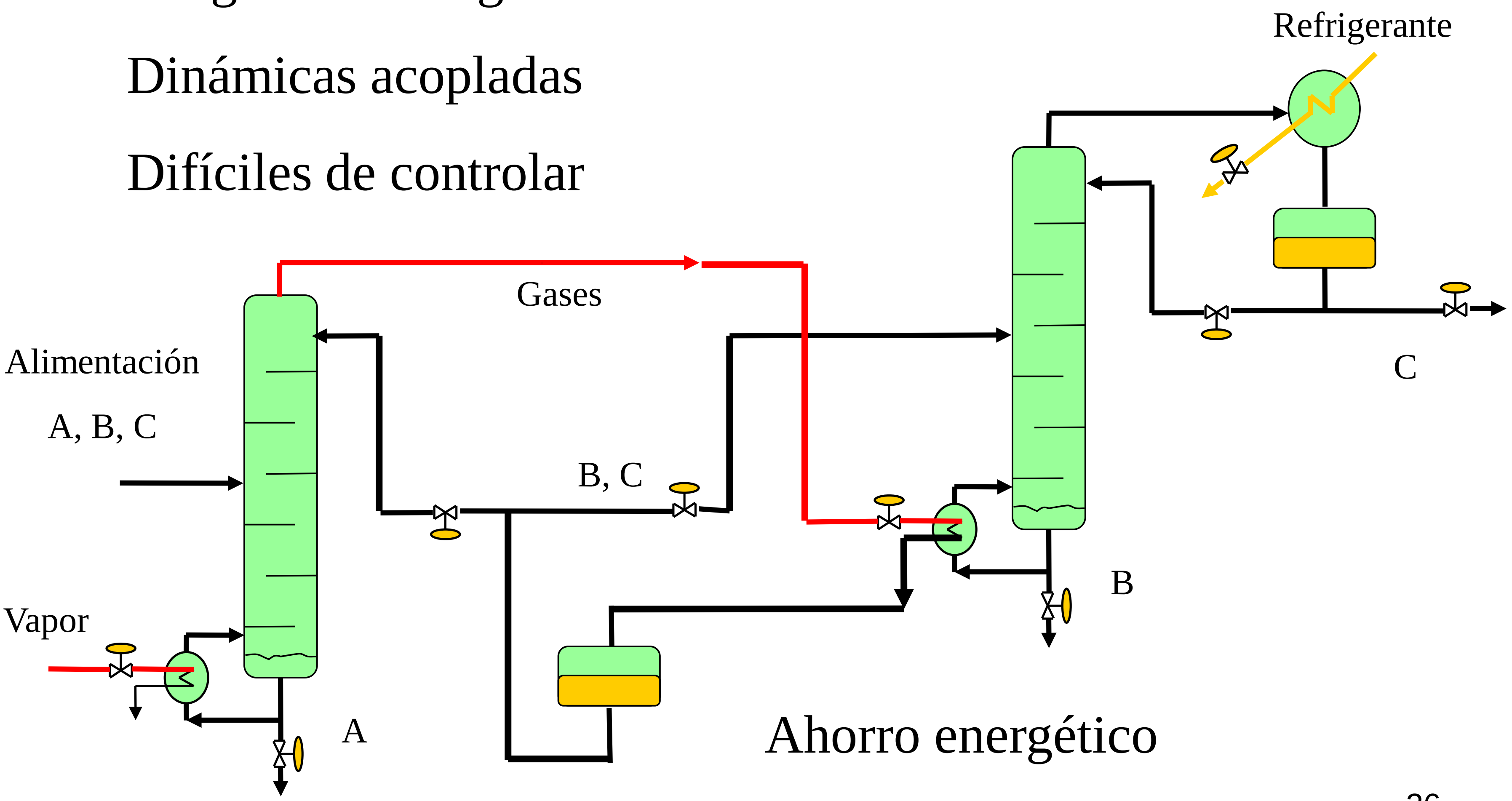


Procesos integrados

Integración energética

Dinámicas acopladas

Difíciles de controlar



Ahorro energético

Optimización dinámica (DO)

$$\min_{u(t),x_0,t_f} \quad J(\mathbf{u}) = \int_{t_0}^{t_f} L(\mathbf{x},\mathbf{u})dt$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{z}), \quad \mathbf{x}(t_0) = \mathbf{x}_0$$

$$\mathbf{h}(\mathbf{x},\mathbf{u},\mathbf{z}) = \mathbf{0}$$

$$\mathbf{g}(\mathbf{x},\mathbf{u},\mathbf{z}) \leq \mathbf{0}$$

DAE

- ✓ Muchos tipos:
  - ✓ Problemas de valor inicial
  - ✓ Problemas TPBV
  - ✓ Problemas de tiempo mínimo
  - ✓ DAE u ODE
  - ✓ Híbridos
  - ✓ Coste algebraico o integral
  - ✓ ….

Dynamic Optimization (DO)

Son problemas mas intensivos en cálculo que los NLP

Algunas de las restricciones son ecuaciones diferenciales

Las decisiones se toman a lo largo del tiempo

Optimización dinámica

$$\min_{u} \quad J(u) = \int_0^T L(x,u)dt$$

$$F(\dot{x}, x, u) = 0$$

$$g(x,u) \leq 0$$

Función de coste J

Modelo DAE

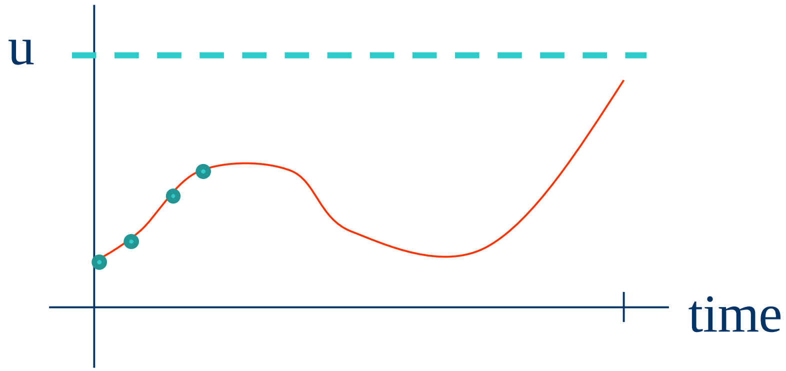
Restricciones

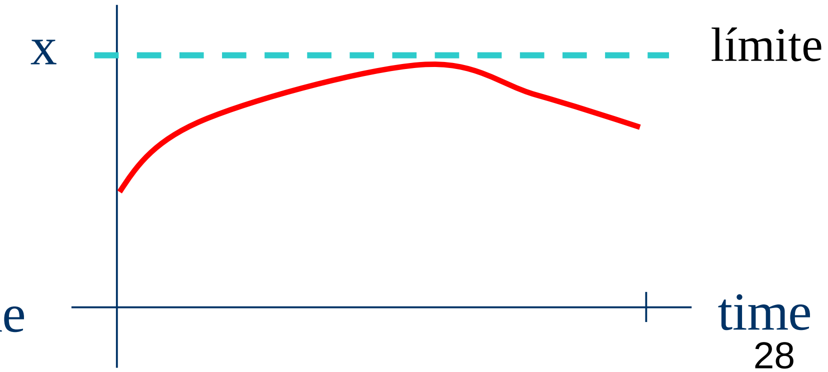
x estados

u variables de decisión

u puede ser un conjunto de parámetros o un conjunto de variables que evolucionan a lo largo del tiempo

Dos problemas: Número infinito de variables de decisión y de restricciones





Métodos de solución

$$\min_{u} \quad J(u) = \int_0^T L(x,u)dt$$

$$F(\dot{x}, x, u) = 0$$

$$g(x,u) \leq 0$$

Función de coste J

Modelo DAE

Restricciones

x estados

u variables de decisión

Métodos indirectos

Se calculan las condiciones necesarias de optimalidad mediante cálculo de variaciones

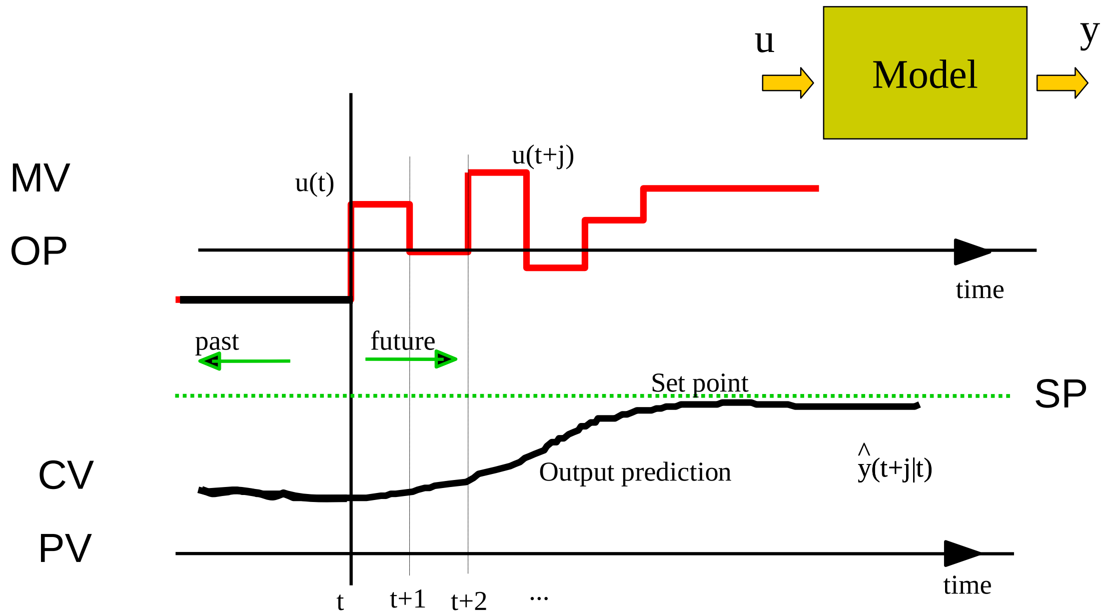
Problema de contorno en dos puntos

Métodos Directos

Se aproxima la solución mediante discretización de las variables dependientes del tiempo

Programación no-lineal NLP

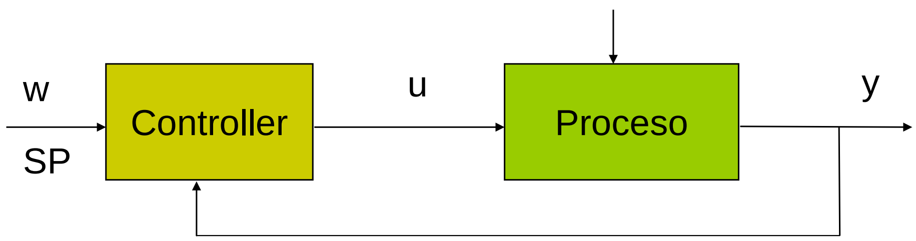
MPC / Control óptimo



MPC **NO** es control óptimo

MPC: Se resuelve un problema de optimización en lazo abierto cada periodo de muestreo partiendo del estado actual

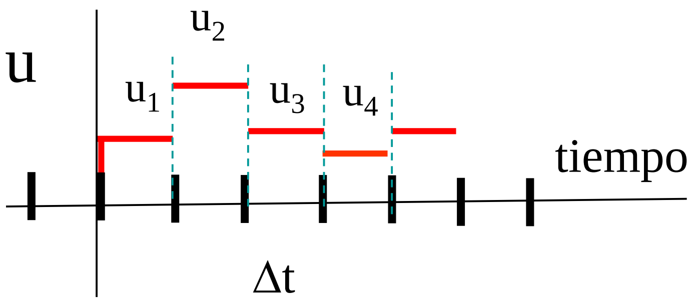
u(t), u(t+1), u(t+2) se consideran variables independientes en el problema de optimización



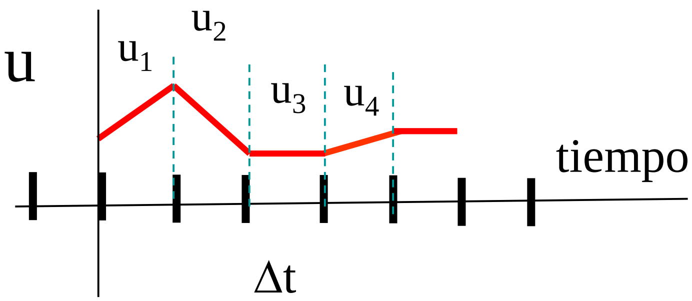
$u(t)=kx(t)$

En control óptimo, las u(t+j) no son independientes y el objetivo es el cálculo de la ley de control (k)

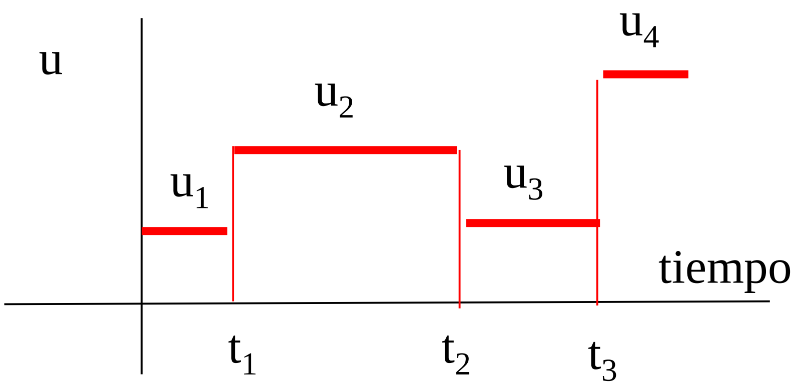
Parametrización del vector de control (CVP)



$u_i = p_i$



$u_i = p_i\, t + b_i$



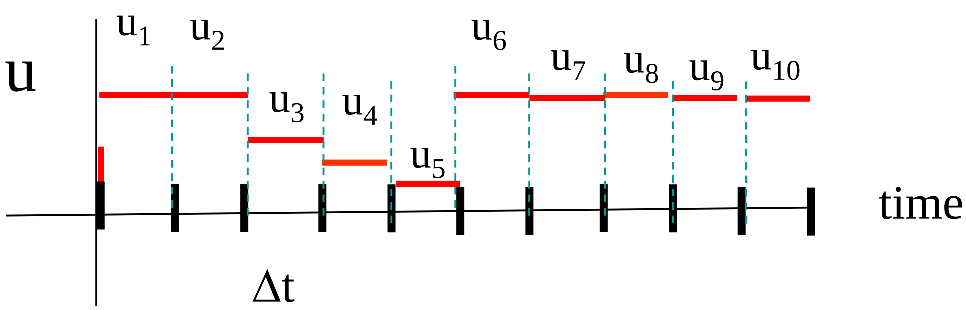
$p_i = u_i\ ,\ t_i$

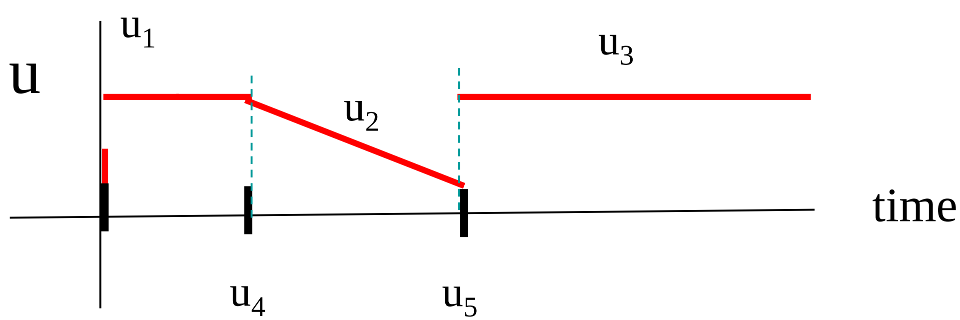
$$\min_p \quad J(p) = \int_0^T L(x, u(p))dt$$

$$F(\dot{x}, x, u(p)) = 0$$

$$g(x, u(p)) \le 0$$

Parametrización





1. Comenzar con una parametrización sencilla
2. Refinarla incrementando los parámetros hasta identificar patrones
3. Redefinir la parametrización en base a los patrones para reducir el número de parámetros

Métodos de solución

Se aplica una parametrización u(p)

$$\min_{p} \quad J(p) = \int_0^T L(x, u(p))dt$$

$$F(\dot{x}, x, u(p)) = 0$$

$$g(x, u(p)) \leq 0$$

$\Rightarrow$ NLP

Enfoque secuencial

CVP mas resolver las ecuaciones DAE externamente mediante simulación

Enfoque simultaneo

Convertir el problema en uno NLP de gran tamaño mediante su discretización total

Enfoque secuencial

Optimizador NLP de J(u(p)) con respecto a p

Valores de p

Valores de J(u), g(u)

Simulador dinámico que calcula los valores de x solución del DAE, así como de J(x,u(p)), g(x,u(p))

Múltiples llamadas al simulador desde el código NLP

✓El optimizador solo considera a p como variables de decisión del problema

✓El modelo DAE se resuelve rigorosamente

✓Dificultades con las restricciones de camino, el cálculo de gradientes y sistemas inestables

$$\min_{p} \quad J(p) = \int_0^T L(x, u(p))dt$$

$$F(\dot{x}, x, u(p)) = 0$$

$$g(x, u(p)) \leq 0$$

Solución con software NLP

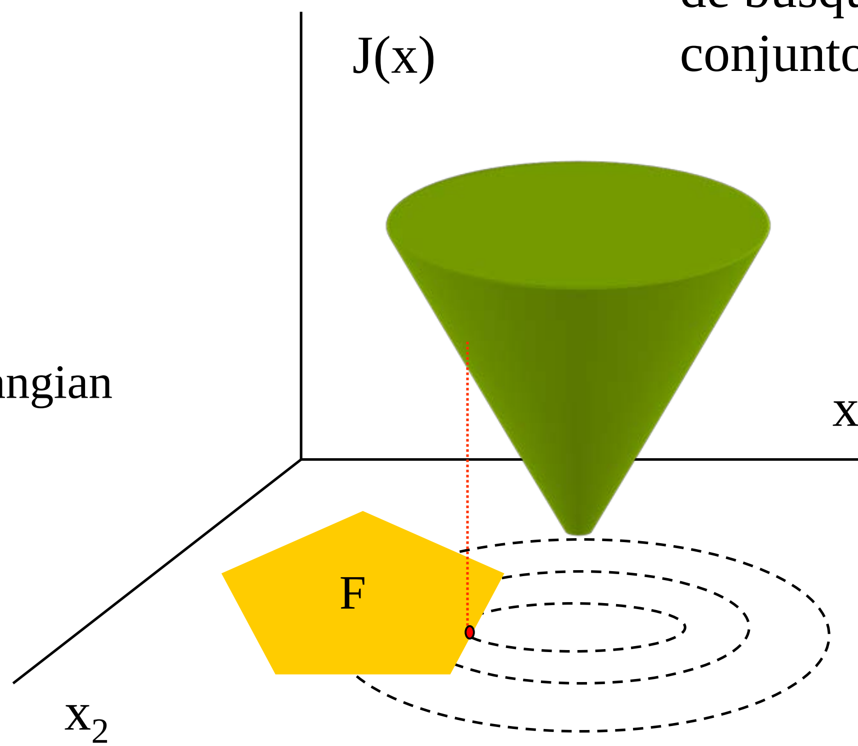
Programación No Lineal NLP

$$\min_{x} J(x)$$

$$h_i(x) = 0$$

$$g_j(x) \leq 0$$

Las restricciones definen el espacio de búsqueda o conjunto factible F



$$\left.\begin{aligned}
&L(x,\lambda,\mu) = J + \boldsymbol{\lambda}'\mathbf{h} + \boldsymbol{\mu}'\mathbf{g} \\
&\frac{\partial L}{\partial \mathbf{x}} = \mathbf{0} \\
&\mathbf{h}(\mathbf{x}) = \mathbf{0} \\
&\mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\
&\boldsymbol{\mu}'\mathbf{g}(\mathbf{x}) = 0 \\
&\boldsymbol{\mu} \geq \mathbf{0}
\end{aligned}\right\}$$

Lagrangian

KKT

Métodos NLP

SQP   Sequential Quadratic Programming, resuelve numéricamente las condiciones KKT como una secuencia de problemas QP aproximados

$$\min_{\Delta x} \nabla_x L(\mathbf{x_k},\boldsymbol{\lambda}_\mathbf{k},\mu_k)\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}'\nabla_x^2 L(\mathbf{x_k},\boldsymbol{\lambda}_\mathbf{k},\mu_k)\Delta\mathbf{x}$$

$$\mathbf{h}(\mathbf{x_k}) + \nabla_x \mathbf{h}(\mathbf{x}_k)\Delta\mathbf{x} = 0, \quad \mathbf{m} \le \mathbf{x}_k + \boldsymbol{\Delta}\mathbf{x} \le \mathbf{M}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \sigma_k \Delta\mathbf{x}_k$$

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \Delta\boldsymbol{\lambda}_k$$

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \Delta\boldsymbol{\mu}_k$$

Códigos: NPSOL, SNOPT, MUSCOD-II, fmincon, NAG, …

Métodos NLP

IP Interior Point incorporan las restricciones de desigualdad como funciones de barrera y resuelve una secuencia de problemas KKT relajados, por Newton- Raphson

$$\min_{x} J(\mathbf{x}) \qquad \mathbf{h}(\mathbf{x}) = \mathbf{0} \qquad \mathbf{g}(\mathbf{x}) \le \mathbf{0}$$

$$\Rightarrow$$

$$\min_{\mathbf{x},\boldsymbol{\varepsilon}} J(\mathbf{x}) \qquad \mathbf{h}(\mathbf{x}) = \mathbf{0} \qquad \mathbf{g}(\mathbf{x}) + \boldsymbol{\varepsilon} = \mathbf{0} \qquad \boldsymbol{\varepsilon} \ge \mathbf{0}$$

$$\Rightarrow$$

$$\min_{\mathbf{x},\boldsymbol{\varepsilon}} J(\mathbf{x}) - \eta \sum_{i=1}^{n_\varepsilon} \ln(\varepsilon_i) \qquad \mathbf{h}(\mathbf{x}) = \mathbf{0} \qquad \mathbf{g}(\mathbf{x}) + \boldsymbol{\varepsilon} = \mathbf{0}$$

Secuencia de problemas con $\eta \to 0$ en cada uno de los cuales se resuelven por Newton las condiciones KKT relajadas:
Códigos: IPOPT, KNITRO,…

$$\nabla_x J(\mathbf{x}) + \boldsymbol{\lambda}' \nabla_x \mathbf{c}(\mathbf{z}) - \boldsymbol{\nu}' = \mathbf{0}$$

$$\mathbf{c}(\mathbf{z}) = \mathbf{0}$$

$$X\boldsymbol{\nu} = \eta \mathbf{e}$$

$$\mathbf{c}(\mathbf{z}) = \begin{bmatrix} \mathbf{h}(\mathbf{x}) \\ \mathbf{g}(\mathbf{x}) + \boldsymbol{\varepsilon} \end{bmatrix}$$

Sensibilidad del óptimo

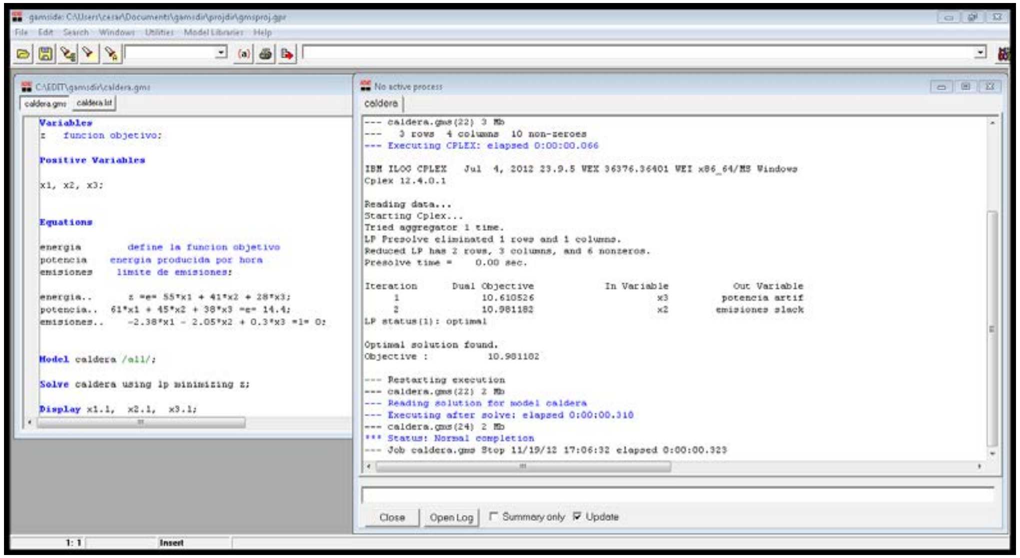
$$\min_x J(\mathbf{x})$$
$$\mathbf{h}(\mathbf{x}) = \mathbf{b}$$
$$\mathbf{g}(\mathbf{x}) \leq \mathbf{c}$$

Los multiplicadores de Lagrange λ, μ proporcionan la sensibilidad de la solución óptima J* respecto a cambios en las restricciones:

$$\boxed{\frac{\partial J(\mathbf{x}^*)}{\partial \mathbf{b}} = -\boldsymbol{\lambda}^{*\prime} \quad \frac{\partial J(\mathbf{x}^*)}{\partial \mathbf{c}} = -\boldsymbol{\mu}^{*\prime}}$$

Estos valores nos permiten calcular como se modifica el valor del óptimo cuando se relajan en una unidad las restricciones, lo cual puede ser importante en la toma de decisiones

Software



Entornos de modelado y optimización GAMS, AIMMS, XPRESS, Gurobi,…

Fáciles de usar, muchos solvers disponibles, **calculo de derivadas automático**, pero pocas facilidades de comunicación externa

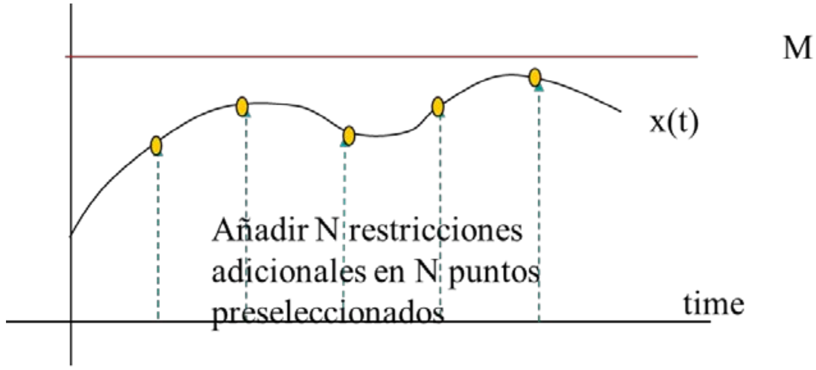
Librerías con solvers que pueden integrarse con otras aplicaciones:

NAG, CPLEX, IMSL, TOMLAB, Optimization Toolbox, WORHP,…

Requieren los gradientes o los calculan por diferencias finitas

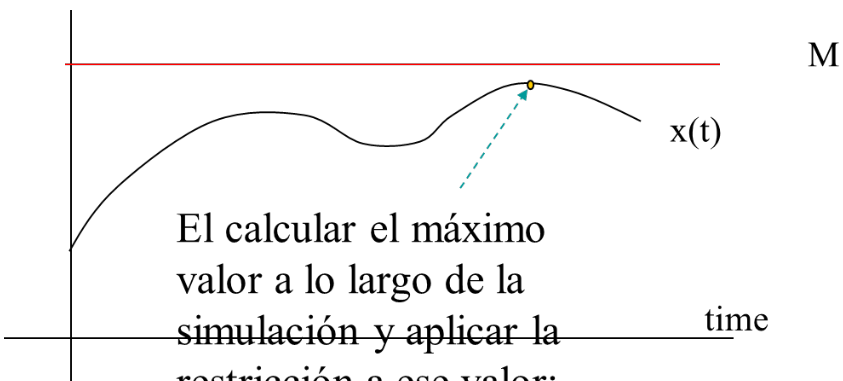
Restricciones de camino

Path Constraints



Añadir N restricciones adicionales en N puntos preseleccionados

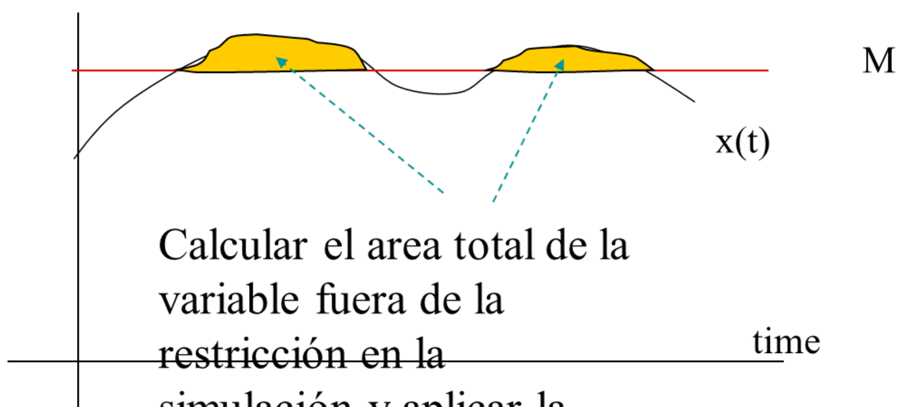
$x(t_1) \leq M, \; \ldots \; x(t_N) \leq M$



El calcular el máximo valor a lo largo de la simulación y aplicar la restricción a ese valor:

Max $x(t) \leq M$

Diversas estrategias



Calcular el area total de la variable fuera de la restricción en la simulación y aplicar la restricción adicional:

Area$(x) \leq 0$

$$\int_0^T \max(0, x - M)dt \leq 0$$

Gradientes (Sistemas continuos)

$$\min_{p} \quad J(p) = \int_0^T L(x, u(p))dt$$

$$F(\dot{x}, x, u(p)) = 0$$

$$g(x, u(p)) \leq 0$$

u(p) p parámetros libres

En general, problemas densos para los que los métodos SQP son adecuados

SQP y otros algoritmos NLP necesitan calcular los gradientes de la función de costo y las restricciones respecto a las variables de decisión

Métodos

- ✓ Diferencias finitas
- ✓ Ecuaciones de sensibilidad / Sistema adjunto
- ✓ Diferenciación automática, Simbólica

Ecuaciones de sensibilidad

$$\min_p \quad J(p) = \int_0^T L(x, u(p))dt$$

$$F(\dot{x}, x, u(p)) = 0$$

$$g(x, u(p)) \leq 0$$

u(p) p parámetros libres: variables de decisión

Cálculo de sensibilidades

$$s = \frac{\partial x}{\partial p} \qquad \text{sensitivity}$$

$$\frac{dJ}{dp_i} = \int_0^T \left(\frac{\partial L}{\partial x} \boxed{\frac{\partial x}{\partial p_i}} + \frac{\partial L}{\partial u} \frac{\partial u}{\partial p_i} \right) dt$$

$$\frac{\partial F}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial p_i} = \boxed{\frac{\partial F}{\partial \dot{x}} \frac{d}{dt} \boxed{\frac{\partial x}{\partial p_i}} = \frac{\partial F}{\partial x} \boxed{\frac{\partial x}{\partial p_i}} + \frac{\partial F}{\partial u} \frac{\partial u}{\partial p_i}}$$

Conjunto de ecuaciones diferenciales lineales de primer orden en s

Sistema extendido

$$s = \frac{\partial x}{\partial p}$$

$$\frac{\partial F}{\partial \dot{x}}\frac{ds_i}{dt} + \boxed{\frac{\partial F}{\partial x}} s_i + \frac{\partial F}{\partial u}\frac{\partial u}{\partial p_i} = 0$$

$$F(\dot{x}, x, u) = 0$$

✓ODEs de primer orden en s.

✓Las sensibilidades pueden obtenerse integrando el sistema extendido

✓Muchas ecuaciones ($n_x\ n_p + n_F$) pero las ecuaciones de sensibilidad tienen el mismo Jacobiano que el sistema original F

Se puede hacer un cálculo eficiente de las sensibilidades reusando el Jacobiano $\partial F/\partial x$ que se usa en la integración del sistema original

Diferenciación Automática para el cálculo del Jacobiano

Pueden usarse DASPK 3.0, IDAS,… como algoritmos de integración para obtener las sensibilidades en línea.

Integración del sistema extendido

$$\frac{\partial F}{\partial \dot{x}}\frac{ds_i}{dt} + \boxed{\frac{\partial F}{\partial x}} s_i + \frac{\partial F}{\partial u}\frac{\partial u}{\partial p_i} = 0$$

$$F(\dot{x}, x, u) = 0 \qquad s(0) = 0$$

Tras un paso de integración del DAE, todos los parámetros (tamaño del paso,…) se congelan y se calculan las sensibilidades para ese paso.

Los métodos por etapas permiten usar distintos criterios de error en el DAE y en las ecuaciones de sensibilidad.

Simultaneous corrector:

Las ecuaciones del modelo y sensibilidad se integran conjunta y simultáneamente.

Staggered methods:

Primero se integran las ecuaciones del modelo y luego las de sensibilidad

Staggered direct: La factorización del Jacobiano se hace cada paso.
Staggered corrector: La factorización del Jacobiano se hace cuando se necesita

Sensibilidades Adjoint method

$$\min_{u} \quad J(u) = \int_0^T L(x,u)dt$$

$$F(\dot{x}, x, u) = 0$$

$$g(x,u) \le 0$$

En muchos casos el interés directo no es el cálculo de las sensibilidades $s = \frac{\partial x}{\partial u}$

sino el gradiente de una variable J(u)

$$\frac{dJ}{du} = \int_0^T \left(\frac{\partial L}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial L}{\partial u} \right) dt$$

No depende de t

El cálculo de las sensibilidades usando la integración del sistema extendido (Forward method) no es eficiente cuando la dimensión de u, n_u, es alta.

En el caso de cálculo de la sensibilidad de una variable (independiente de t) respecto a un número de parámetros alto, es mejor usar el método del sistema adjunto (Adjoint method)

Método del sistema adjunto

Dadas las ecuaciones

$$J(u) = \int_0^T L(x,u)dt$$

$$F(\dot{x},x,u) = 0$$

Se puede definir la función:

$$\mathcal{L}(x,u) = J(u) - \int_0^T \lambda^* F(\dot{x},x,u)dt$$

como $\quad F(\dot{x},x,u) = 0$

$$\frac{d\mathcal{L}}{du} = \frac{dJ}{du} = \int_0^T \left(\frac{\partial L}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial L}{\partial u} \right) dt - \int_0^T \lambda^* \left(\frac{\partial F}{\partial u} + \frac{\partial F}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial F}{\partial \dot{x}}\frac{\partial \dot{x}}{\partial u} \right) dt$$

Integrando por partes el término

$$\int_0^T \lambda^* \left(\frac{\partial F}{\partial \dot{x}}\frac{\partial \dot{x}}{\partial u} \right) dt = \lambda^* \frac{\partial F}{\partial \dot{x}}\frac{\partial x}{\partial u}\bigg|_0^T - \int_0^T \frac{d}{dt}\left(\lambda^* \frac{\partial F}{\partial \dot{x}} \right) \frac{\partial x}{\partial u} dt$$

Con $\; u = \lambda^* \frac{\partial F}{\partial \dot{x}} \Big|$ $\qquad dv = \frac{\partial \dot{x}}{\partial u} dt$ $\qquad$ Resulta:

Método del sistema adjunto

$$\frac{dJ}{du} = \int_0^T \left(\frac{\partial L}{\partial u} - \lambda^* \frac{\partial F}{\partial u} \right) dt - \int_0^T \left(-\frac{\partial L}{\partial x} + \lambda^* \frac{\partial F}{\partial x} - \frac{d}{dt}\left(\lambda^* \frac{\partial F}{\partial \dot{x}} \right) \right) \frac{\partial x}{\partial u} dt -$$

$$- \left(\lambda^* \frac{\partial F}{\partial \dot{x}} \frac{\partial x}{\partial u} \right)\Bigg|_T + \left(\lambda^* \frac{\partial F}{\partial \dot{x}} \frac{\partial x}{\partial u} \right)\Bigg|_0$$

Selecting λ such that

$$\boxed{\lambda^* \frac{\partial F}{\partial \dot{x}}\Bigg|_{t=T} = 0}$$

$$\boxed{-\frac{\partial L}{\partial x} + \lambda^* \frac{\partial F}{\partial x} - \frac{d}{dt}\left(\lambda^* \frac{\partial F}{\partial \dot{x}} \right) = 0}$$

Sistema adjunto

Integrando el modelo pueden obtenerse valores para integrar el sistema adjunto y con los valores de λ calcular el gradiente deseado. Nótese que ∂x/∂u =0 en t =0. El problema se resuelve integrando el sistema adjunto hacia atrás en el tiempo con condición inicial dada en t = T (para DAEs de orden 0 y 1)

$$\frac{dJ}{du} = \int_0^T \left(\frac{\partial L}{\partial u} - \lambda^* \frac{\partial F}{\partial u} \right) dt$$

Ejemplo reactor: Jacobiano

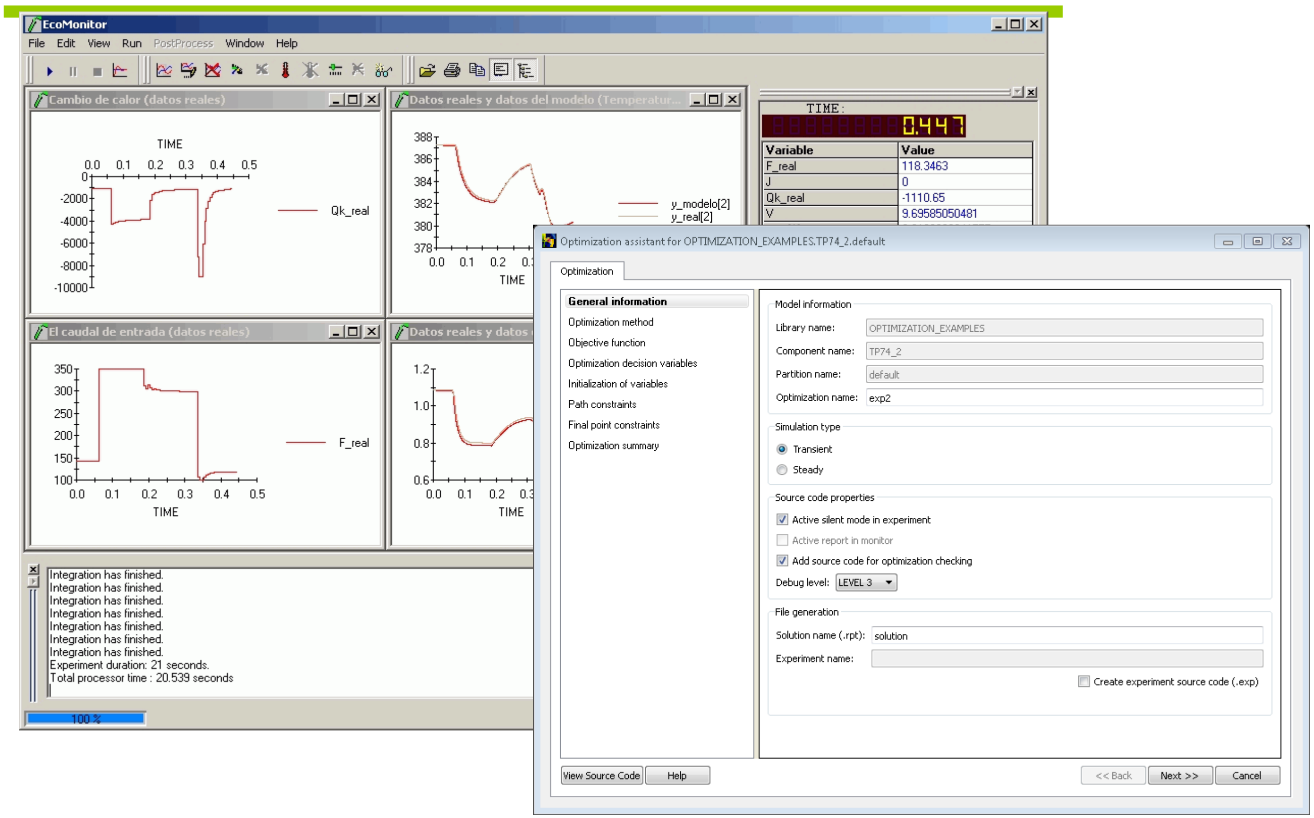
$$V\frac{dc_A}{dt} = q(c_{Ai} - c_A) - Vkc_A \qquad k = \beta e^{-E/R(T+273.14)}$$

$$V\rho c_p \frac{dT}{dt} = q\rho c_p (T_i - T) - Vkc_A H - UA(T - T_r)$$

$$V_r\rho_r c_{pr}\frac{dT_r}{dt} = F_r\rho_r c_{pr}(T_{ri} - T_r) + UA(T - T_r)$$

$$\frac{\partial F}{\partial x} = \begin{bmatrix} \frac{q}{V} + k & \frac{c_A kE}{R(T+273.14)^2} & 0 \\ \frac{kH}{\rho c_p} & \frac{q}{V} + \frac{UA}{V\rho c_p} + \frac{c_A HkE}{\rho c_p R(T+273.14)^2} & -\frac{UA}{V\rho c_p} \\ 0 & \frac{-UA}{V_r\rho_r c_{pr}} & \frac{F_r}{V_r} + \frac{UA}{V_r\rho_r c_{pr}} \end{bmatrix}$$

Software



Entornos de simulación unidos a solvers NLP

Asistentes para la definición del problema y generación automática de código de optimización

EcosimPro, gProms, Dymola,..

Muy importante el cálculo de sensibilidades para la calidad de la solución.
Errores relativos Simulación / Optimización

Ejemplo: Reactor químico

$$V\frac{dc_A}{dt} = q(c_{Ai} - c_A) - V\beta e^{-E/RT} c_A$$

$$V\rho c_p \frac{dT}{dt} = q\rho c_p (T_i - T) - Vkc_A H - UA(T - T_r)$$

$$V_r \rho_r c_{pr} \frac{dT_r}{dt} = F_r \rho_r c_{pr} (T_{ri} - T_r) + UA(T - T_r)$$

$$c_B = c_{Ai} - c_A$$

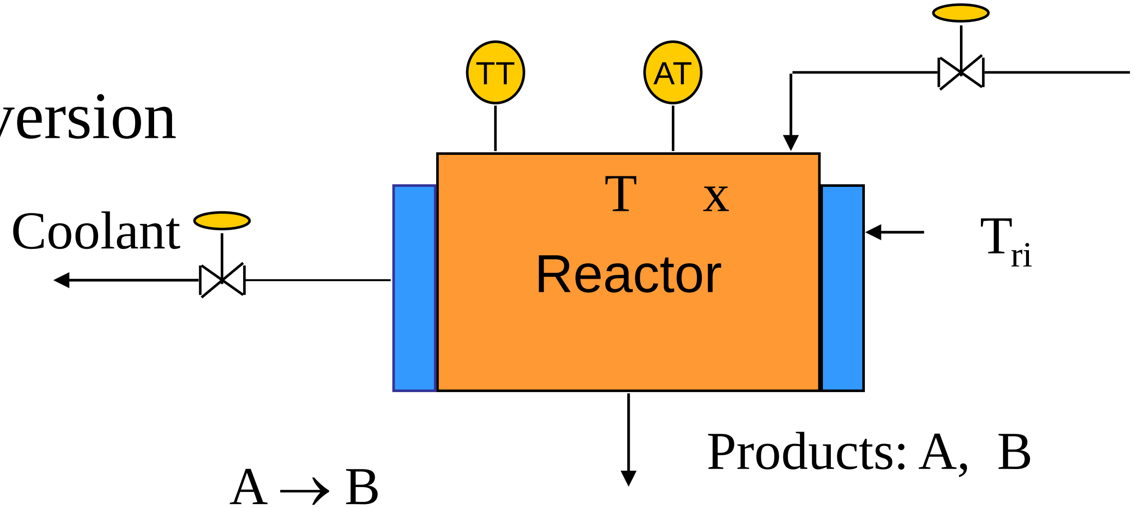
$$x = c_B / c_{Ai} \qquad x \text{ conversion}$$

$$\boxed{\text{Max} \quad J = q\, c_B}$$

$$\boxed{\begin{array}{l} T_{min} \le T \le T_{max} \\ x_{min} \le x \le 1 \\ q_{min} \le q \le q_{max} \\ F_{r\,min} \le F_r \le F_{r\,max} \end{array}}$$

Row material:
Product A

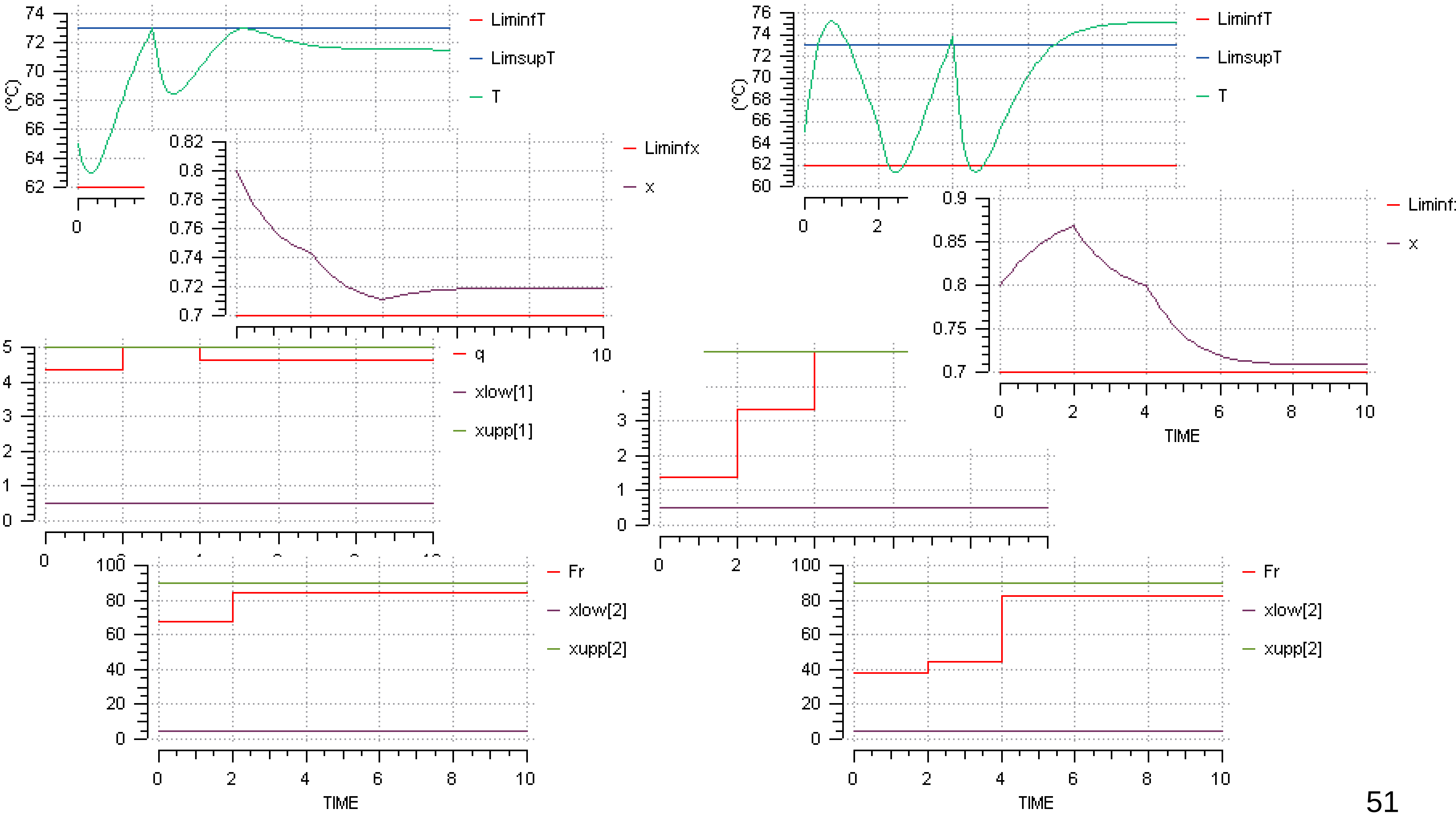
T_i, q, $c_A i$



$A \rightarrow B$

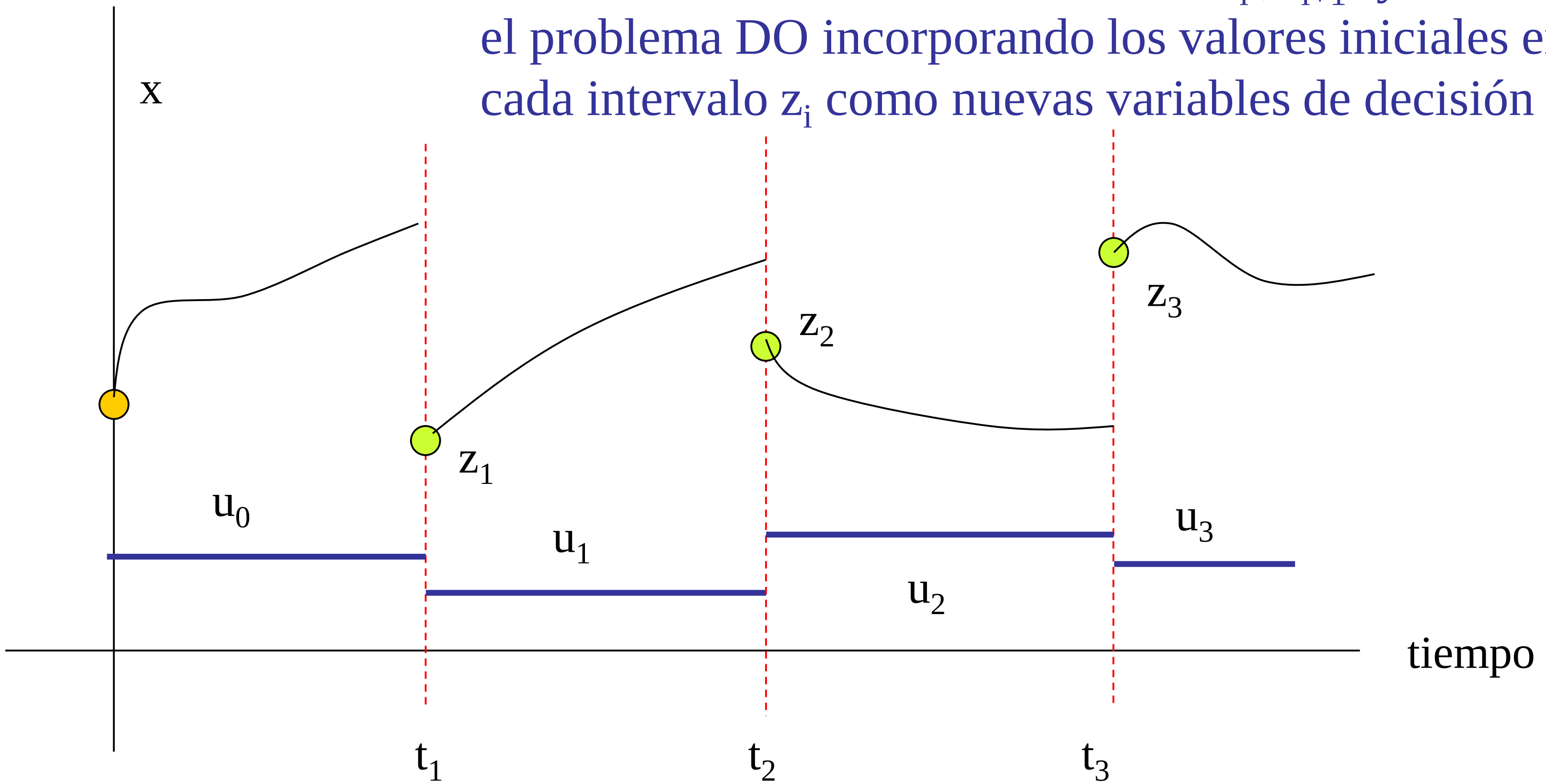
50

Con / sin sensibilidades



Multiple shooting

Dividir el horizonte en n intervalos $[t_i , t_{i+1}]$ y resolver el problema DO incorporando los valores iniciales en cada intervalo z_i como nuevas variables de decisión



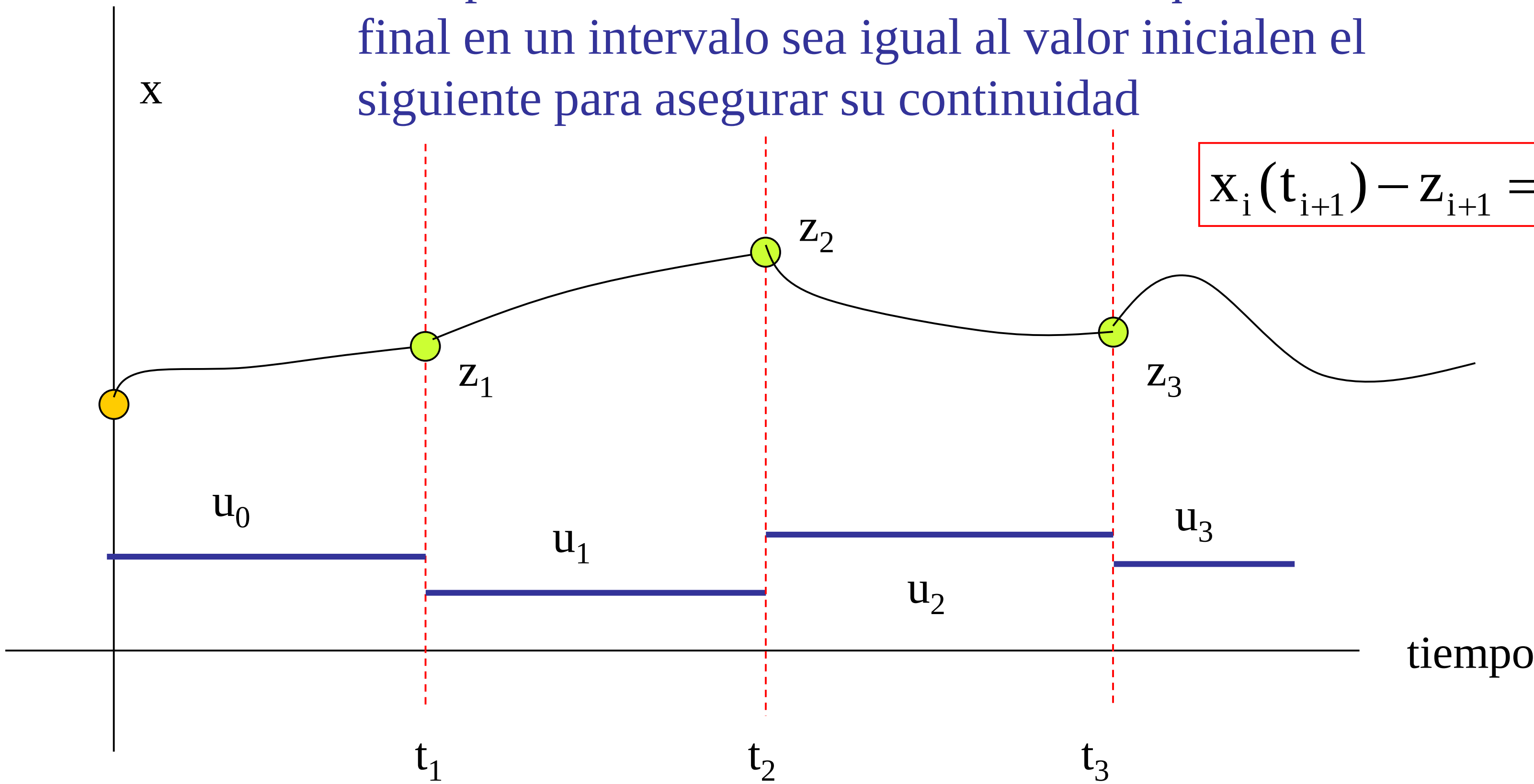
$$J = \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} L(x_i(t), u_i, z_i))dt$$

x_i x en el intervalo i

Multiple shooting

…Imponiendo además la condición de que el estado final en un intervalo sea igual al valor inicialen el siguiente para asegurar su continuidad

$$\boxed{x_i(t_{i+1}) - z_{i+1} = 0}$$



Multiple shooting

- Ventajas:
  - La inicialización de x puede estar mas cerca de la trayectoria deseada, lo que facilita l convergencia
  - Se pueden imponer restricciones de camino sobre z_i
  - La evolución de la etapa i es independiente de la i+1
  - Facilita la paralelización
  - Permite usar métodos secuenciales con sistemas inestables
- Desventajas:
  - Mayor complejidad

$$J = \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} L(x_i(t), u_i, z_i))dt$$

$$v = \{u_0, z_1, u_1, \dots, z_i, u_i, \dots, z_{n-1}, u_{n-1}\}$$

$$\min_v \quad J$$

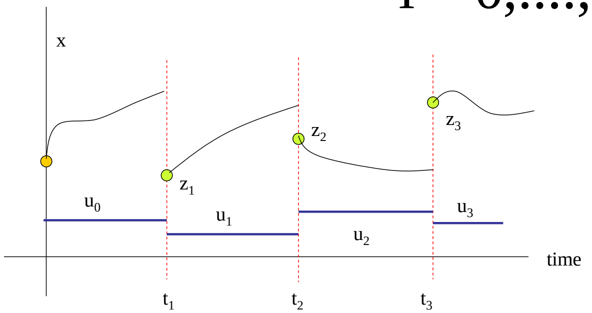
$$F_i(\dot{x}_i, x_i, u_i, t) = 0$$

$$x_i(t_i) = z_i$$

$$g_i(z_i, u_i, t_i) \le 0$$

$$x_i(t_{i+1}) - z_{i+1} = 0$$

$$i = 0, \dots, n$$



Enfoque simultaneo

$$\min_{p} \quad J(p) = \int_0^T L(x, u(p))dt$$
$$F(\dot{x}, x, u(p)) = 0$$
$$g(x, u(p)) \leq 0$$

Discretizar totalmente las ecuaciones

$$F(\dot{x}, x, u(p)) = 0$$

$$\Downarrow$$

$$F(\frac{x(t+\Delta t) - x(t)}{\Delta t}, x(t+\Delta t), u(p)) = 0$$

t = 0, Δt, 2Δt,……

$$\int_0^T L(x, u(p))dt$$

$$\Downarrow$$

$$\sum_{j=0}^{N} [L(x(j), u(j,p)] \Delta t$$

Sistema de ecuaciones algebraicas

Enfoque simultaneo

$$\min_{p} \quad J(p) = \int_0^T L(x, u(p))dt$$

$$F(\dot{x}, x, u(p)) = 0$$

$$g(x, u(p)) \leq 0$$

✓El número de variables de decisión y ecuaciones se incrementa de acuerdo a la discretización.

✓Facilita la imposición de restricciones de camino, el cálculo de gradientes y se puede usar en sistemas inestables

✓Puede haber problemas con la discretización de las DAE

$$\min_{p,x} \quad J = \sum_{j=0}^{N} [L(x(j), u(j,p)]\Delta t$$

$$F(x(1), x(0), u(0,p)) = 0 \qquad g(x(0), u(0,p)) \leq 0$$

$$F(x(2), x(1), u(1,p)) = 0 \qquad g(x(1), u(1,p)) \leq 0$$

$$F(x(3), x(2), u(2,p)) = 0 \qquad g(x(2), u(2,p)) \leq 0$$

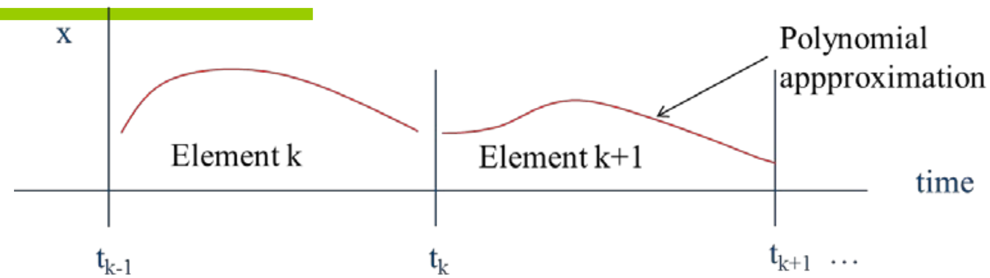
.....

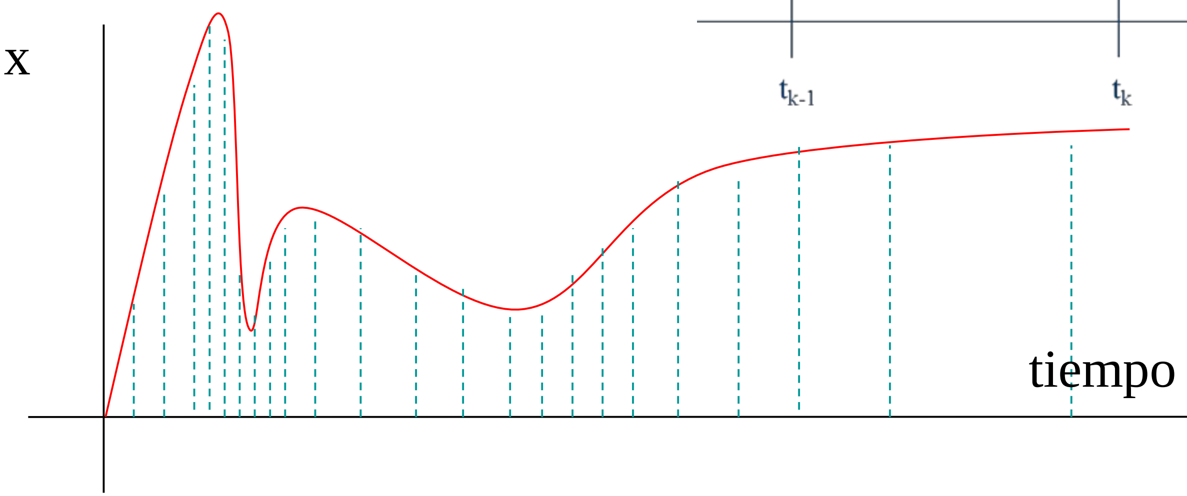
$$F(x(N), x(N-1), u(N-1,p)) = 0 \qquad g(x(N-1), u(N-1,p)) \leq 0$$

x(i) y p son variables de decisión

Solución con software NLP

Discretización





Colocación ortogonal por elementos finitos

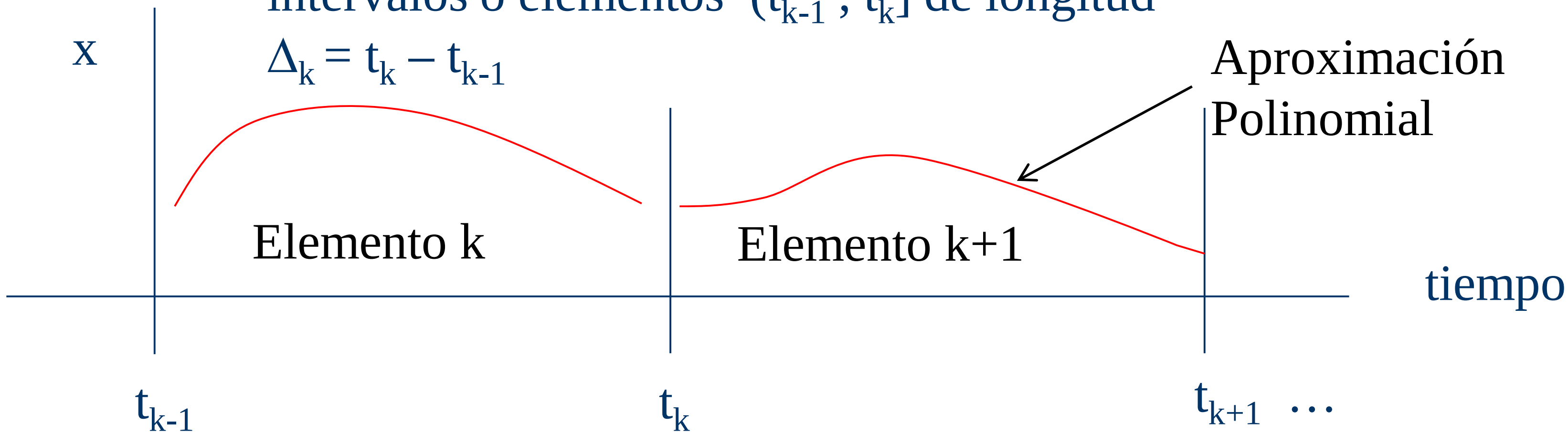
¿nº de elementos?

La integración de sistemas stiff usa métodos de paso y estructura variable para mantener el error de integración bajo cotas.

El uso de métodos de paso fijo obliga a usar un gran número de intervalos, resultando en un alto número de ecuaciones y variables y no garantiza la calidad

Colocación en elementos finitos

El horizonte temporal se divide en K intervalos o elementos $(t_{k-1}, t_k]$ de longitud $\Delta_k = t_k - t_{k-1}$

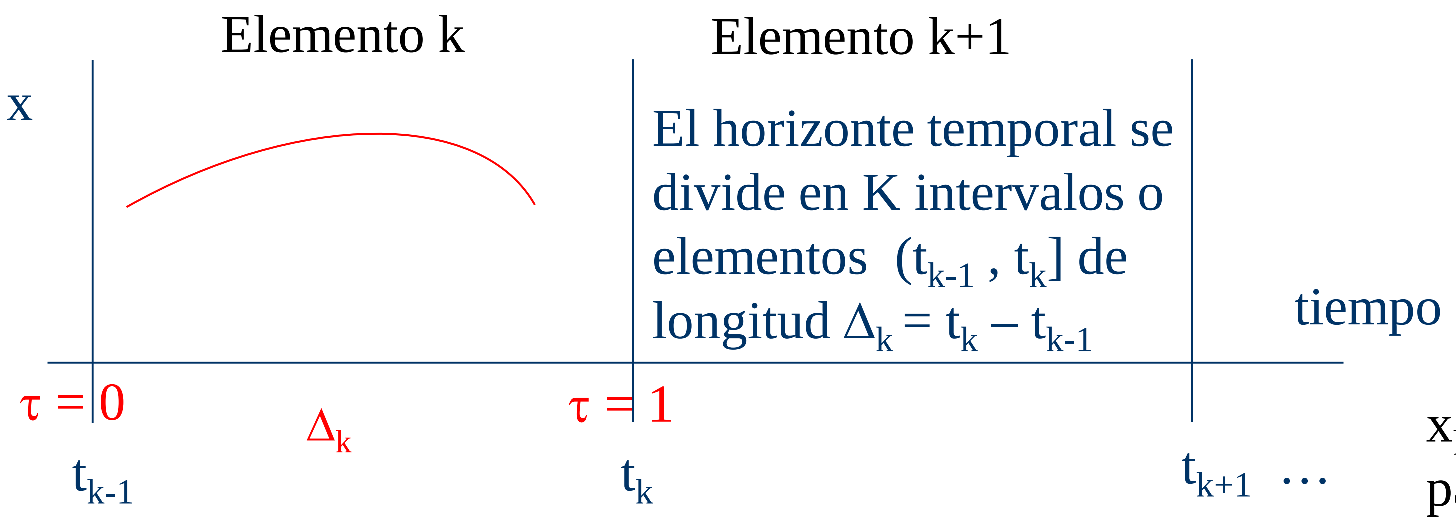


En cada intervalo $(t_{k-1}, t_k]$ la solución x se aproxima por una fórmula polinómica. Esto proporciona una aproximación suave en elemento, al tiempo que permite discontinuidades en la señal de control.

Pueden usarse muchos tipos de aproximaciones polinómicas

El número de elementos r K no tiene por que ser grande

Colocación en elementos finitos



El horizonte temporal se divide en K intervalos o elementos $(t_{k-1}, t_k]$ de longitud $\Delta_k = t_k - t_{k-1}$

x_{kj} parámetros a calcular

$$\mathbf{x}(t) \approx \sum_{j=0}^{P} P_j(\tau)\mathbf{x}_{kj}$$

$k = 1,\ldots,K$

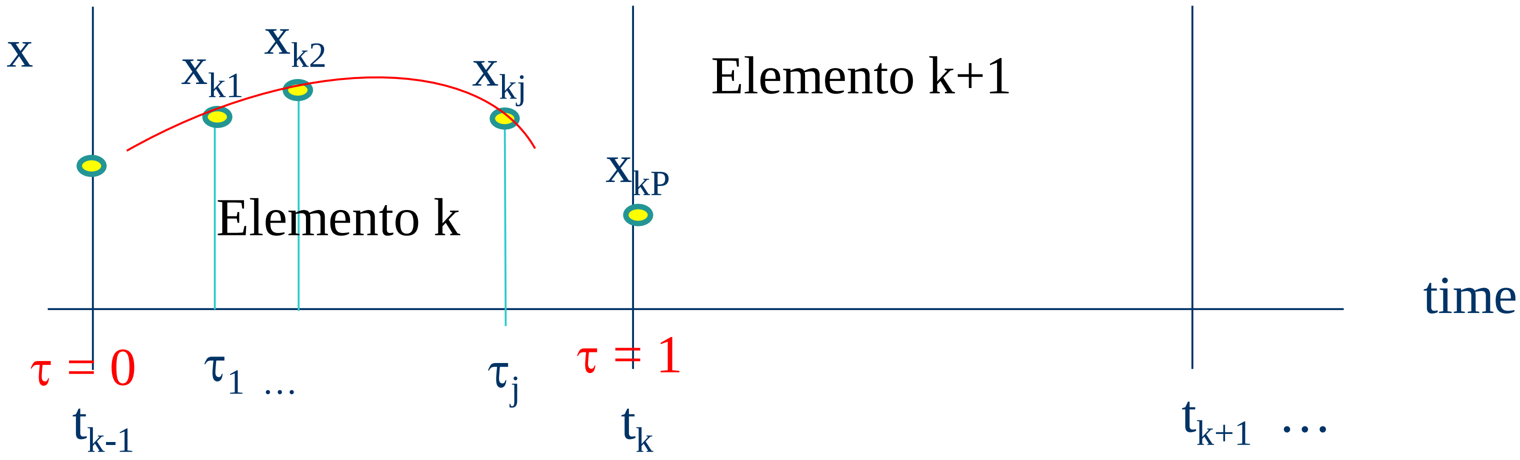
$$t = t_{k-1} + \tau\Delta_k \qquad \tau \in (0,1]$$

$$\dot{\mathbf{x}}(t) \approx \sum_{j=0}^{P} \frac{\dot{P}_j(\tau)\mathbf{x}_{kj}}{\Delta_k}$$

τ Tiempo normalizado

Una posibilidad es aproximar la evolución temporal de las variables por una combinación lineal de polinomios conocidos $P_j(\tau)$ de orden P. Típicamente se usan polinomios de interpolación de Lagrange.

Polinomios de interpolación de Lagrange



$$\mathbf{x}(t) \approx \sum_{j=0}^{P} P_j(\tau)\mathbf{x}_{kj}$$

$$P_j(\tau) = \prod_{i=0, i\neq j}^{P} \frac{\tau - \tau_i}{\tau_j - \tau_i}$$

Se seleccionan P+1 puntos de interpolación $\tau_0 = 0, \tau_1, .., \tau_P$

$$t = t_{k-1} + \tau\Delta_k \qquad \tau \in (0,1]$$

$$x(t_{kj}) = x(t_{k-1} + \tau_j\Delta_k) = x_{kj} \qquad \tau_i < \tau_{i+1}$$

$$\dot{\mathbf{x}}(t) \approx \sum_{j=0}^{P} \frac{\dot{P}_j(\tau)\mathbf{x}_{kj}}{\Delta_k}$$

Los parámetros x_{kj} tienen un significado claro cuando se usan los polinomios de Lagrange

Polinomios de Lagrange

$$P_j(\tau) = \prod_{i=0, i\neq j}^{P} \frac{\tau - \tau_i}{\tau_j - \tau_i}$$

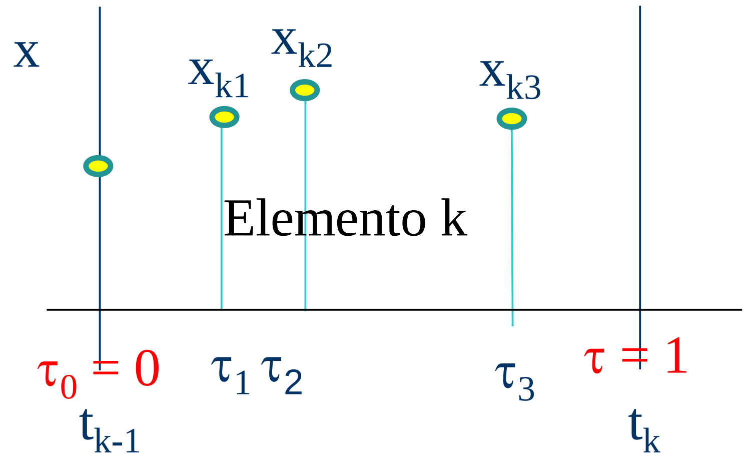
$$P_0 = \frac{\tau - \tau_1}{\tau_0 - \tau_1}\frac{\tau - \tau_2}{\tau_0 - \tau_2}\frac{\tau - \tau_3}{\tau_0 - \tau_3}$$

$$P_1 = \frac{\tau - \tau_0}{\tau_1 - \tau_0}\frac{\tau - \tau_2}{\tau_1 - \tau_2}\frac{\tau - \tau_3}{\tau_1 - \tau_3}$$

$$P_2 = \frac{\tau - \tau_0}{\tau_2 - \tau_0}\frac{\tau - \tau_1}{\tau_2 - \tau_1}\frac{\tau - \tau_3}{\tau_2 - \tau_3}$$

$$P_3 = \frac{\tau - \tau_0}{\tau_3 - \tau_0}\frac{\tau - \tau_1}{\tau_3 - \tau_1}\frac{\tau - \tau_2}{\tau_3 - \tau_2}$$

$$x(t_{k-1} + \tau_j \Delta_k) = x_{kj}$$



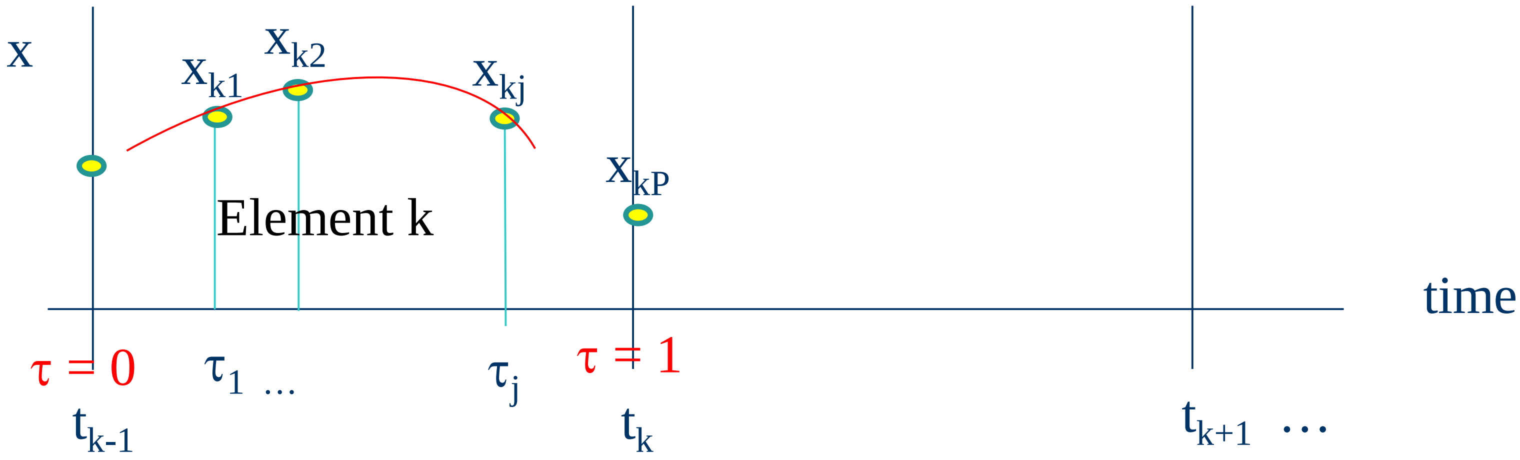
Ejemplo con P= 3 $\quad \mathbf{x}(t) \approx \sum_{j=0}^{P} P_j(\tau)\mathbf{x}_{kj}$

Para $\tau = \tau_1$ $\quad P_0 = P_2 = P_3 = 0 \qquad P_1 = 1$

$x(t_{k-1} + \tau_1\Delta_k) =$

$= P_0\, x_{k0} + P_1\, x_{k1} + P_2\, x_{k2} + P_3\, x_{k3} = x_{k1}$

Colocación en elementos finitos



Se impone que se satisfagan las ecuaciones DAE en los puntos de colocación.

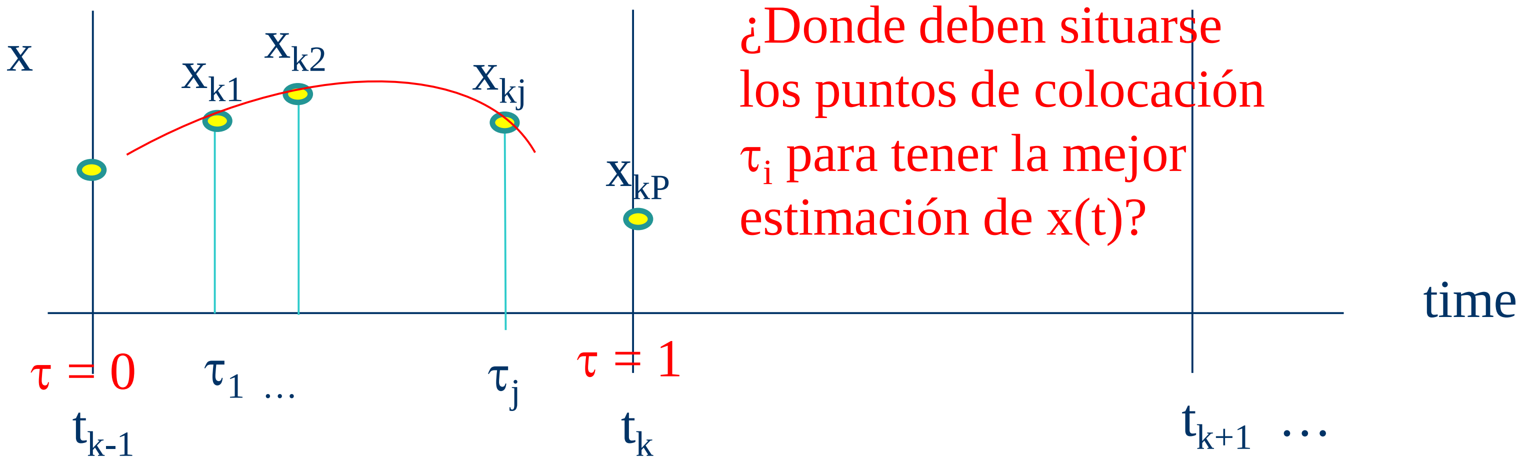
$$F(\dot{x}, x, u(p)) = 0$$

$$F\left(\sum_{j=0}^{P} \frac{\dot{P}_j(\tau_i)\mathbf{x}_{kj}}{\Delta_k}, \mathbf{x}_{ki}, u(p)\right) = 0 \qquad k = 1,..K$$

Esta condición proporciona un conjunto de ecuaciones que permiten calcular los coeficientes x_{ki} desconocidos

Los P+1 puntos de colocación se sitúan en posiciones fijas τ_i en cada elemento k. Existen diferentes métodos para situarlos

Colocación Ortogonal



$$F\left(\sum_{j=0}^{P}\frac{\dot{P}_j(\tau_i)\mathbf{x}_{kj}}{\Delta_k}, \mathbf{x}_{ki}, u(p)\right) = 0$$

k = 1,..K
i = 1, …P

Para reducir P se escogen polinomios ortogonales

$$\int_0^1 P_j(\tau)P_i(\tau)d\tau = 0 \quad i \neq j$$

Colocación Ortogonal

Shifted Gauss–Legendre and Radau roots as collocation points.

Degree *P*	Legendre Roots	Radau Roots
1	0.500000	1.000000
2	0.211325	0.333333
	0.788675	1.000000
3	0.112702	0.155051
	0.500000	0.644949
	0.887298	1.000000
4	0.069432	0.088588
	0.330009	0.409467
	0.669991	0.787659
	0.930568	1.000000
5	0.046910	0.057104
	0.230765	0.276843
	0.500000	0.583590
	0.769235	0.860240
	0.953090	1.000000

τ_0 es siempre = 0

Los puntos de colocación τ_i, i = 1,…,P se seleccionan como las raíces de polinomios de tipo Gauss-Jacobi, típicamente:

$$P_P^{Legendre}(\tau) = \sum_{j=0}^{P} (-1)^{P-j} \tau^j \gamma_j$$

$$\gamma_0 = 1$$

$$\gamma_j = \frac{(P-j+1)(P+j)}{j^2}$$

Dan mas exactitud

$$P_P^{Radau}(\tau) = \sum_{j=0}^{P} (-1)^{P-j} \tau^j \gamma_j$$

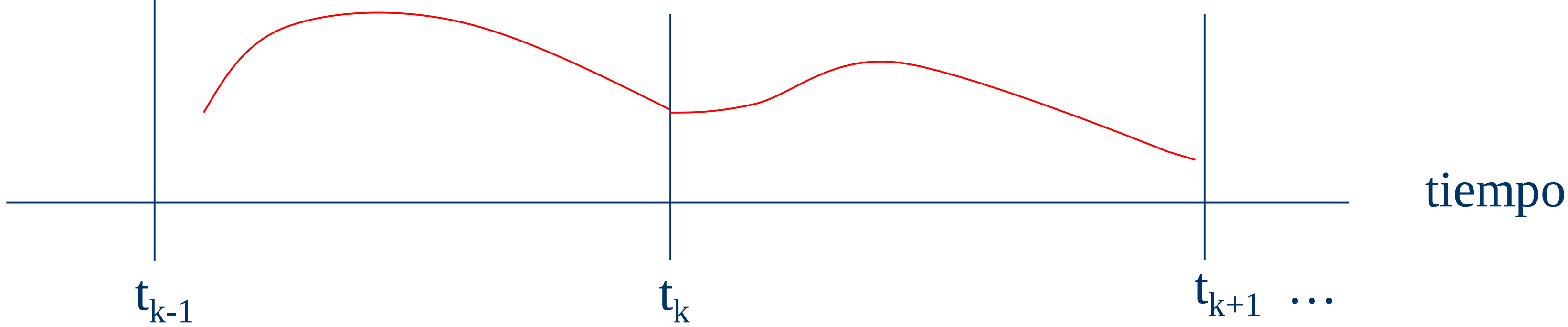
$$\gamma_0 = 1$$

$$\gamma_j = \frac{(P-j+1)(P+j+1)}{j^2}$$

Dan mas robustez

Colocación Ortogonal

La continuidad de las trayectorias a lo largo de los elementos finitos $(t_{k-1}, t_k]$ se impone mediante:



$$F\left(\sum_{j=0}^{P} \frac{\dot{P}_j(\tau_i)\mathbf{x}_{kj}}{\Delta_k}, \mathbf{x}_{ki}, u(p)\right) = 0 \qquad \begin{array}{l} k = 1,..K \\ i = 1, \ldots P \end{array}$$

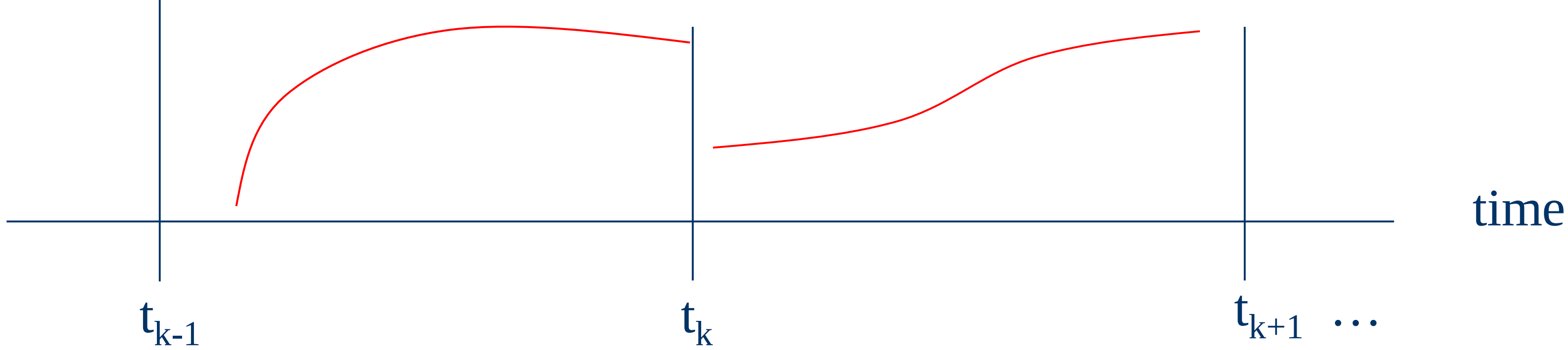
En lugar de estas ecuaciones, en los puntos $\tau_0 = 0$ se usa la continuidad de los estados, y en $t = 0$ las condiciones iniciales para generar ecuaciones que las sustituyan y que garanticen soluciones acorde a lo deseado

$$\mathbf{x}(t_k) = \mathbf{x}_{k+1,0} = \mathbf{x}_{k,P}$$

$$\mathbf{x}(t_0) = \mathbf{x}_{10} = \mathbf{x}_0$$

Colocación Ortogonal

Si se desea, las variables de control pueden representarse también mediante polinomios de interpolación de Lagrange en los elementos finitos $(t_{k-1}, t_k]$



$$\mathbf{u}(t) \approx \sum_{j=1}^{P} \bar{P}_j(\tau)\mathbf{u}_{kj}$$

$$\bar{P}_j(\tau) = \prod_{i=1, i \neq j}^{P} \frac{\tau - \tau_i}{\tau_j - \tau_i}$$

$$t = t_{k-1} + \tau \Delta_k \qquad \tau \in (0,1]$$

No se impone la continuidad de las trayectorias de control en los elementos finitos $(t_{k-1}, t_k]$

Pueden usarse métodos simultáneos de optimización con sistemas inestables

Ejemplo

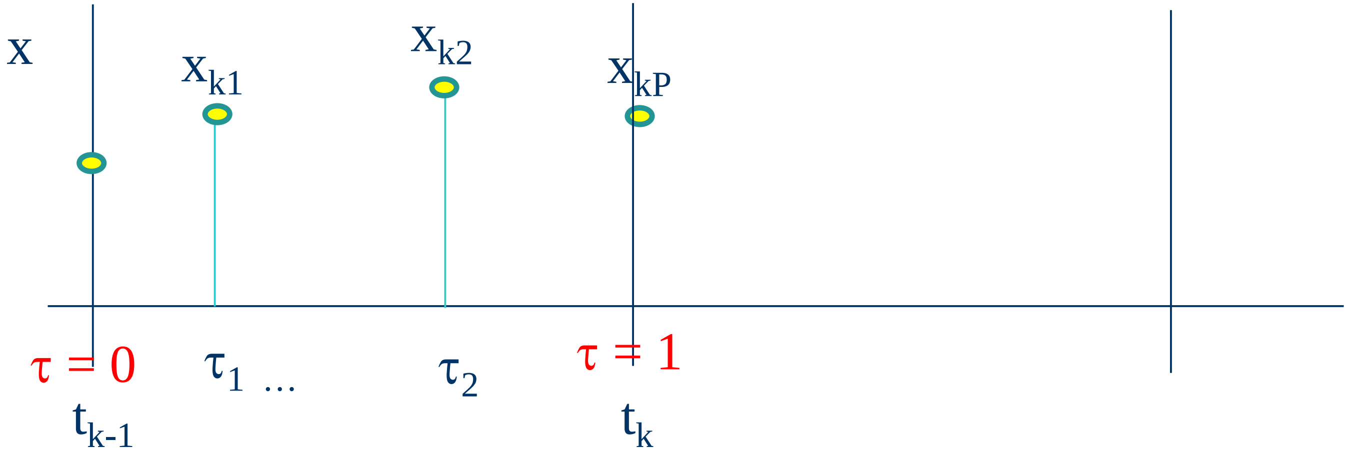
Integrar entre t = 0 y 1

$$\dot{x} = x^2 - 2x + 1 \qquad x(0) = -3$$

Se seleccionan K = 2 elementos finitos de igual tamaño

$\Delta_k = (1 - 0)/2 = 0.5$

P = 3 puntos de colocación



Los puntos de colocación de Radau para P =3 son:

$\tau_0 = 0 \quad \tau_1 = 0.155051 \quad \tau_2 = 0.644949 \quad \tau_3 = 1$

Ejemplo

$$P_j(\tau) = \prod_{i=0, i\neq j}^{P} \frac{\tau - \tau_i}{\tau_j - \tau_i}$$

Los puntos de colocación de Radau para P =3 son:
$\tau_0 = 0 \quad \tau_1 = 0.155051 \quad \tau_2 = 0.644949 \quad \tau_3 = 1$

$$P_0 = \frac{\tau - \tau_1}{\tau_0 - \tau_1}\frac{\tau - \tau_2}{\tau_0 - \tau_2}\frac{\tau - \tau_3}{\tau_0 - \tau_3} = -10\tau^3 + 18\tau^2 - 9\tau + 1$$

$$P_1 = \frac{\tau - \tau_0}{\tau_1 - \tau_0}\frac{\tau - \tau_2}{\tau_1 - \tau_2}\frac{\tau - \tau_3}{\tau_1 - \tau_3} = 15.5808\,\tau^3 - 25.6296\tau^2 + 10.0488\tau$$

$$P_2 = \frac{\tau - \tau_0}{\tau_2 - \tau_0}\frac{\tau - \tau_1}{\tau_2 - \tau_1}\frac{\tau - \tau_3}{\tau_2 - \tau_3} = -8.9141\tau^3 + 10.2963\tau^2 - 1.3821\tau$$

$$P_3 = \frac{\tau - \tau_0}{\tau_3 - \tau_0}\frac{\tau - \tau_1}{\tau_3 - \tau_1}\frac{\tau - \tau_2}{\tau_3 - \tau_2} = 3.3333\tau^3 - 2.6667\tau^2 + 0.3333\tau$$

$$x(t_{k-1} + \tau_j\Delta_k) = x_{kj} \qquad \mathbf{x}(t) \approx \sum_{j=0}^{P} P_j(\tau)\mathbf{x}_{kj} \qquad t = t_{k-1} + \tau\Delta_k \qquad \tau \in (0,1]$$

Ejemplo

$$\dot{\mathbf{x}}(t) \approx \sum_{j=0}^{P} \frac{\dot{P}_j(\tau)\mathbf{x}_{kj}}{\Delta_k}$$

Los puntos de colocación de Radau para P =3 son:

$\tau_0 = 0 \quad \tau_1 = 0.155051 \quad \tau_2 = 0.644949 \quad \tau_3 = 1$

$$\dot{P}_0(\tau) = -30\tau^2 + 36\tau - 9$$

$$\dot{x} = x^2 - 2x + 1 \qquad x(0) = -3$$

$$\dot{P}_1(\tau) = 46.7423\tau^2 - 51.2592\tau + 10.0488$$

$$\dot{P}_2(\tau) = -26.7423\tau^2 + 20.5925\tau - 1.3821$$

$$\dot{P}_3(\tau) = 10\tau^2 - 5.3333\tau + 0.3333$$

$$\sum_{j=0}^{3} \frac{\dot{P}_j(\tau)\mathbf{x}_{kj}}{0.5} = \mathbf{x}^2 - 2\mathbf{x} + 1$$

$$\mathbf{x}(t_{k-1} + \tau_j\Delta_k) = \mathbf{x}_{kj}$$

$k = 1,2$

$$t = t_{k-1} + \tau\Delta_k \qquad \tau \in (0,1]$$

Ejemplo

$$\dot{x} = x^2 - 2x + 1 \qquad x(0) = -3 \qquad \Longrightarrow \qquad \sum_{j=0}^{3} \frac{\dot{P}_j(\tau)\mathbf{x}_{kj}}{0.5} = \mathbf{x}^2 - 2\mathbf{x} + 1 \qquad k = 1,2$$

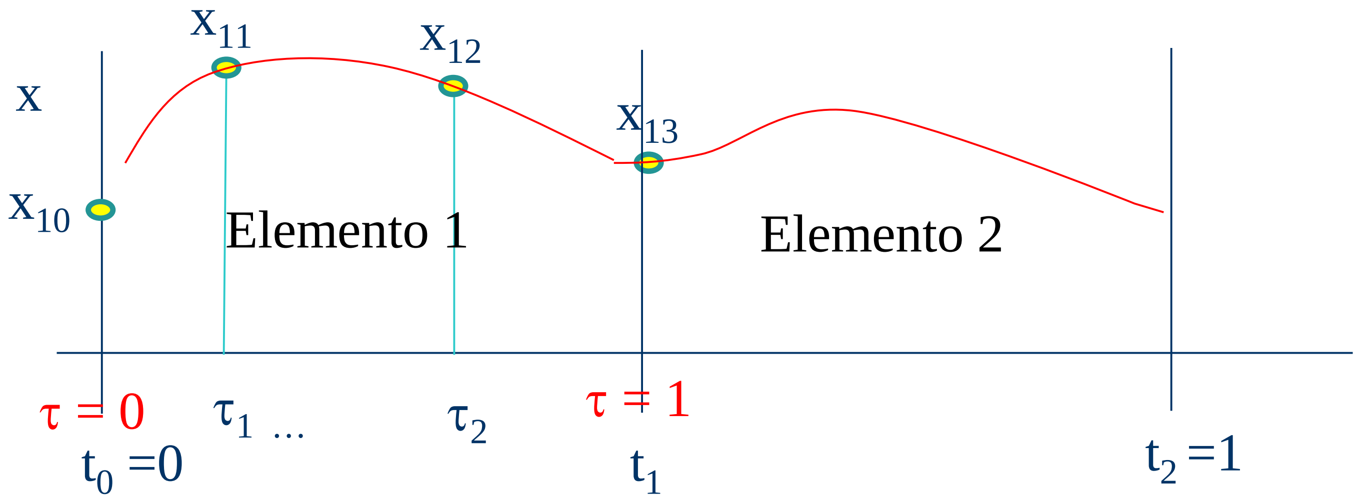
En los puntos de colocación τ_i : $\displaystyle\sum_{j=0}^{3} \frac{\dot{P}_j(\tau_i)\mathbf{x}_{kj}}{0.5} = \mathbf{x}_{ki}^2 - 2\mathbf{x}_{ki} + 1 \qquad k = 1,2 \quad i = 1, \ldots 3$

$$(-30\tau_i^2 + 36\tau_i - 9)x_{10} + (46.7423\tau_i^2 - 51.2592\tau_i + 10.0488)x_{11} +$$
$$+ (-26.7423\tau_i^2 + 20.5925\tau_i - 1.3821)x_{12} + (10\tau_i^2 - 5.3333\tau_i + 0.3333)x_{13} =$$
$$= 0.5(x_{1i}^2 - 2x_{1i} + 1) \qquad i = 1,2,3$$

$$(-30\tau_i^2 + 36\tau_i - 9)x_{20} + (46.7423\tau_i^2 - 51.2592\tau_i + 10.0488)x_{21} +$$
$$+ (-26.7423\tau_i^2 + 20.5925\tau_i - 1.3821)x_{22} + (10\tau_i^2 - 5.3333\tau_i + 0.3333)x_{23} =$$
$$= 0.5(x_{2i}^2 - 2x_{2i} + 1) \qquad i = 1,2,3$$

8 incógnitas, 6 ecuaciones

Ejemplo



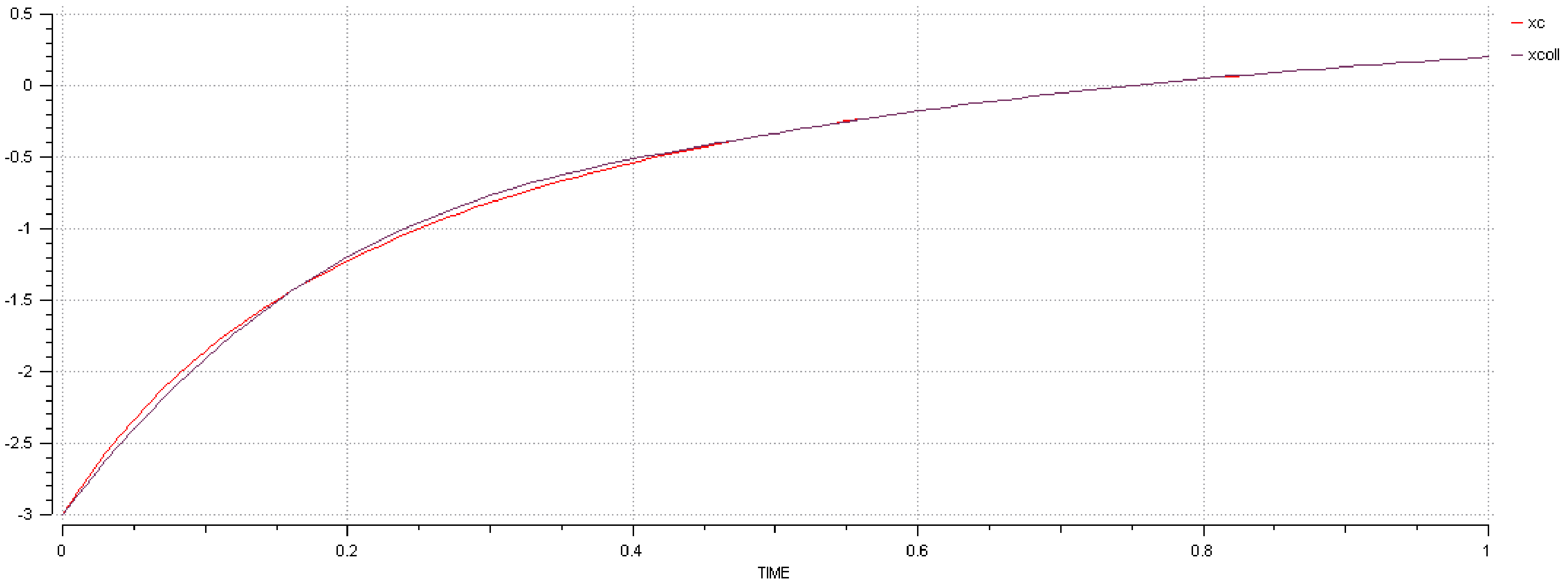
$$\mathbf{x}(t_k) = \mathbf{x}_{k+1,0} = \mathbf{x}_{k,P} = \sum_{j=0}^{P} P_j(1)\mathbf{x}_{k,j} \qquad \mathbf{x}(0.5) = \mathbf{x}_{20} = \mathbf{x}_{13} = \sum_{j=0}^{3} P_j(1)\mathbf{x}_{1j}$$

$$\mathbf{x}(t_0) = \mathbf{x}_{10} = \mathbf{x}_0 \qquad \mathbf{x}(0) = \mathbf{x}_{10} = -3$$

8 incógnitas, 8 ecuaciones

Las condiciones iniciales y de continuidad proporcionan las otras dos ecuaciones

Ejemplo



$$\dot{x} = x^2 - 2x + 1 \qquad x(0) = -3$$

P = 2

$$P_j(\tau) = \prod_{i=0, i\neq j}^{P} \frac{\tau - \tau_i}{\tau_j - \tau_i}$$

The Radau collocation points for P =2 are:
$\tau_0 = 0$ $\tau_1 = 0.333333$ $\tau_2 = 1$

$$P_0 = \frac{\tau - \tau_1}{\tau_0 - \tau_1}\frac{\tau - \tau_2}{\tau_0 - \tau_2} = 3\tau^2 - 4\tau + 1 \qquad \dot{P}_0 = 6\tau - 4$$

$$P_1 = \frac{\tau - \tau_0}{\tau_1 - \tau_0}\frac{\tau - \tau_2}{\tau_1 - \tau_2} = -4.5\tau^2 + 4.5\tau \qquad \dot{P}_1 = -9\tau + 4.5$$

$$P_2 = \frac{\tau - \tau_0}{\tau_2 - \tau_0}\frac{\tau - \tau_1}{\tau_2 - \tau_1} = 1.5\tau^2 - 0.5\tau \qquad \dot{P}_2 = 3\tau - 0.5$$

$$x(t_{k-1} + \tau_j \Delta_k) = x_{kj}$$

Derivative of $P_j(\tau)$ at τ_i

P= 3, Radau

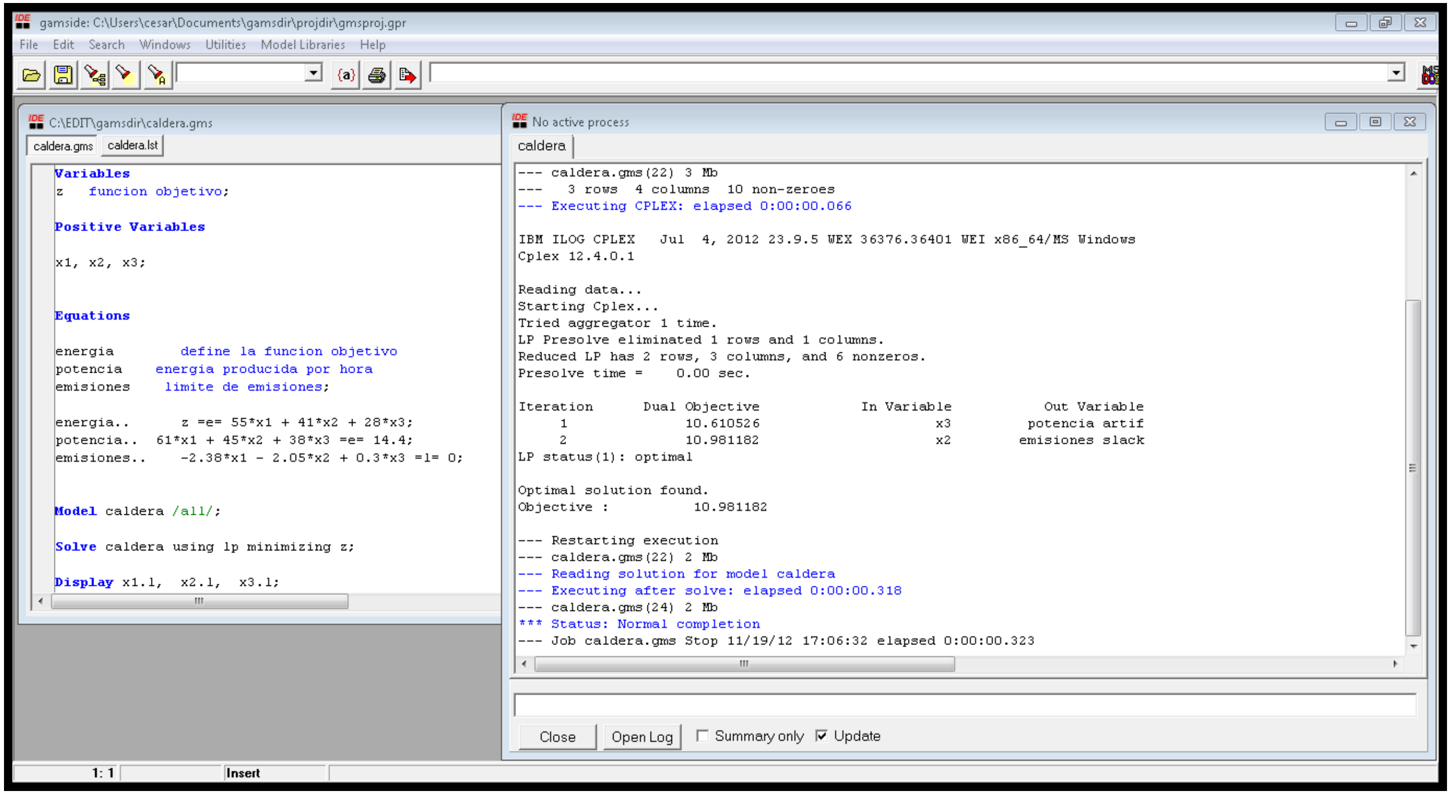
Values of the derivative of $P_j(\tau)$ at τ_I can be precomputed for different orders of P

	τ_0	τ_1	τ_2	τ_3
$\dot{P}_0$	-9.000001008080126	-4.139388773624379	1.739387967160278	-3.000000252020032
$\dot{P}_1$	10.048810106494384	3.224746191683931	-3.567840077120938	5.531972415060629
$\dot{P}_2$	-1.382142403745367	1.167839841902244	0.775254648382856	-7.531972331053937
$\dot{P}_3$	0.333333305331110	-0.253197259961796	1.053197461577804	5.000000168013340

P= 2, Radau

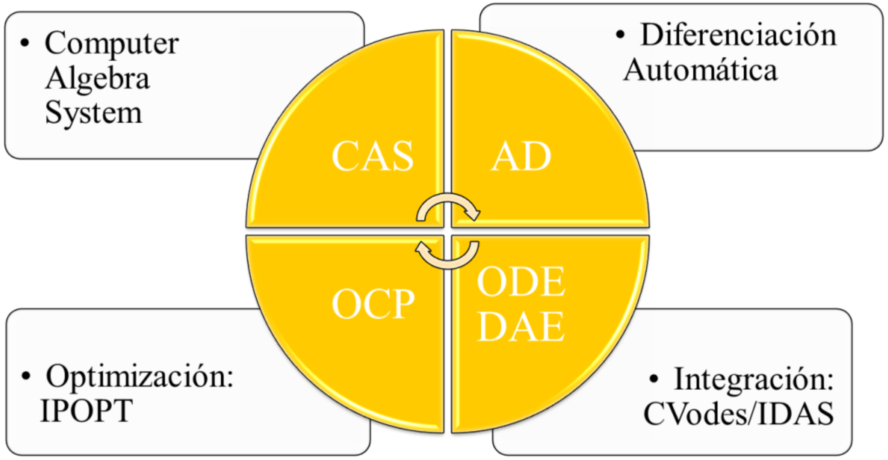
	τ_0	τ_1	τ_2
$\dot{P}_0$	-4	-2	2
$\dot{P}_1$	4.5	1.5	-4.5
$\dot{P}_2$	-0.5	0.5	2.5

Software: GAMS



Entornos de modelado y optimización como GAMS, AIMMS, XPRESS, Gurobi,… pueden usarse tras la discretización

Software



Solución eficiente de problemas de gran escala

Pero no soporta:

- Discontinuidades
- Optimización mixta-entera

Problemas de memoria

Entorno pobre de modelado

Computational Infrastructure for Operations Research (COIN-OR) Open source codes

Sensibilidades paramétricas

CasADi es un entorno simbólico para optimización numérica que facilita la discretización e implementa diferenciación automática (gradientes y Hesianos).

Genera código C e implementa interfaces a códigos DAE y de optimización como SUNDIALS, IPOPT etc.

Se gestiona desde una interfaz con Python

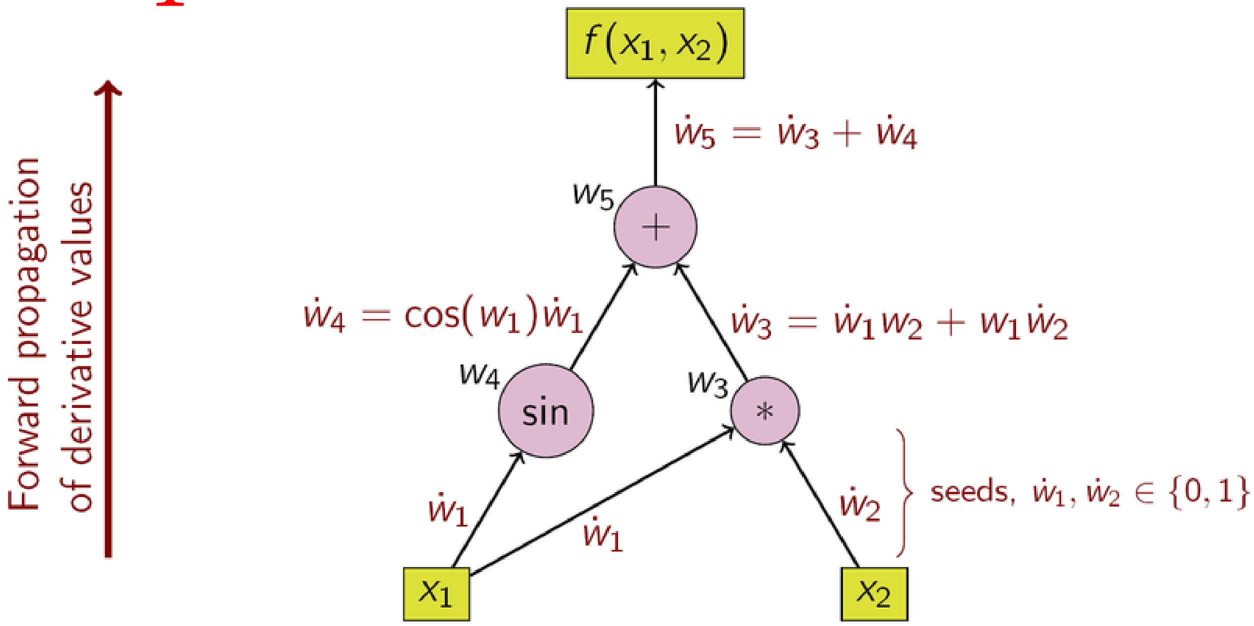
Diferenciación Automática

Ejemplo::

$$f = x_1 x_2 + \sin(x_1)$$

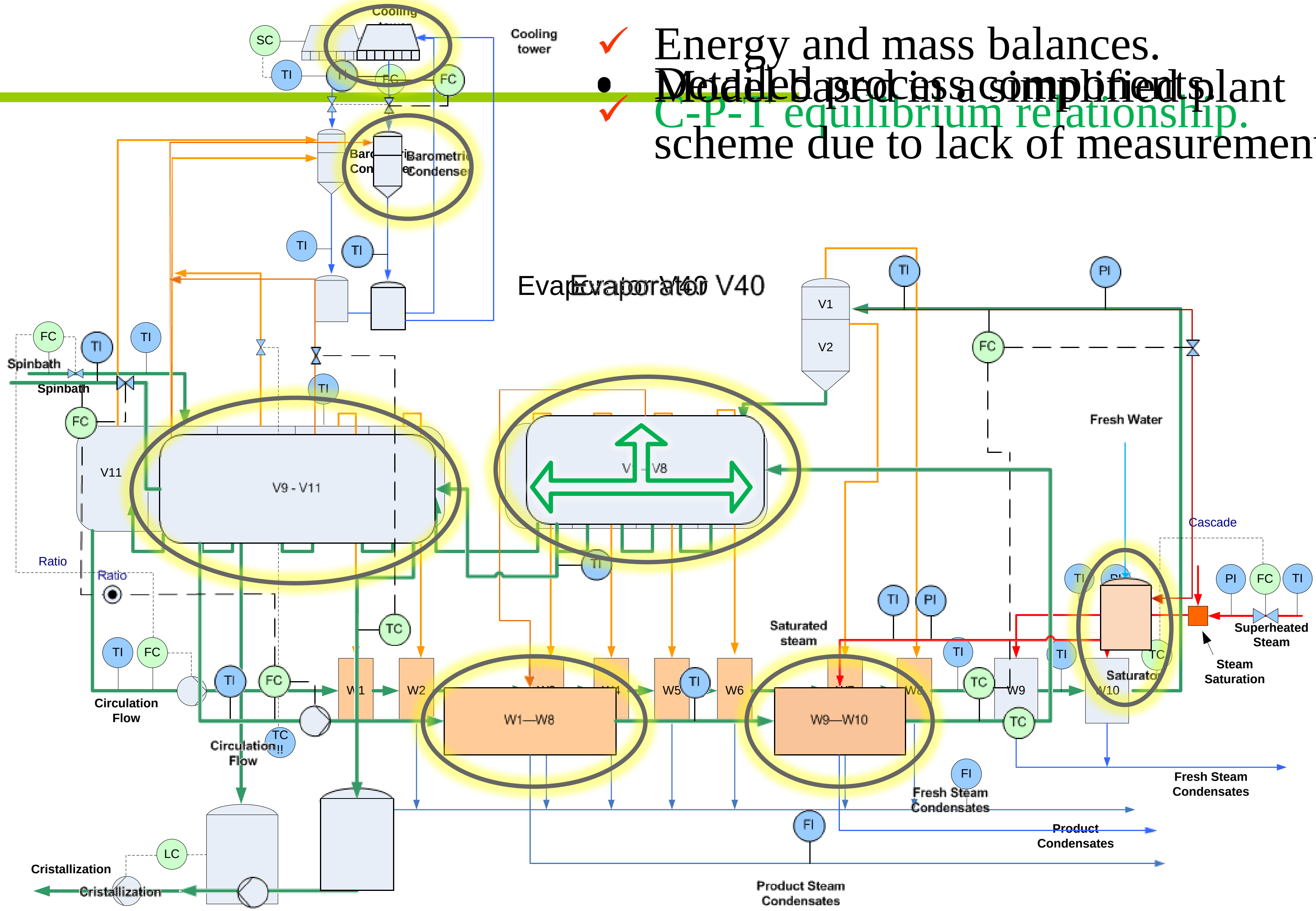
$$¿\frac{\partial f}{\partial x_1}?$$

Assignation	Derivatives
$w_1 = x_1$	$w_1' = 1\ (seed)$
$w_2 = x_2$	$w_2' = 0\ (seed)$
$w_3 = w_1 w_2$	$w_3' = w_1' w_2 + w_1 w_2' = x_2$
$w_4 = \sin(w_1)$	$w_4' = \cos(w_1) w_1' = \cos(x_1)$
$w_5 = w_3 + w_4$	$w_5' = w_3' + w_4' = x_2 + \cos(x_1)$



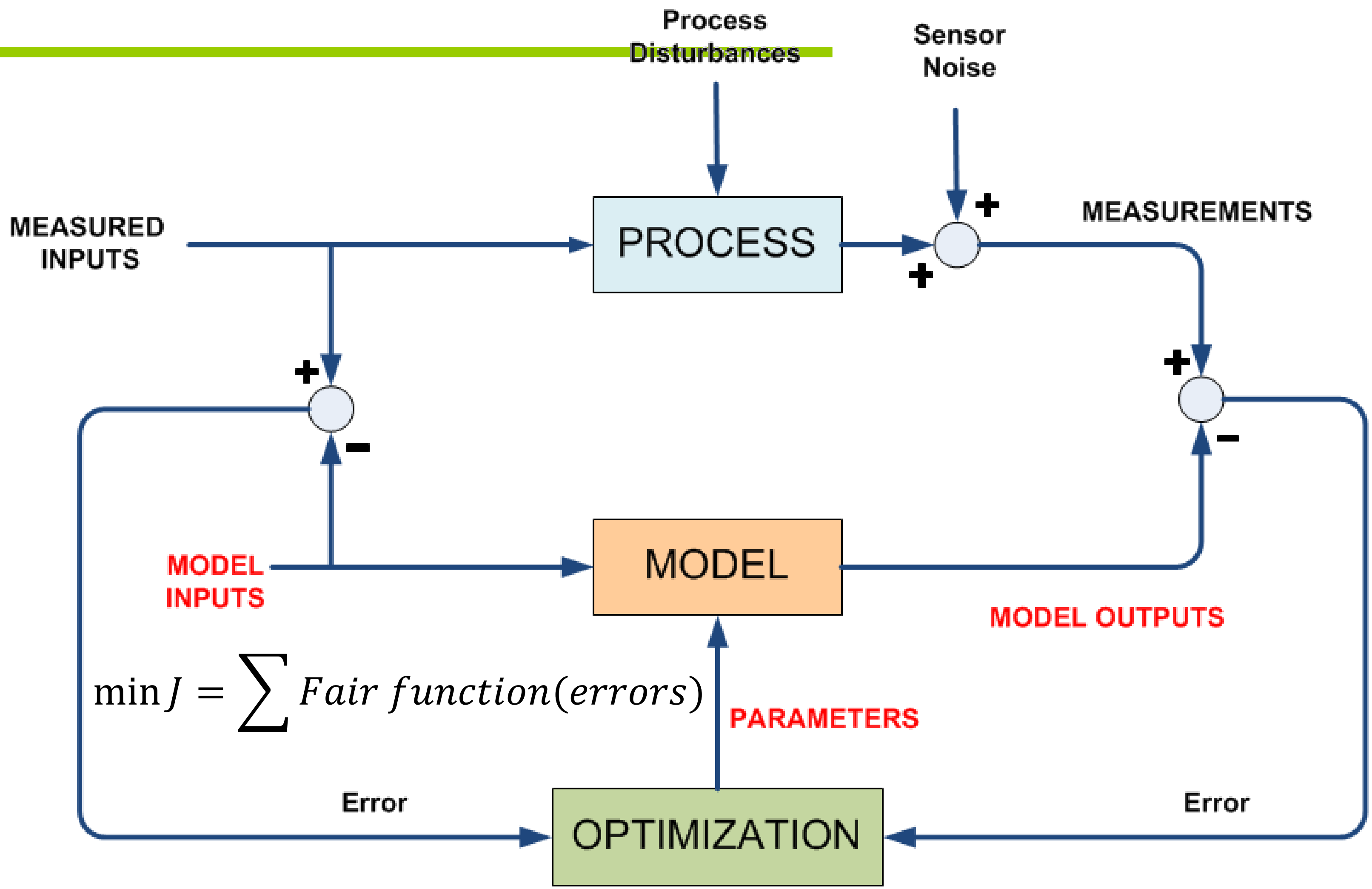
Evaporación, papelera

- ✓ Energy and mass balances.
- Model based in a simplified plant scheme due to lack of measurements.
- Detailed process components.
- ✓ C-P-T equilibrium relationship.

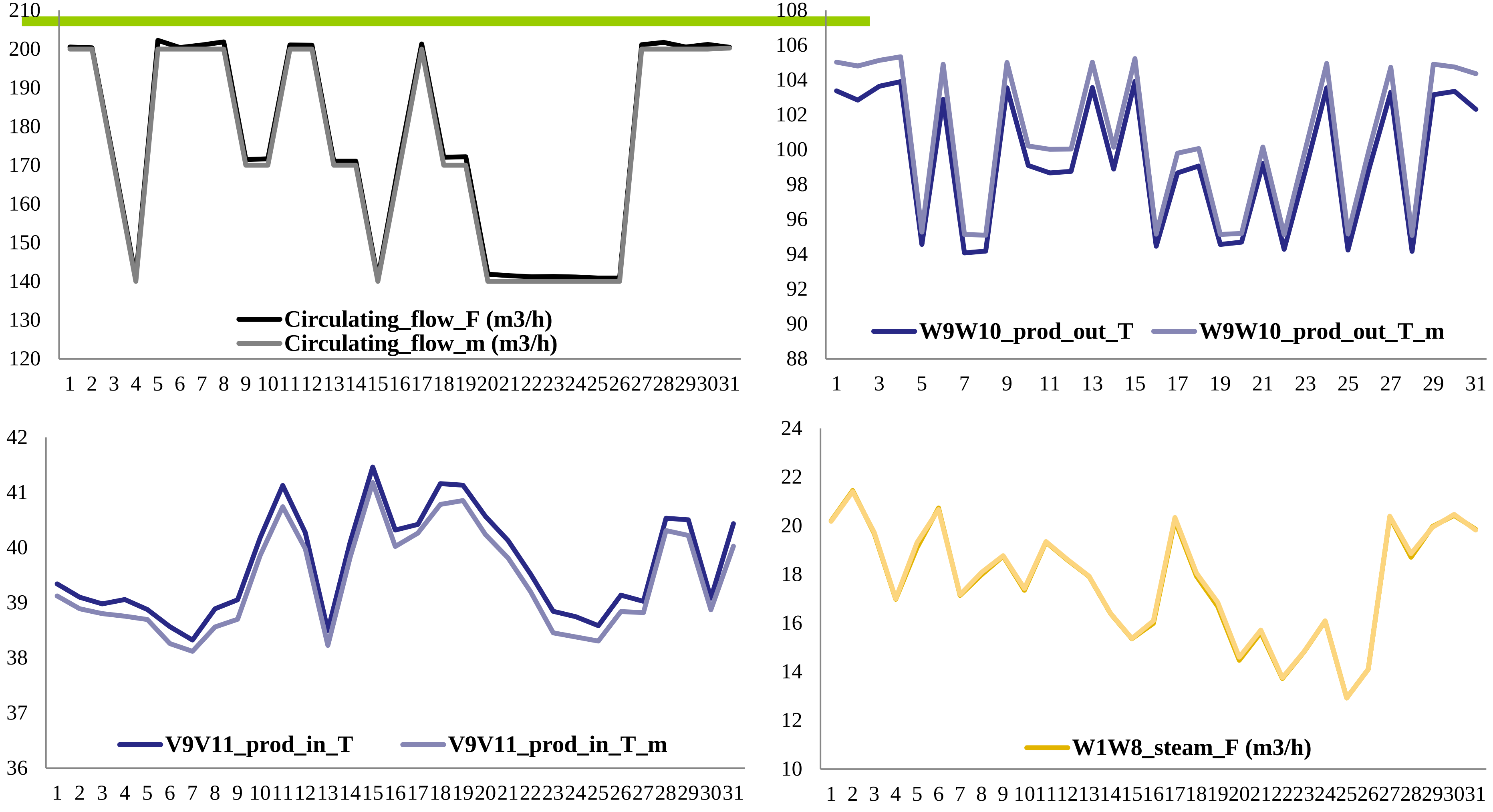


Objetivo: Operar la planta cono consumo específico de vapor mínimo

Data reconciliation



Data reconciliation



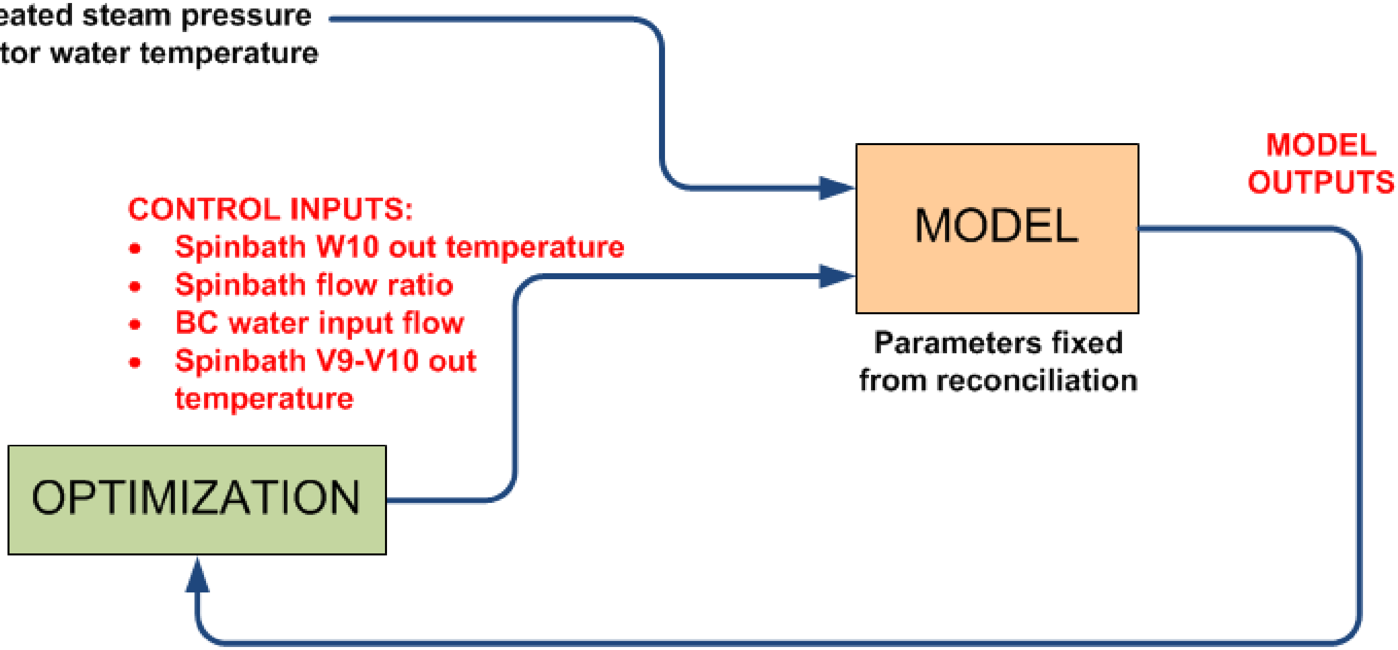
Measurement fitness

RTO

$$\min \quad J = Specific\ steam\ consumption$$
$$\text{s.t.} \quad Evaporation\ rate \geq target$$
$$Fixed\ cooling\ capacity\ in\ the\ tower$$

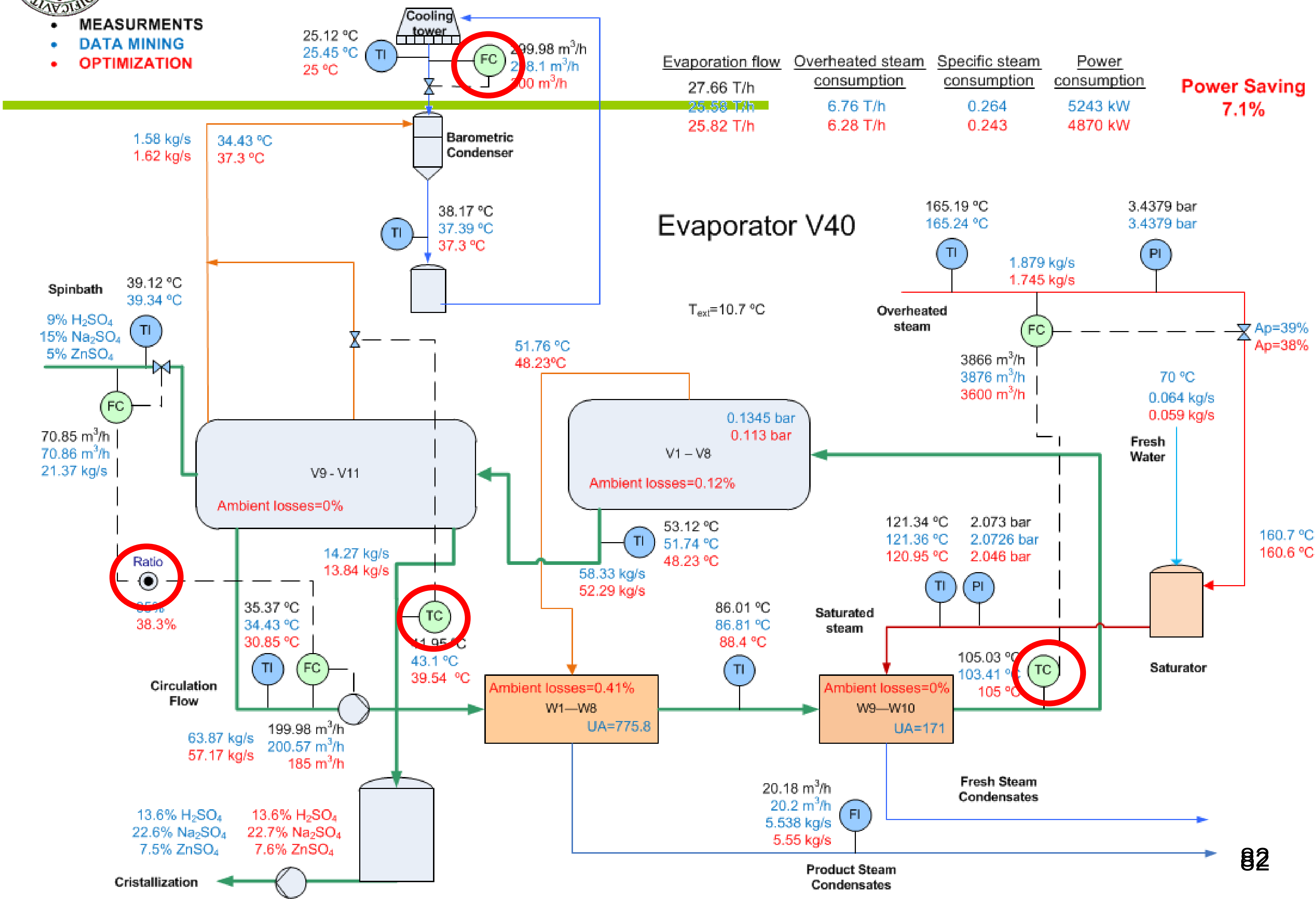
EXTERNAL INPUTS:

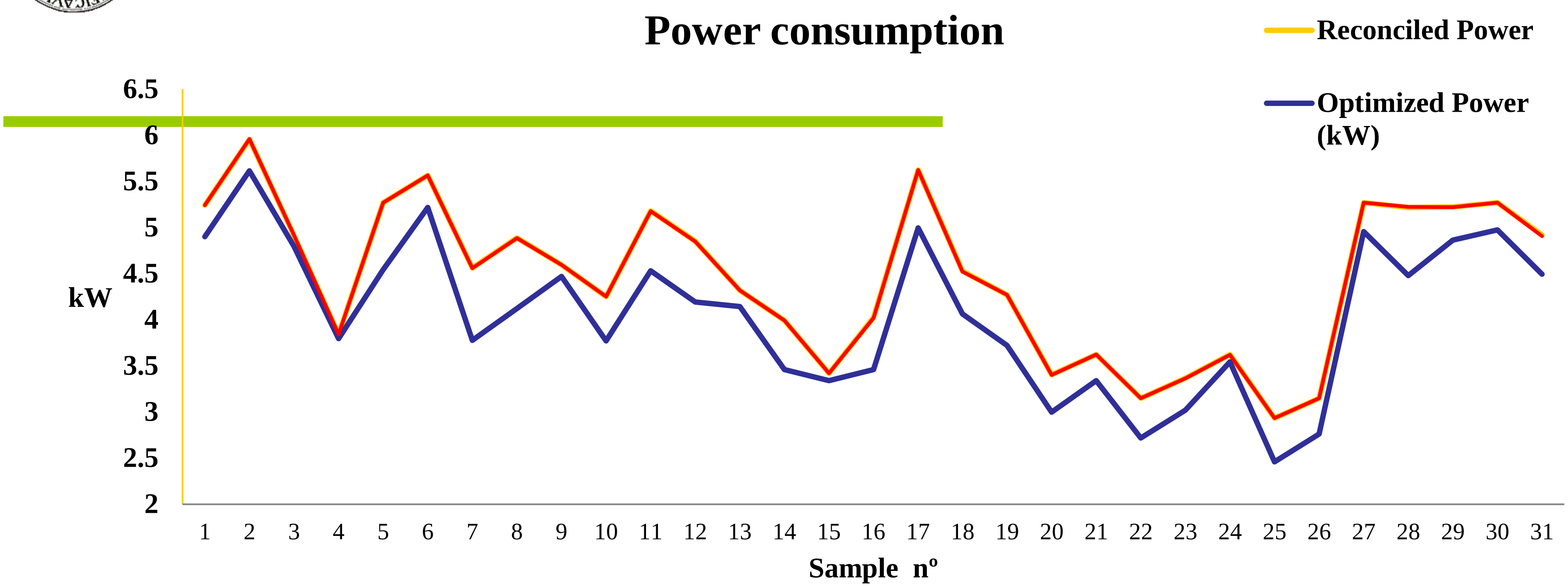
- Spinbath input flow
- Spinbath input temperature
- Overheated steam temperature
- Overheated steam pressure
- Saturator water temperature

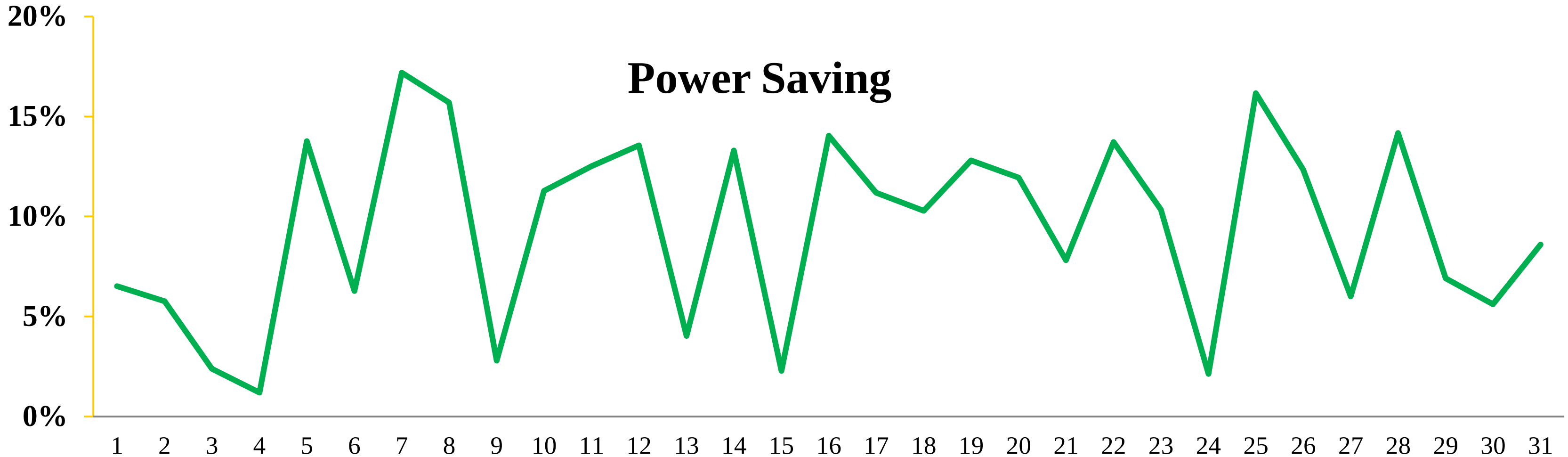


RTO

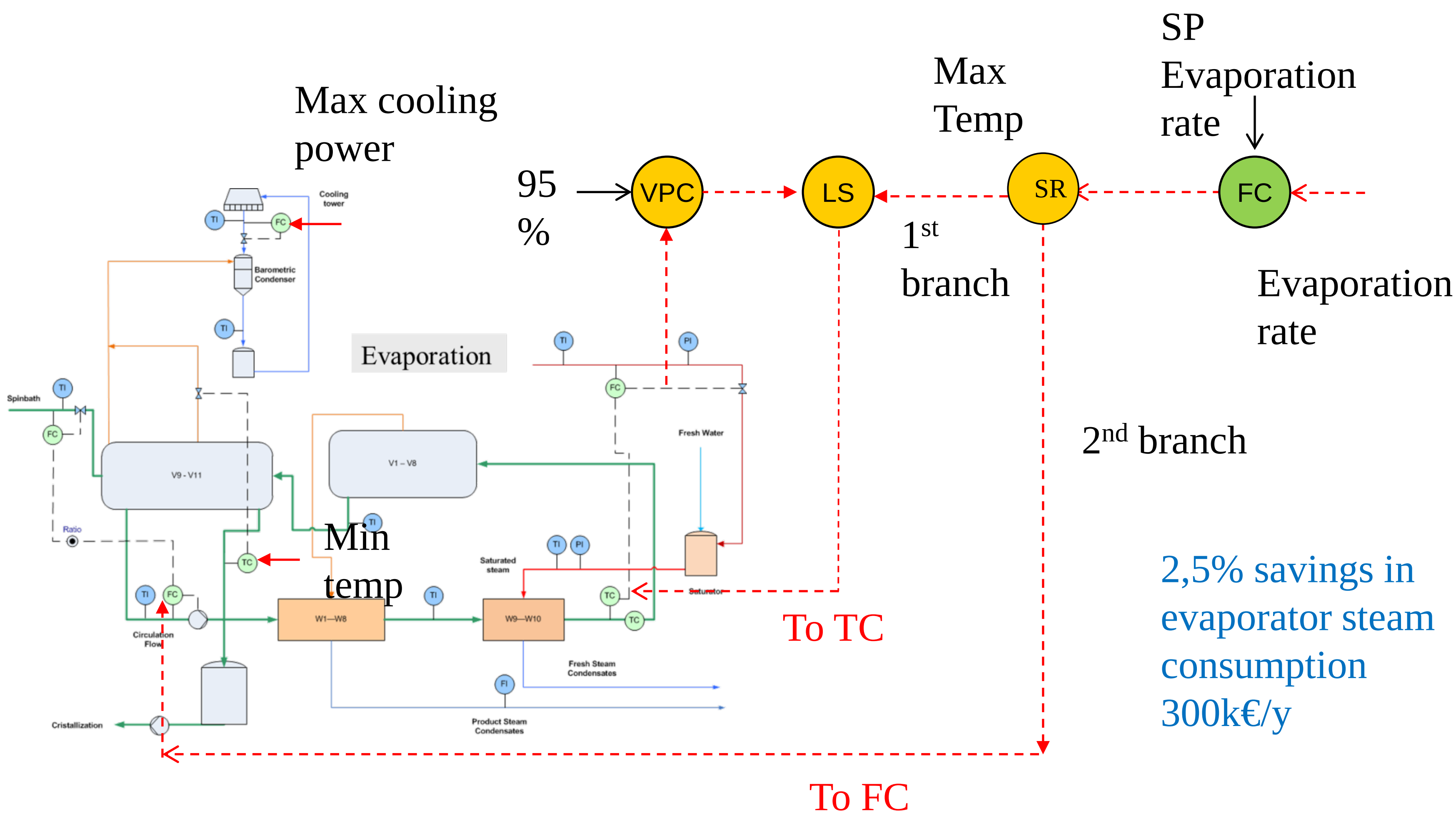
- **MEASURMENTS**
- **DATA MINING**
- **OPTIMIZATION**




Power consumption
Reconciled Power
Optimized Power (kW)
kW
6.5
6
5.5
5
4.5
4
3.5
3
2.5
2
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Sample nº

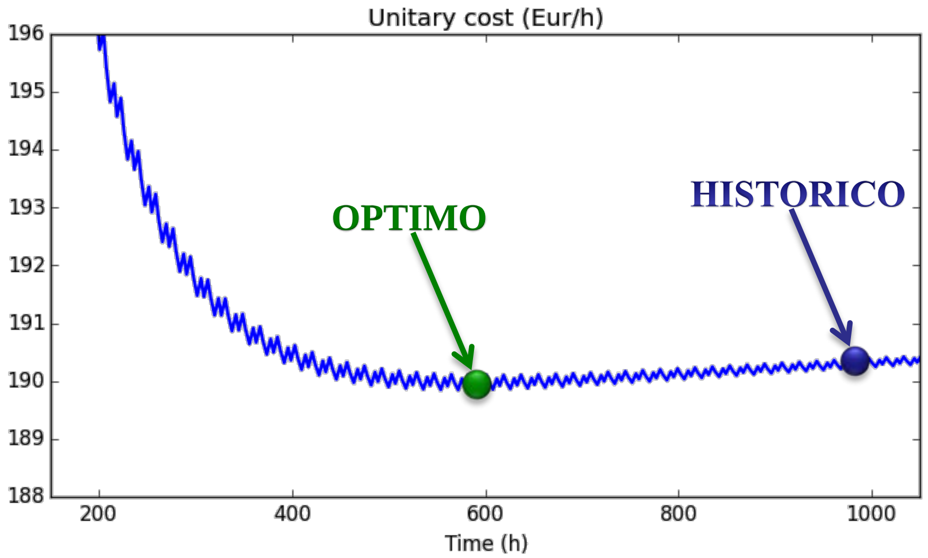

Power Saving
20%
15%
10%
5%
0%
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

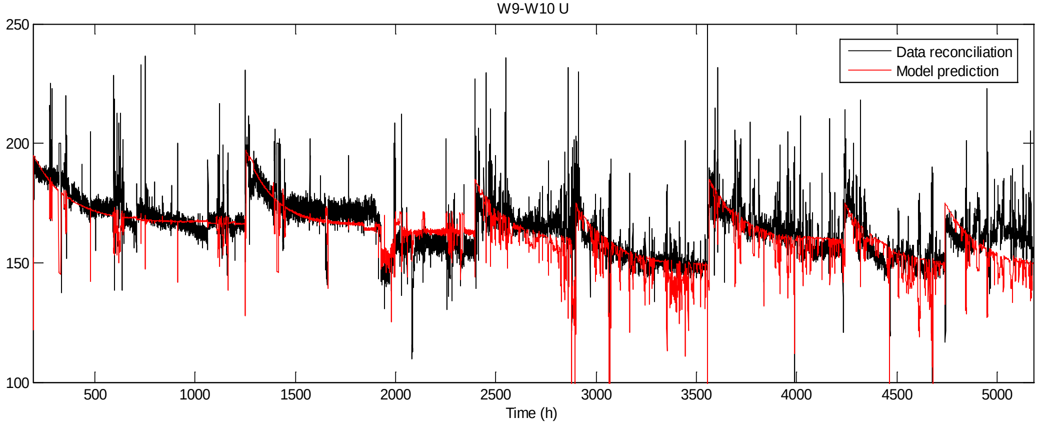
Implementación como sistema de control



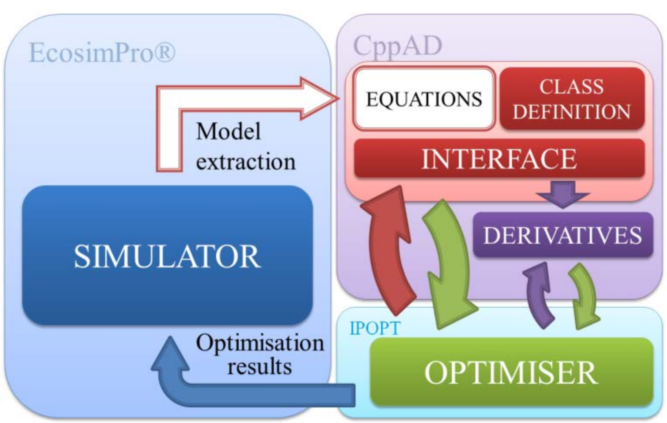
2,5% savings in evaporator steam consumption 300k€/y

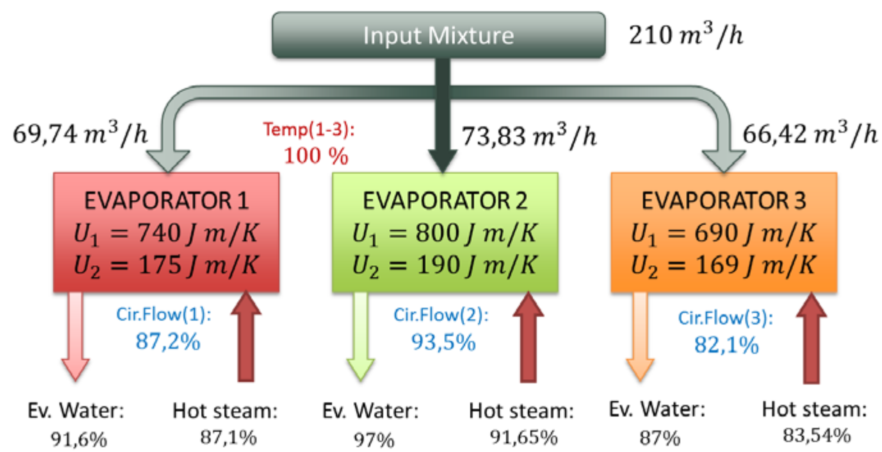
Ensuciamiento / Limpiezas



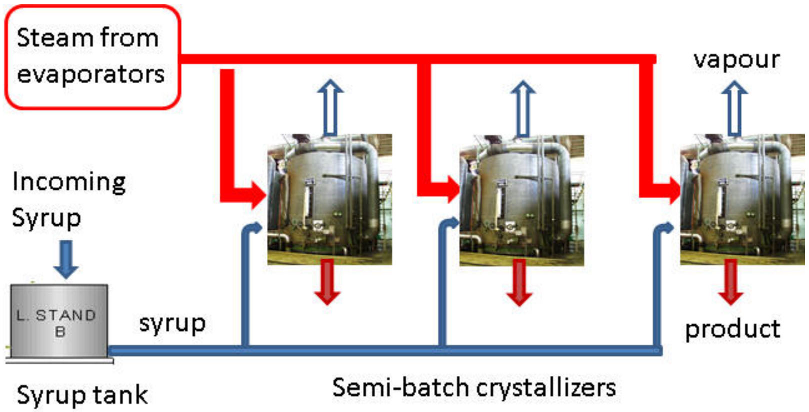


Distribución óptima de cargas





Problemas abiertos: Sistemas híbridos /Estructura variable



Scheduling + control óptimo local de los cristalizadores + recursos compartidos + optimización energética / producción

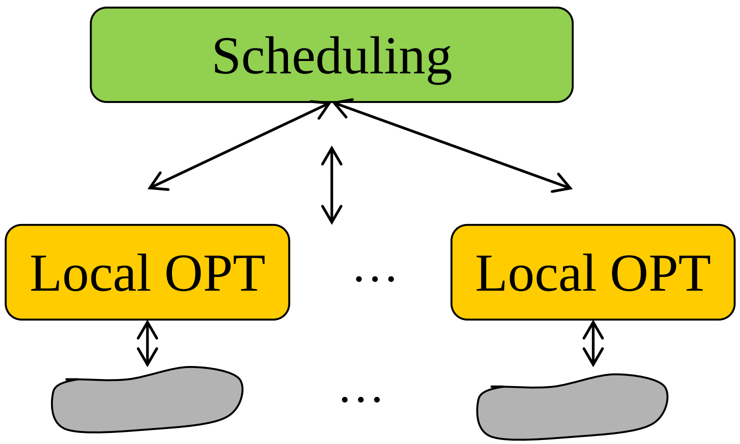
Discontinuidades en las derivadas

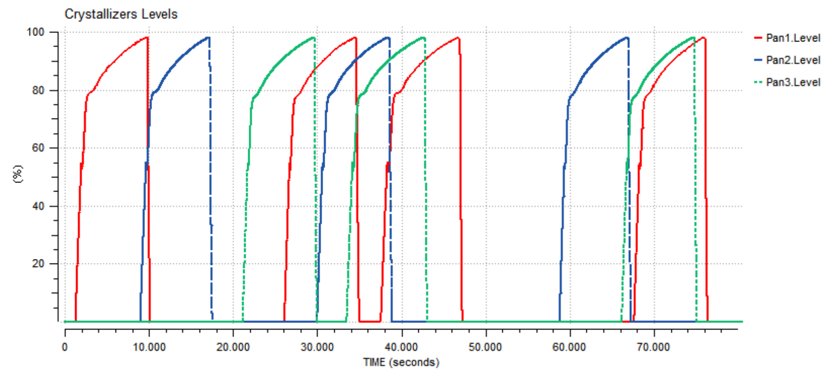
En algún caso se pueden transformar en NLP

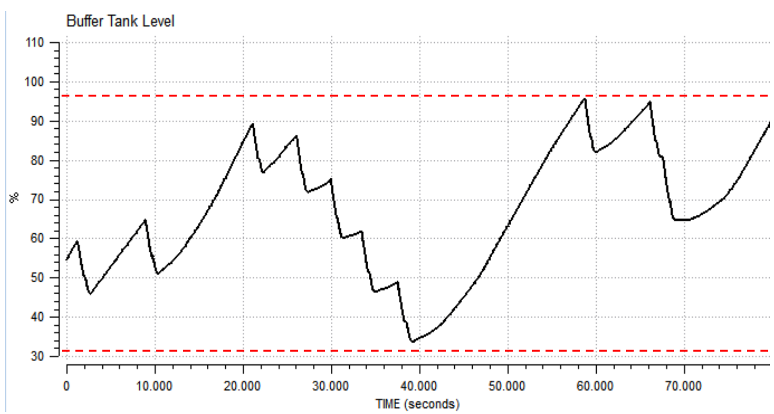
Formulación como problemas MINLP

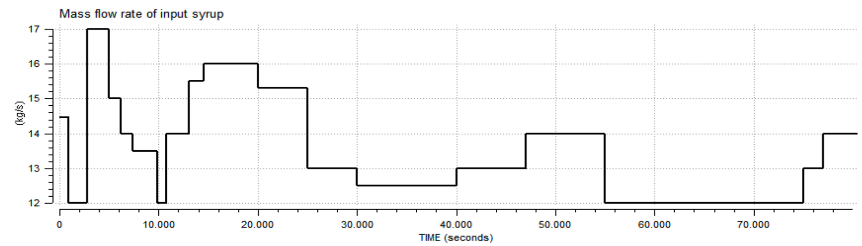
Descomposición jerárquica en varios tipos de problemas: Scheduling, optimización local, etc

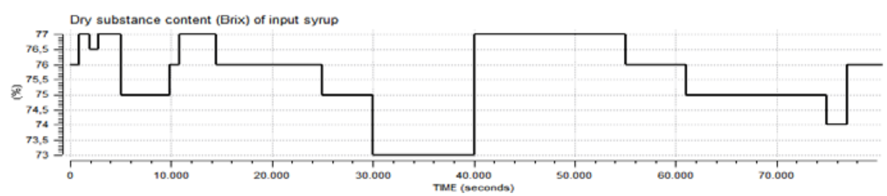
Explotar la estructura para descomponer el problema











Incertidumbre modelo / proceso

$$\min_{u} \quad J(u) = \int_0^T L(x,u)dt$$

$$F(\dot{x}, x, u) = 0$$

$$g(x,u) \le 0$$

La optimización calcula el óptimo del modelo, no del proceso

¿Qué ocurre si el modelo no es correcto?

¿Cómo se toman en cuenta las perturbaciones, incertidumbres, etc.?

Enfoques:

- ✓ **Actualizar el modelo**
- ✓ **Modificar la formulación de la optimización**
- ✓ **Optimización estocástica / robusta**

Actualización del modelo

Control / Optimización

Decisiones basadas en modelos

Modelo

Actualizar el modelo para ajustarlo a datos de planta

Medidas

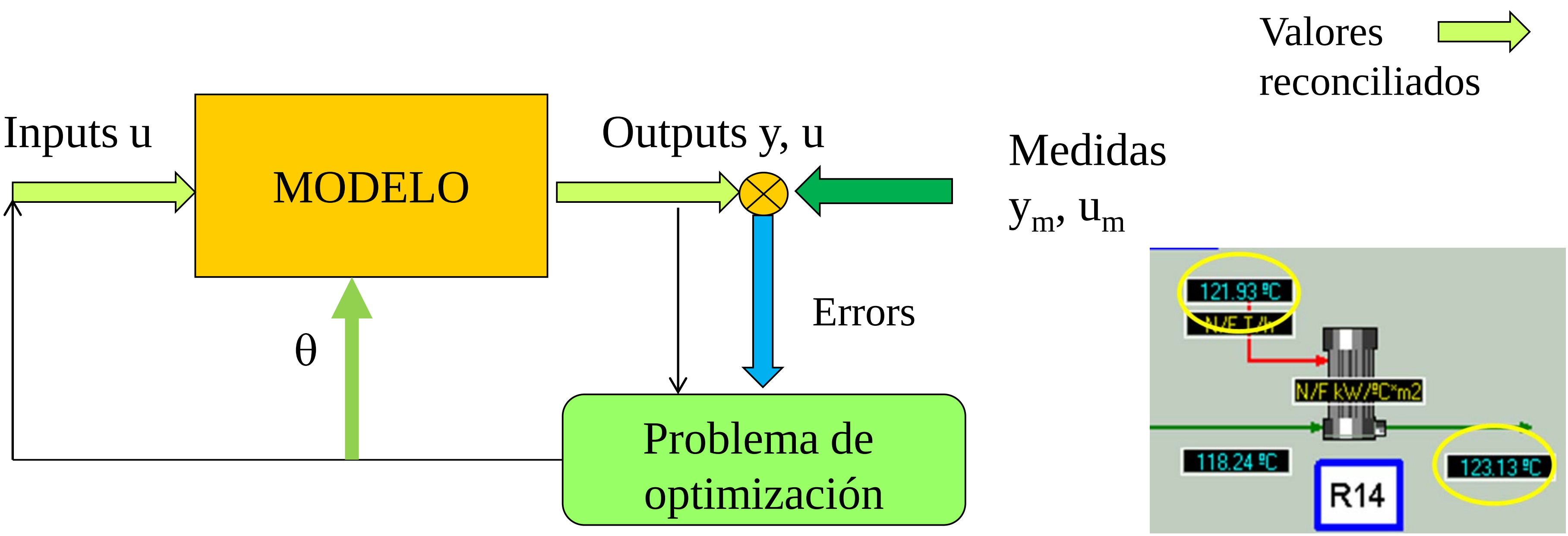
Reconciliación de datos y parámetros

- Algunas medidas no son consistentes o fiables
- Hay variables no medidas y parámetros desconocidos
- Se necesita un cierto grado de redundancia

$$\min_{u,y,\theta} \sum_{i=1}^{N_{measured}} \alpha_i \left(y_i - y_{m,i}\right)^2 + \beta_i \left(u_i - u_{m,i}\right)^2$$

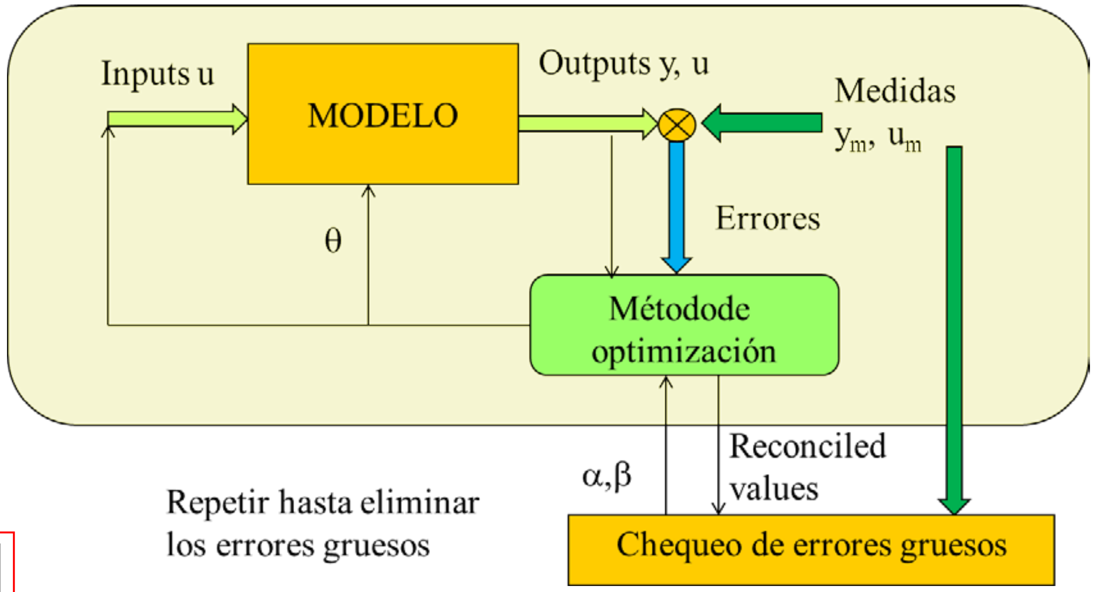
$$\frac{dx}{dt} = f(x,u,\theta) \qquad y = h(x,u,\theta)$$

$$g(x,y,u,\theta) \le 0$$



Errores gruesos

La reconciliación de datos se suele plantear en estado estacionario junto al RTO

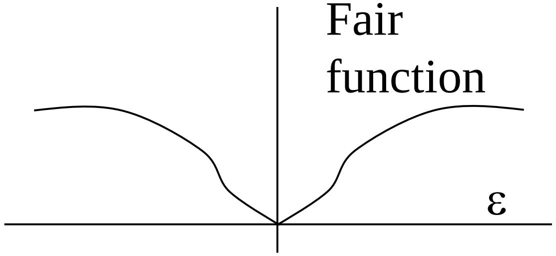


Estimadores robustos

$$\min_{x} \sum_{j \in M} c^2 \left[\frac{|\varepsilon_j|}{c} - \log\left(1 + \frac{|\varepsilon_j|}{c} \right) \right]$$

$$\varepsilon_j = \frac{x_j - x_{mj}}{\sigma}$$

$$f_i(x) = 0$$



Modifier adaptation

Control / Optimization

Modelo fijo

Incorporar los errores en un problema
de optimización modificado

Estimar los errores en las restricciones
y en los gradientes de J y g respecto al proceso

Se requieren medidas del proceso (y
dependiendo del método una cierta excitación)

Dimensionalidad

Transformarlo en un problema de control equivalente

Variables
de decisión
óptimas

SP

0

Controlador

Si son cero se
obtiene el óptimo
de la planta

Medidas o funciones
adecuadas

Ecuaciones NCO
con datos de planta

Estimación de las restricciones
activas o de los gradientes del costo
y las restricciones de la planta

NCO tracking
Self optimizing
Extremum
seeking,…

Dimensionalidad

Optimización estocástica

$$\min_{\mathbf{u}} J(\mathbf{x},\mathbf{u},\xi)$$

$$\frac{d\,\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\mathbf{u},\xi)$$

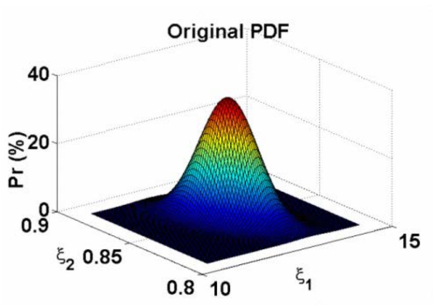
$$\mathbf{h}(\mathbf{x},\mathbf{u},\xi) = \mathbf{0}$$

$$\mathbf{g}(\mathbf{x},\mathbf{u},\xi) \leq \mathbf{0}$$

ξ Variable estocástica con una distribución de probabilidad

Optimización estocástica

Algunas de las variables son de naturaleza estocástica



- ✓ Chance constraints
- ✓ Escenarios
- ✓ Optimización de dos etapas

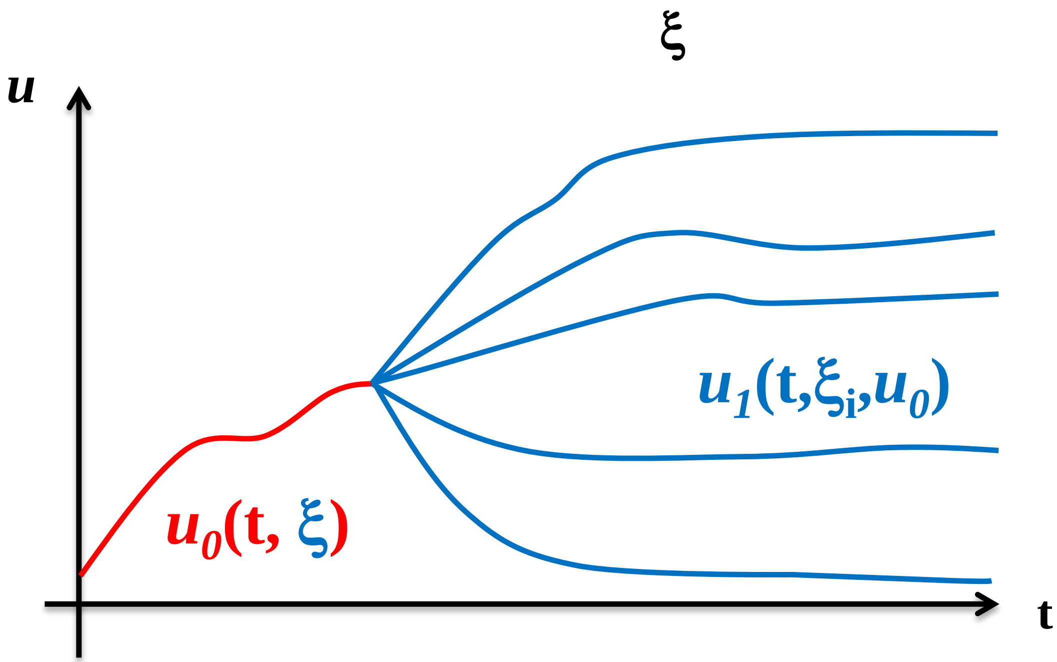
….

Optimización robusta

$$\underset{u}{\text{Min}}\ \underset{\xi}{\text{Max}}\ J(u,\xi)$$

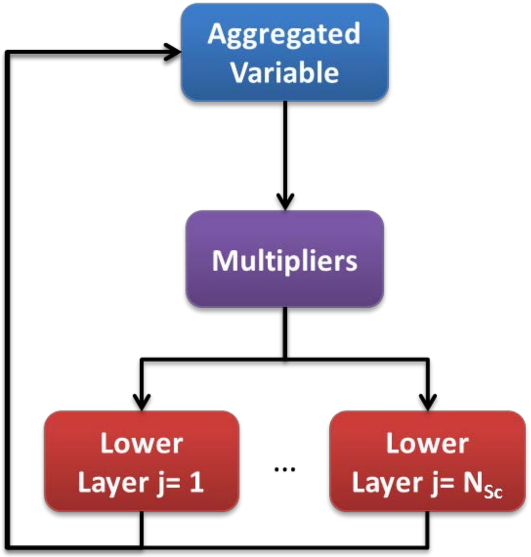
Solución del peor caso

Optimización estocástica multietapa



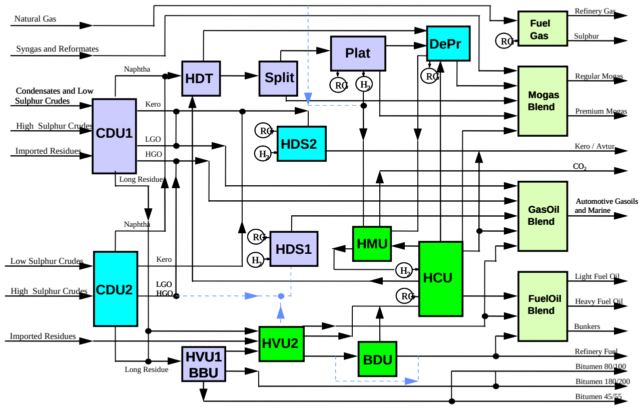
Se consideran una serie de escenarios

Enfoques centralizados / Lagrangiano aumentado



Tras un tiempo, la nueva información disponible permite determinar la incertidumbre ξ y, por tanto, aplicar un control específico u_1 para esa incertidumbre en la segunda etapa, dando lugar a soluciones menos conservadoras. u_0

¿Calculo distribuido? Paralelización



Jerárquicos

Coordinador

Opt.1 Opt.2 Opt.n

Proceso 1 Proceso 2 Proceso n

División funcional

Ajuste de consignas, tareas

Coordinación de precios

Mercado

Opt. 1 Opt.2 Opt.n

Proceso 1 Proceso 2 Proceso n

Ajuste de las J_i(precio)

Distribuidos

Opt.2

Opt.1 Opt.n

Proceso 1 Proceso 2 Proceso n

Intercambio de información

Gracias por vuestra atención