Distributed Optimization II

Mathias Hudoba de Badyn

Advanced Topics in Control
May 30, 2022

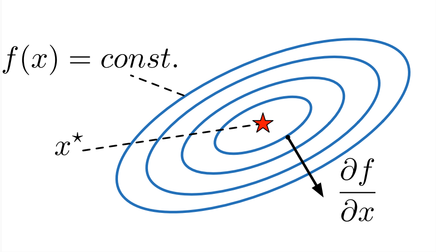
recap:
convex optimization

Unconstrained optimization

- **unconstrained optimization**:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\, f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and convex



unconstrained optimization

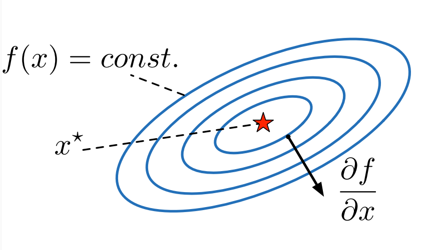
Unconstrained optimization

- **unconstrained optimization**:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\, f(x)$$

  where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and convex

- **gradient flow**: $\dot{x} = -K \frac{\partial f(x)}{\partial x}$ converges to minimizer $x^\star$



unconstrained optimization

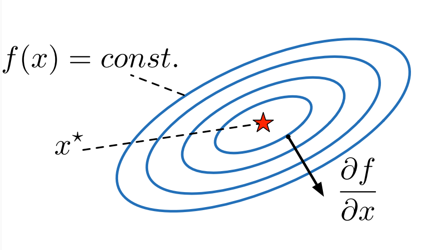
Unconstrained optimization

- **unconstrained optimization**:

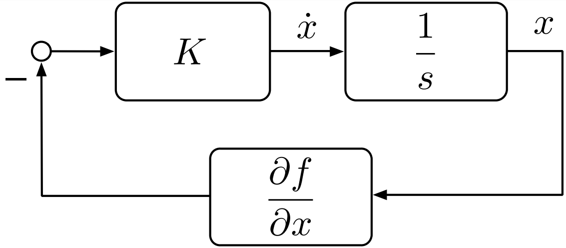
$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\, f(x)$$

  where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and convex

- **gradient flow**: $\dot{x} = -K \frac{\partial f(x)}{\partial x}$ converges to minimizer $x^\star$



unconstrained optimization



block-diagram of negative gradient flow

Constrained optimization

linearly-constrained optimization:

$$\begin{aligned} \mathrm{minimize}_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \end{aligned}$$

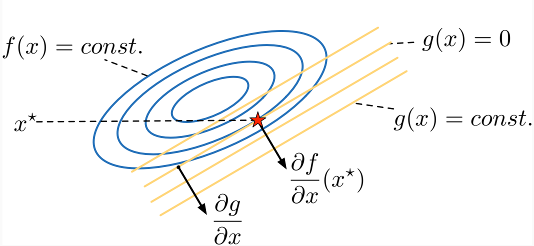
where f is cont. diff. & convex and $g(x) = Ax - b$ is linear-affine.

Constrained optimization

linearly-constrained optimization:

$$\begin{aligned} &\text{minimize}_{x\in\mathbb{R}^n} && f(x) \\ &\text{s.t.} && g(x) = \mathbf{0} \end{aligned}$$

where f is cont. diff. & convex and $g(x) = Ax - b$ is linear-affine.

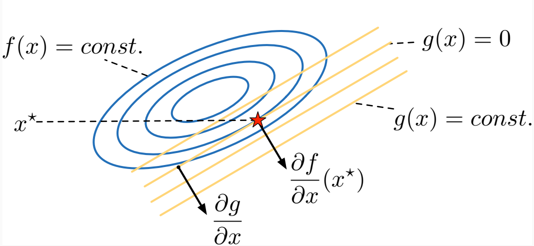


Constrained optimization

linearly-constrained optimization:

$$\begin{array}{ll} \text{minimize}_{x\in\mathbb{R}^n} & f(x) \\ \text{s.t.} & g(x) = \mathbf{0} \end{array}$$

where f is cont. diff. & convex and $g(x) = Ax - b$ is linear-affine.



[⇒] minimizer $x^\star$ and multiplier $\lambda^\star$ satisfy **KKT conditions**:

(i) constraint equation: $\mathbf{0} = g(x) = Ax - b$

(ii) tangency condition: $\mathbf{0} = \frac{\partial f(x)}{\partial x} + \frac{\partial g(x)}{\partial x}^\top \lambda$

- linearly-constrained optimization: $\mathrm{minimize}_{x \in \mathbb{R}^n} f(x) \text{ s.t. } g(x) = 0$

  where f is cont. diff. & convex and $g(x) = Ax - b$ is linear-affine

- linearly-constrained optimization: $\operatorname{minimize}_{x \in \mathbb{R}^n} f(x)$ s.t. $g(x) = 0$

  where f is cont. diff. & convex and $g(x) = Ax - b$ is linear-affine

- Lagrangian: $\mathcal{L}(x, \lambda) = f(x) + \lambda^\top g(x)$ is convex in x and concave in λ

- **linearly-constrained optimization:** $\mathrm{minimize}_{x \in \mathbb{R}^n} f(x)$ s.t. $g(x) = \mathbf{0}$

  where f is cont. diff. & convex and $g(x) = Ax - b$ is linear-affine

- **Lagrangian:** $\mathcal{L}(x, \lambda) = f(x) + \lambda^\top g(x)$ is convex in x and concave in λ

$\Rightarrow$ $\mathcal{L}(x, \lambda)$ has **saddle points**: $\mathcal{L}(x^\star, \lambda) \leq \mathcal{L}(x^\star, \lambda^\star) \leq \mathcal{L}(x, \lambda^\star)$

- **linearly-constrained optimization:** $\mathrm{minimize}_{x \in \mathbb{R}^n} f(x)$ s.t. $g(x) = \mathbf{0}$

  where f is cont. diff. & convex and $g(x) = Ax - b$ is linear-affine

- **Lagrangian:** $\mathcal{L}(x, \lambda) = f(x) + \lambda^\top g(x)$ is convex in x and concave in λ

$\Rightarrow$ $\mathcal{L}(x, \lambda)$ has **saddle points**: $\mathcal{L}(x^\star, \lambda) \leq \mathcal{L}(x^\star, \lambda^\star) \leq \mathcal{L}(x, \lambda^\star)$

$\Rightarrow$ saddle points = { minimizer $x^\star$ and multiplier $\lambda^\star$ } from KKT:

$$(i)\ \text{constraint equation: } \mathbf{0} = \frac{\partial \mathcal{L}}{\partial \lambda} = g(x) = Ax - b$$

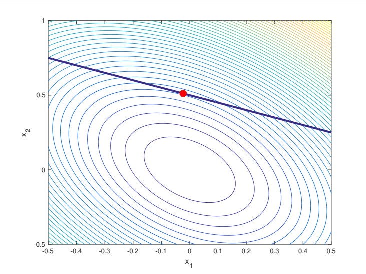
$$(ii)\ \text{tangency condition: } \mathbf{0} = \frac{\partial \mathcal{L}}{\partial x} = \frac{\partial f(x)}{\partial x} + \frac{\partial g(x)}{\partial x}^\top \lambda$$

Example for linear quadratic (LQ) case

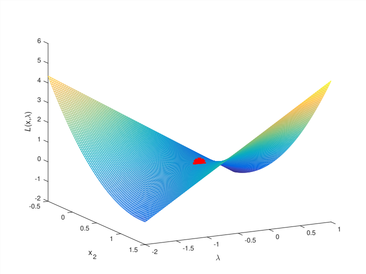
$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\ f(x) = \frac{1}{2}(x - \underline{x})^\top P (x - \underline{x}) \ \ \text{subject to} \ \ g(x) = Ax - b = 0$$

data: $P = \begin{bmatrix} 2.6 & 0.8 \\ 0.8 & 1.4 \end{bmatrix}$, $\underline{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $A = \begin{bmatrix} 1 & 2 \end{bmatrix}$, $b = 1$

facts: P is positive definite with eigenvalues $\{1, 3\}$, optimizer $x^\star = [-0.0233\ 0.5116]^\top$, and optimal multiplier $\lambda^\star = -0.3488$



geometry of the LQ problem



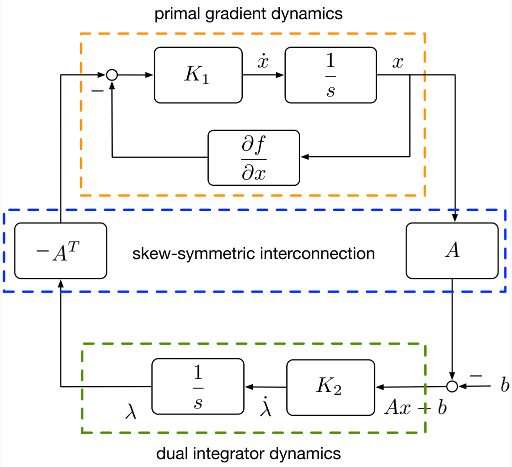
Lagrangian saddle point

Block-diagram of primal-dual saddle-point flow

Lagrangian $\mathcal{L}(x, \lambda) = f(x) + \lambda^\top g(x) \;\Rightarrow$ $\quad \dot{x} = -K_1 \frac{\partial \mathcal{L}}{\partial x} \;\; , \;\; \dot{\lambda} = +K_2 \frac{\partial \mathcal{L}}{\partial \lambda}$

Block-diagram of primal-dual saddle-point flow

Lagrangian $\mathcal{L}(x,\lambda) = f(x) + \lambda^\top g(x) \;\Rightarrow$ $\quad \dot{x} = -K_1 \frac{\partial \mathcal{L}}{\partial x} \;,\quad \dot{\lambda} = +K_2 \frac{\partial \mathcal{L}}{\partial \lambda}$



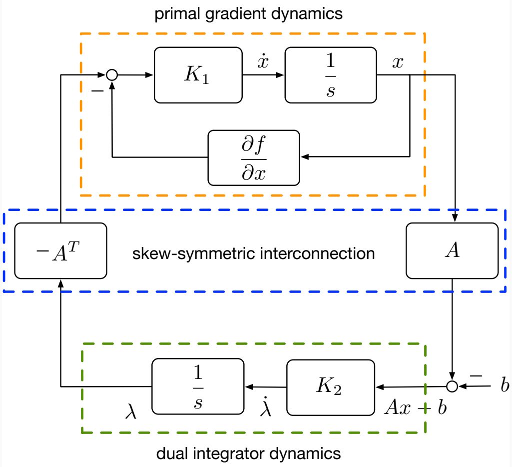
$$\dot{x} = -K_1 \frac{\partial f(x)}{\partial x} - K_1 A^\top \lambda$$

$$\dot{\lambda} = K_2 \,(Ax - b)$$

1. primal gradient descent
2. dual integral control penalizing constraint violation

Block-diagram of primal-dual saddle-point flow

Lagrangian $\mathcal{L}(x,\lambda) = f(x) + \lambda^\top g(x) \;\Rightarrow\;$ $\dot{x} = -K_1 \frac{\partial \mathcal{L}}{\partial x}$, $\dot{\lambda} = +K_2 \frac{\partial \mathcal{L}}{\partial \lambda}$



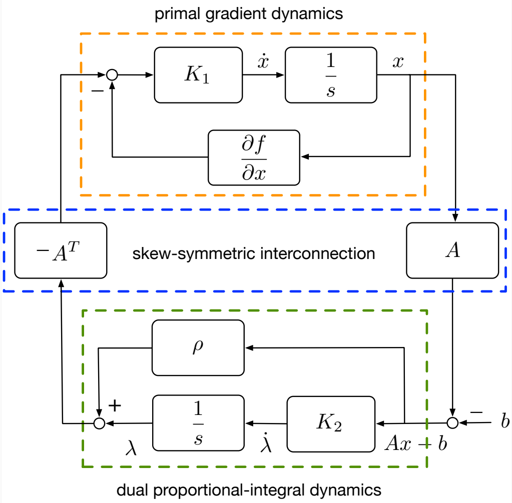
$$\dot{x} = -K_1 \frac{\partial f(x)}{\partial x} - K_1 A^\top \lambda$$

$$\dot{\lambda} = K_2\,(Ax - b)$$

1. primal gradient descent
2. dual integral control penalizing constraint violation
3. skew-symmetric interconnection

Block-diagram of augmented primal-dual saddle-point flow

for **augmented Lagrangian** $\mathcal{L}_{aug}(x,\lambda) = f(x) + \lambda^\top g(x) + \frac{\rho}{2}\|Ax - b\|^2$

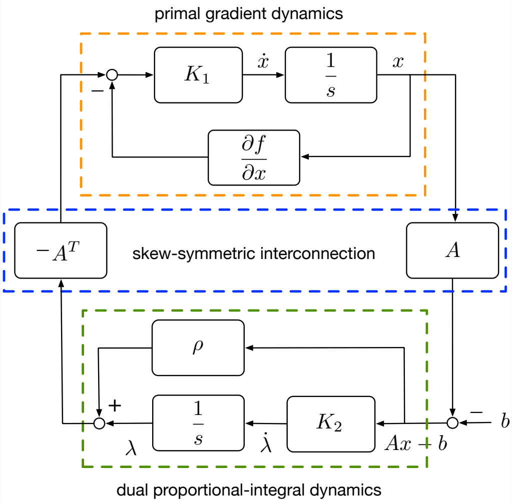


$$\dot{x} = -K_1 \frac{\partial f(x)}{\partial x} - K_1 A^\top \lambda$$
$$-\rho K_1 A^\top (Ax - b)$$
$$\dot{\lambda} = K_2 (Ax - b)$$

1. primal gradient descent

Block-diagram of augmented primal-dual saddle-point flow

for **augmented Lagrangian** $\mathcal{L}_{aug}(x,\lambda) = f(x) + \lambda^\top g(x) + \frac{\rho}{2}\|Ax - b\|^2$

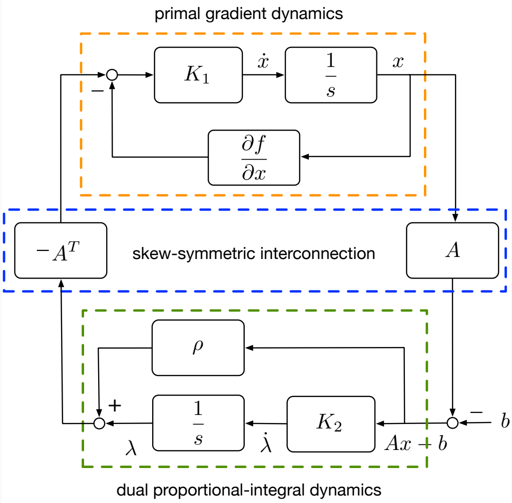


$$\dot{x} = -K_1 \frac{\partial f(x)}{\partial x} - K_1 A^\top \lambda$$
$$-\rho K_1 A^\top (Ax - b)$$
$$\dot{\lambda} = K_2\,(Ax - b)$$

1. primal gradient descent
2. dual proportional-integral control penalizing constraint violation

Block-diagram of augmented primal-dual saddle-point flow

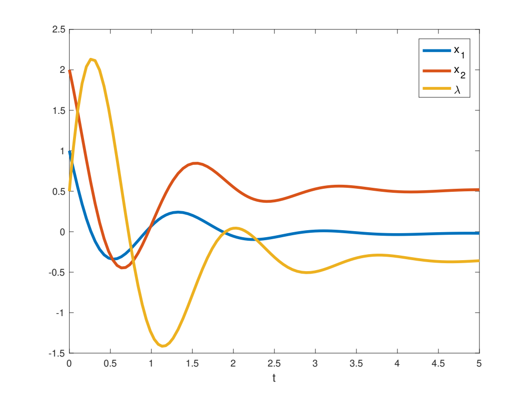
for **augmented Lagrangian** $\mathcal{L}_{aug}(x, \lambda) = f(x) + \lambda^\top g(x) + \frac{\rho}{2}\|Ax - b\|^2$



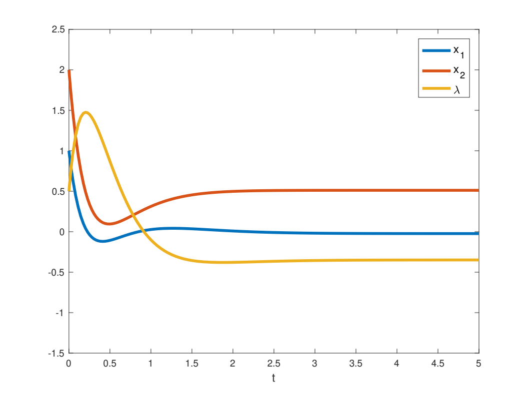
$$\dot{x} = -K_1 \frac{\partial f(x)}{\partial x} - K_1 A^\top \lambda$$
$$-\rho K_1 A^\top (Ax - b)$$
$$\dot{\lambda} = K_2 (Ax - b)$$

1. primal gradient descent
2. dual proportional-integral control penalizing constraint violation
3. skew-symmetric interconnection

Standard and augmented saddle-point flow for LQ program



standard saddle-point flow



augmented saddle-point flow ($\rho = 1$)

augmentation induces (stricter) convexity & better performance (damping)

recap: distributed optimization

Setup in distributed optimization

1. **basic problem:** n distributed agents want to solve

$$\begin{array}{ll} \text{minimize}_{x \in \mathbb{R}^n} & \sum_{i=1}^{n} f_i(x_i) \\ \text{s.t.} & Lx = 0 \text{ or } E^T x = 0 \end{array}$$

f_i is cont. diff. & strictly convex

Setup in distributed optimization

1. **basic problem:** n distributed agents want to solve

$$\begin{array}{ll} \mathrm{minimize}_{x \in \mathbb{R}^n} & \sum_{i=1}^{n} f_i(x_i) \\ \text{s.t.} & Lx = 0 \text{ or } E^T x = 0 \end{array}$$

f_i is cont. diff. & strictly convex

2. **knowledge:** $f_i : \mathbb{R} \to \mathbb{R}$ is private cost function known only to agent i

Setup in distributed optimization

1. **basic problem:** n distributed agents want to solve

$$\begin{array}{ll} \text{minimize}_{x \in \mathbb{R}^n} & \sum_{i=1}^n f_i(x_i) \\ \text{s.t.} & Lx = 0 \text{ or } E^T x = 0 \end{array}$$

f_i is cont. diff. & strictly convex

2. **knowledge:** $f_i : \mathbb{R} \to \mathbb{R}$ is private cost function known only to agent i

$\Rightarrow$ since x is a *global* variable, the agents need to **coordinate**

Setup in distributed optimization

1. **basic problem:** n distributed agents want to solve

$$\begin{array}{ll} \text{minimize}_{x \in \mathbb{R}^n} & \sum_{i=1}^n f_i(x_i) \\ \text{s.t.} & Lx = 0 \text{ or } E^T x = 0 \end{array}$$

f_i is cont. diff. & strictly convex

2. **knowledge:** $f_i : \mathbb{R} \to \mathbb{R}$ is private cost function known only to agent i

$\Rightarrow$ since x is a *global* variable, the agents need to **coordinate**

3. **communication:** undirected and connected communication graph with n nodes (processors) and m edges (communication links)

Setup in distributed optimization

1. **basic problem:** n distributed agents want to solve

$$\begin{aligned} \text{minimize}_{x \in \mathbb{R}^n} \quad & \textstyle\sum_{i=1}^{n} f_i(x_i) \\ \text{s.t.} \quad & Lx = 0 \text{ or } E^T x = 0 \end{aligned}$$

f_i is cont. diff. & strictly convex

2. **knowledge:** $f_i : \mathbb{R} \to \mathbb{R}$ is private cost function known only to agent i

$\Rightarrow$ since x is a *global* variable, the agents need to **coordinate**

3. **communication:** undirected and connected communication graph with n nodes (processors) and m edges (communication links)

$\Rightarrow$ **key idea:** local copies of global variable x and consensus constraint

Distributed primal-dual saddle-point flow

The **saddle-point flow** associated with $\mathcal{L}(y,\lambda) = \tilde{f}(y) + \lambda^\top L y + \frac{1}{2} y^\top L y$ is

$$\dot{y} = -\frac{\partial \mathcal{L}(y,\lambda)}{\partial y} = -\frac{\partial \tilde{f}(y)}{\partial y} - Ly - L\lambda \qquad \dot{\lambda} = +\frac{\partial \mathcal{L}(y,\lambda)}{\partial \lambda} = +Ly$$

Distributed primal-dual saddle-point flow

The **saddle-point flow** associated with $\mathcal{L}(y, \lambda) = \tilde{f}(y) + \lambda^\top L y + \frac{1}{2} y^\top L y$ is

$$\dot{y} = -\frac{\partial \mathcal{L}(y, \lambda)}{\partial y} = -\frac{\partial \tilde{f}(y)}{\partial y} - Ly - L\lambda \qquad \dot{\lambda} = +\frac{\partial \mathcal{L}(y, \lambda)}{\partial \lambda} = +Ly$$

For processor i the saddle-point dynamics **read component-wise** as

$$\begin{aligned} \dot{y}_i &= -\frac{\partial \tilde{f}_i(y_i)}{\partial y_i} - \sum_{j=1}^{n} a_{ij}(y_i - y_j) - \sum_{j=1}^{n} a_{ij}(\lambda_i - \lambda_j) \\ \dot{\lambda}_i &= \sum_{j=1}^{n} a_{ij}(y_i - y_j)\,. \end{aligned}$$

Distributed primal-dual saddle-point flow

The **saddle-point flow** associated with $\mathcal{L}(y,\lambda) = \tilde{f}(y) + \lambda^\top Ly + \frac{1}{2}y^\top Ly$ is

$$\dot{y} = -\frac{\partial \mathcal{L}(y,\lambda)}{\partial y} = -\frac{\partial \tilde{f}(y)}{\partial y} - Ly - L\lambda \qquad \dot{\lambda} = +\frac{\partial \mathcal{L}(y,\lambda)}{\partial \lambda} = +Ly$$

For processor i the saddle-point dynamics **read component-wise** as

$$\begin{aligned} \dot{y}_i &= -\frac{\partial \tilde{f}_i(y_i)}{\partial y_i} - \sum\nolimits_{j=1}^{n} a_{ij}(y_i - y_j) - \sum\nolimits_{j=1}^{n} a_{ij}(\lambda_i - \lambda_j) \\ \dot{\lambda}_i &= \sum\nolimits_{j=1}^{n} a_{ij}(y_i - y_j)\,. \end{aligned}$$

Distributed primal-dual saddle-point flow

The **saddle-point flow** associated with $\mathcal{L}(y, \lambda) = \tilde{f}(y) + \lambda^\top L y + \frac{1}{2} y^\top L y$ is

$$\dot{y} = -\frac{\partial \mathcal{L}(y, \lambda)}{\partial y} = -\frac{\partial \tilde{f}(y)}{\partial y} - Ly - L\lambda \qquad \dot{\lambda} = +\frac{\partial \mathcal{L}(y, \lambda)}{\partial \lambda} = +Ly$$

For processor i the saddle-point dynamics **read component-wise** as

$$\begin{aligned} \dot{y}_i &= -\frac{\partial \tilde{f}_i(y_i)}{\partial y_i} - \sum\nolimits_{j=1}^{n} a_{ij}(y_i - y_j) - \sum\nolimits_{j=1}^{n} a_{ij}(\lambda_i - \lambda_j) \\ \dot{\lambda}_i &= \sum\nolimits_{j=1}^{n} a_{ij}(y_i - y_j)\,. \end{aligned}$$

Distributed primal-dual saddle-point flow

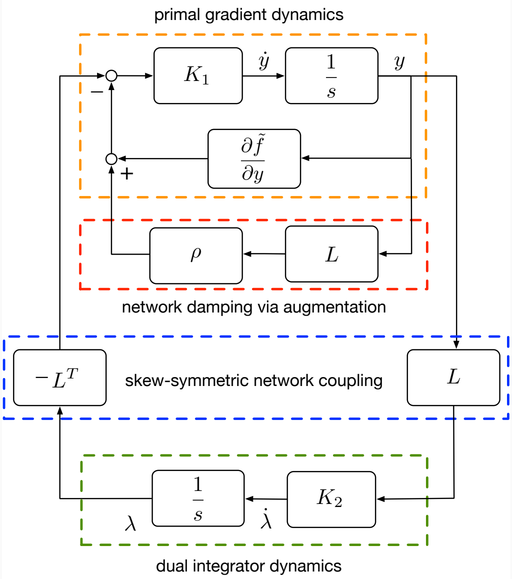
The **saddle-point flow** associated with $\mathcal{L}(y, \lambda) = \tilde{f}(y) + \lambda^\top L y + \frac{1}{2} y^\top L y$ is

$$\dot{y} = -\frac{\partial \mathcal{L}(y, \lambda)}{\partial y} = -\frac{\partial \tilde{f}(y)}{\partial y} - Ly - L\lambda \qquad \dot{\lambda} = +\frac{\partial \mathcal{L}(y, \lambda)}{\partial \lambda} = +Ly$$

For processor i the saddle-point dynamics **read component-wise** as

$$\begin{aligned} \dot{y}_i &= -\frac{\partial \tilde{f}_i(y_i)}{\partial y_i} - \sum\nolimits_{j=1}^{n} a_{ij}(y_i - y_j) - \sum\nolimits_{j=1}^{n} a_{ij}(\lambda_i - \lambda_j) \\ \dot{\lambda}_i &= \sum\nolimits_{j=1}^{n} a_{ij}(y_i - y_j)\,. \end{aligned}$$

Block-diagram of distributed primal-dual saddle-point flow

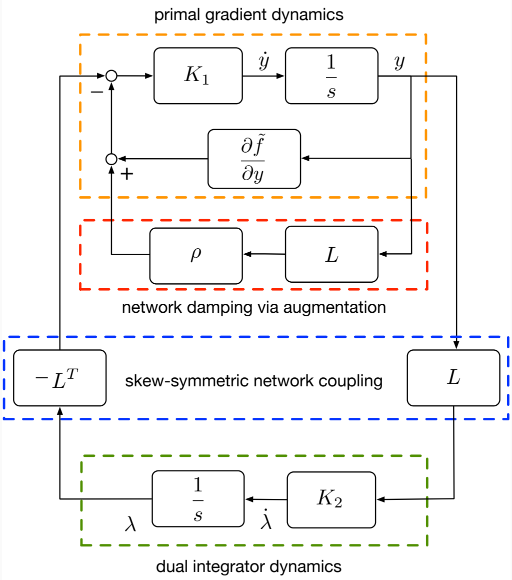


$$\dot{y} = -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - K_1 L \lambda - K_1 \rho L y$$

$$\dot{\lambda} = K_2 L y$$

1. primal gradient descent

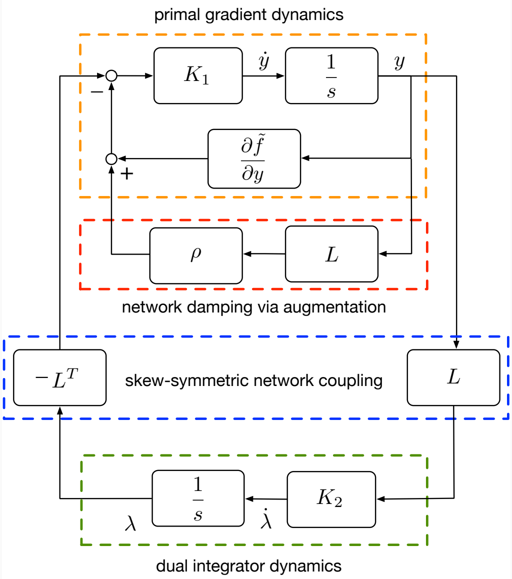
Block-diagram of distributed primal-dual saddle-point flow



$$
\begin{aligned}
\dot{y} &= -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - K_1 L \lambda \\
&\quad - K_1 \rho L y \\
\dot{\lambda} &= K_2 L y
\end{aligned}
$$

1. primal gradient descent
2. dual integral control penalizing violation of consensus constraint

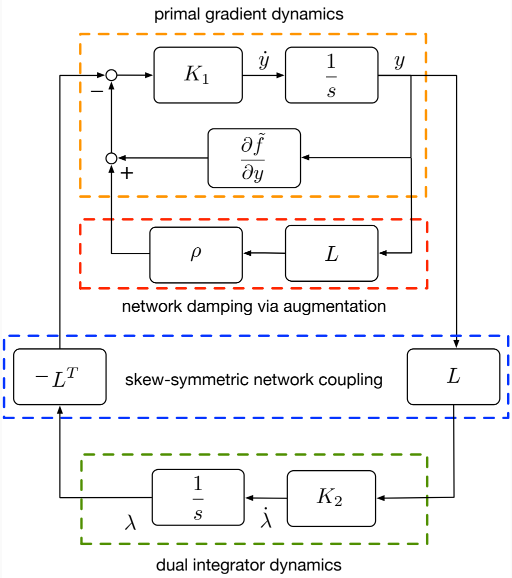
Block-diagram of distributed primal-dual saddle-point flow



$$
\begin{aligned}
\dot{y} &= -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - K_1 L \lambda \\
&\quad - K_1 \rho L y \\
\dot{\lambda} &= K_2 L y
\end{aligned}
$$

1. primal gradient descent
2. dual integral control penalizing violation of consensus constraint
3. skew-symmetric network coupling

Block-diagram of distributed primal-dual saddle-point flow

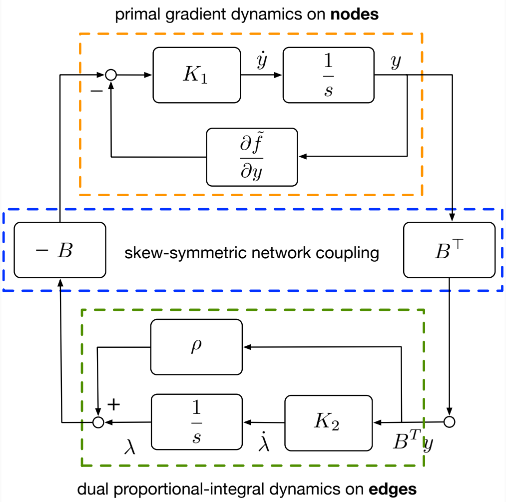


$$
\begin{aligned}
\dot{y} &= -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - K_1 L \lambda \\
&\quad - K_1 \rho L y \\
\dot{\lambda} &= K_2 L y
\end{aligned}
$$

1. primal gradient descent
2. dual integral control penalizing violation of consensus constraint
3. skew-symmetric network coupling
4. Laplacian damping from augmentation

Distributed optimization with incidence constraint

for **augmented Lagrangian** $\mathcal{L}_{aug}(y,\lambda) = \tilde{f}(y) + \lambda^\top B^\top y + \frac{\rho}{2} \cdot y^\top B B^\top y$



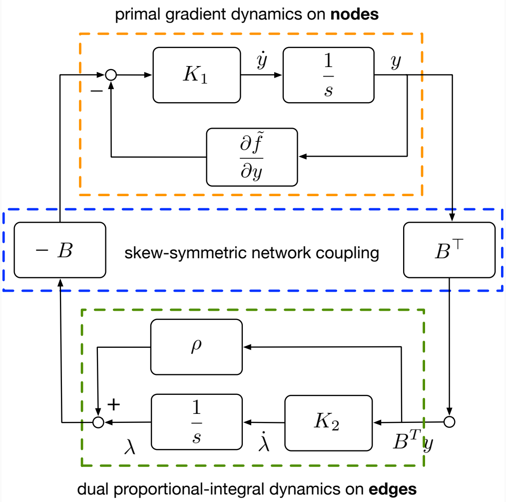
$$\dot{y} = -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - K_1 B\lambda - \rho K_1 B B^\top y$$

$$\dot{\lambda} = K_2 B^\top y$$

1. primal gradient descent

Distributed optimization with incidence constraint

for **augmented Lagrangian** $\mathcal{L}_{aug}(y,\lambda) = \tilde{f}(y) + \lambda^\top B^\top y + \frac{\rho}{2} \cdot y^\top B B^\top y$



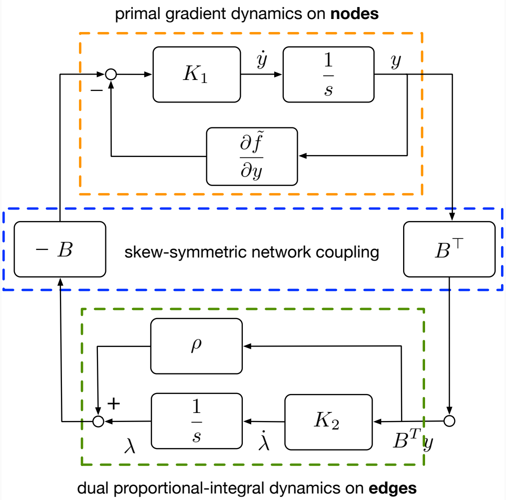
$$\dot{y} = -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - K_1 B\lambda - \rho K_1 B B^\top y$$

$$\dot{\lambda} = K_2 B^\top y$$

1. primal gradient descent
2. dual proportional-integral control penalizing consensus violation $B^T y$

Distributed optimization with incidence constraint

for **augmented Lagrangian** $\mathcal{L}_{aug}(y, \lambda) = \tilde{f}(y) + \lambda^\top B^\top y + \frac{\rho}{2} \cdot y^\top B B^\top y$



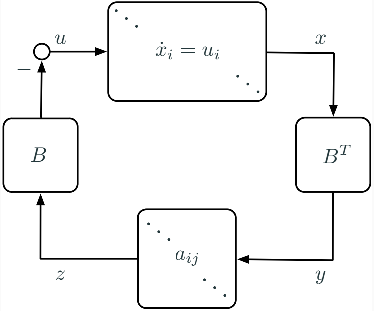
$$\dot{y} = -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - K_1 B\lambda - \rho K_1 B B^\top y$$

$$\dot{\lambda} = K_2 B^\top y$$

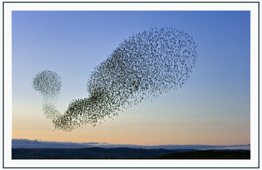
1. primal gradient descent
2. dual proportional-integral control penalizing consensus violation $B^T y$
3. skew-symmetric network coupling & implementation on nodes & edges

emerging patterns

Recall from Prior Lecture: Laplacian flow in relative sensing networks

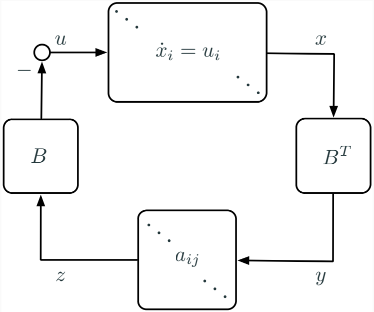


$$\dot{x} = -Lx$$
$$= -B\mathrm{diag}(\{a_e\}_{e\in E})B^\top x$$

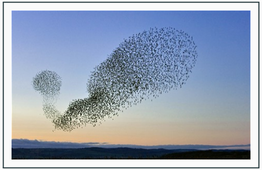


open-loop plant: $\dot{x}_i = u_i\,,\qquad i \in \{1,\dots,n\}\,,\quad$ or $\quad \dot{x} = u\,,$

Recall from Prior Lecture: Laplacian flow in relative sensing networks

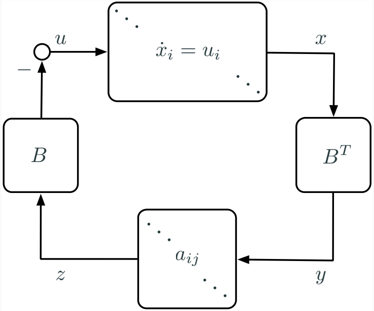


$$\dot{x} = -Lx$$
$$= -B\operatorname{diag}(\{a_e\}_{e\in E})B^\top x$$

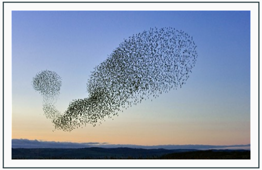


open-loop plant: $\dot{x}_i = u_i\,,\qquad i \in \{1,\ldots,n\}\,,\quad$ or $\quad \dot{x} = u\,,$

measurements: $y_{ij} = x_i - x_j\,,\qquad \{i,j\} \in E\,,\quad$ or $\quad y = B^\top x\,,$

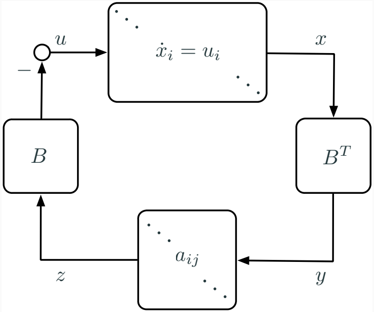


$$
\begin{aligned}
\dot{x} &= -Lx \\
&= -B\mathrm{diag}(\{a_e\}_{e\in E})B^\top x
\end{aligned}
$$

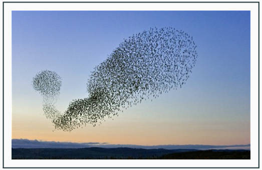


open-loop plant:	$\dot{x}_i = u_i\,,$	$i \in \{1,\dots,n\}\,,$	or	$\dot{x} = u\,,$
measurements:	$y_{ij} = x_i - x_j\,,$	$\{i,j\} \in E\,,$	or	$y = B^\top x\,,$
control gains:	$z_{ij} = a_{ij}y_{ij}\,,$	$\{i,j\} \in E\,,$	or	$z = \mathrm{diag}(a_e)y\,,$

Recall from Prior Lecture: Laplacian flow in relative sensing networks

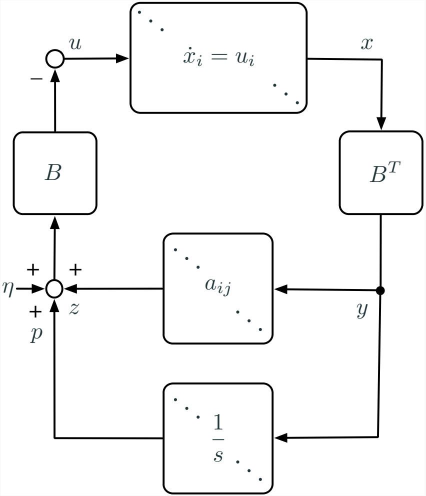


$$\dot{x} = -Lx$$
$$= -B\mathrm{diag}(\{a_e\}_{e\in E})B^\top x$$

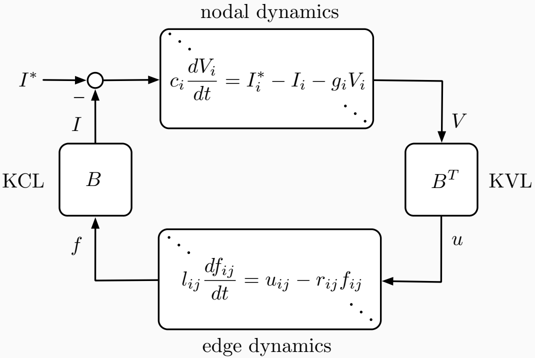


open-loop plant:	$\dot{x}_i = u_i\,,$	$i \in \{1,\dots,n\}\,,$	or	$\dot{x} = u\,,$
measurements:	$y_{ij} = x_i - x_j\,,$	$\{i,j\} \in E\,,$	or	$y = B^\top x\,,$
control gains:	$z_{ij} = a_{ij} y_{ij}\,,$	$\{i,j\} \in E\,,$	or	$z = \mathrm{diag}(a_e) y\,,$
control inputs:	$u_i = -\sum_{\{i,j\}\in E} z_{ij}\,,$	$i \in \{1,\dots,n\}\,,$	or	$u = Bz\,.$

Recall Homework 3 from 2021: distributed integral control



Recall Lecture 7: a prototypical circuit



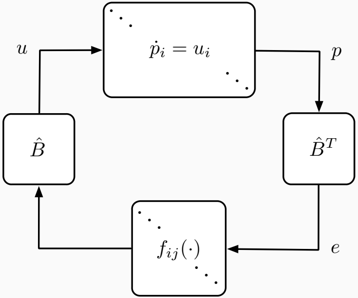
KCL: $I = Bf$

KVL: $u = B^\top V$

ground equation: $I = I^* - C\dot{V} - GV$

branch equation: $L\dot{f} = u - Rf$

Recall previous lecture: nonlinear relative sensing network



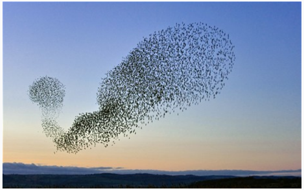
linear consensus: $f_{ij} = -a_{ij}e_{ij}$

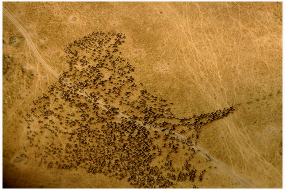
nonlinear consensus: $f_{ij} = -a_{ij}e_{ij}^3$

input constraints: $f_{ij} = -a_{ij}\mathrm{atan}(e_{ij})$

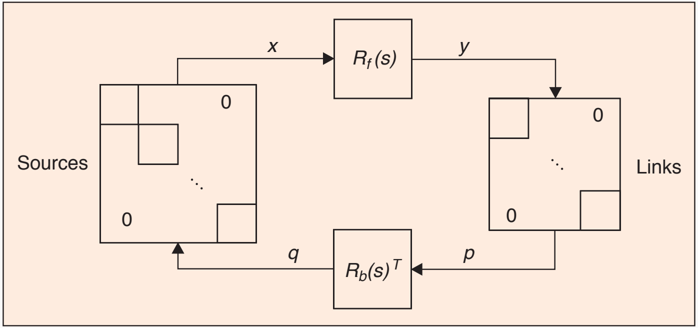
formation control: $f_{ij} = \boxed{}$

collision avoidance: $f_{ij} = \boxed{}$





...or internet congestion control

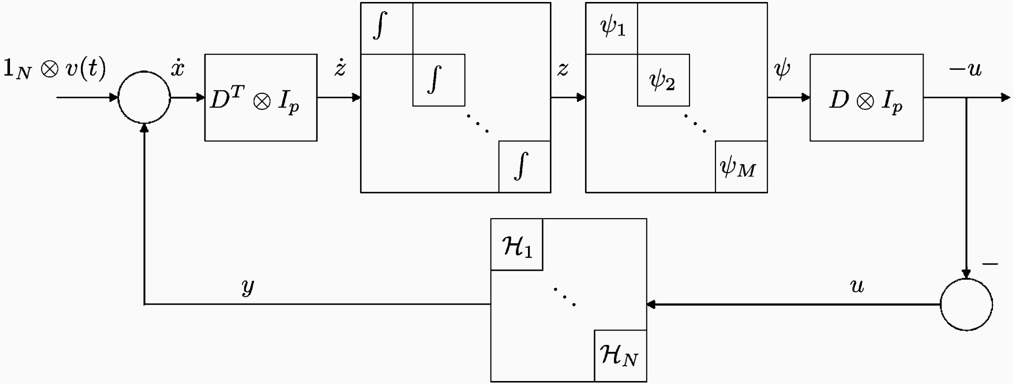


Internet Congestion Control

By Steven H. Low, Fernando Paganini, and John C. Doyle

This is essentially how **internet routing** (TCP/IP) works: distributed optimization coordinating dynamic protocols on sources and links via routing (incidence) matrix R

…or distributed coordination of passive systems



1380 IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 52, NO. 8, AUGUST 2007

Passivity as a Design Tool for Group Coordination

Murat Arcak, *Senior Member, IEEE*

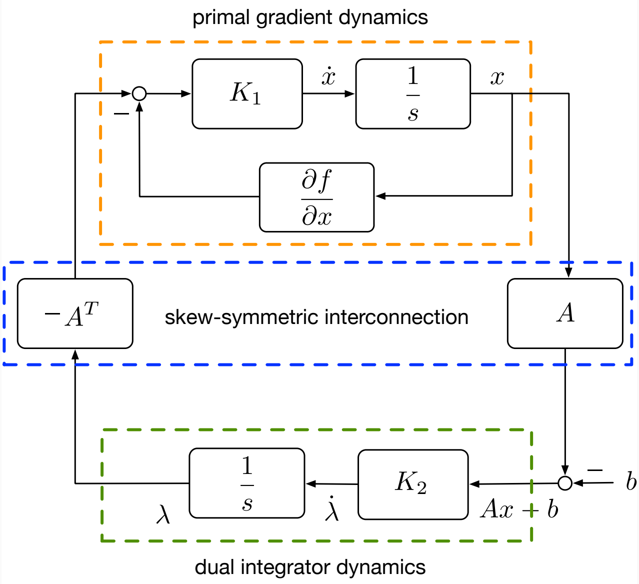
You can build an entire **system theory** around this block diagram …

…and many others that
follow the same scheme …

now let's look at other distributed optimization algorithms

dual ascent

Block-diagram of primal-dual saddle-point flow



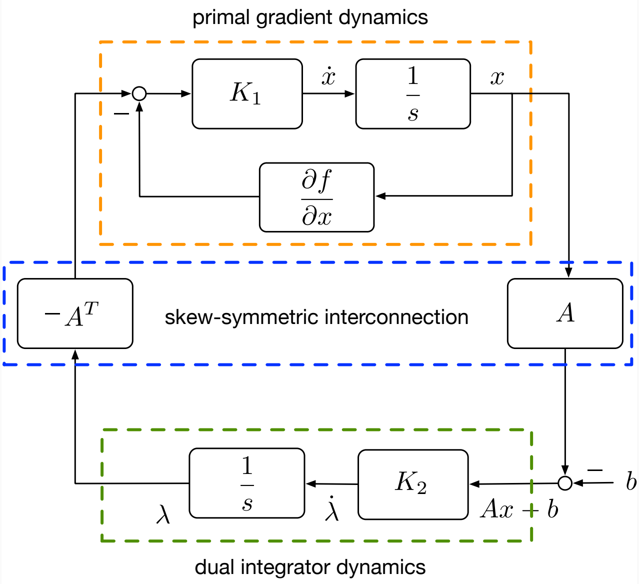
$$\dot{x} = -K_1 \frac{\partial f(x)}{\partial x} - K_1 A^\top \lambda$$

$$\dot{\lambda} = K_2 (Ax - b)$$

- **idea:** make primal dynamics faster than dual dynamics

$$\Rightarrow \|K_1\| \gg \|K_2\|$$

Block-diagram of primal-dual saddle-point flow



$$\dot{x} = -K_1 \frac{\partial f(x)}{\partial x} - K_1 A^\top \lambda$$

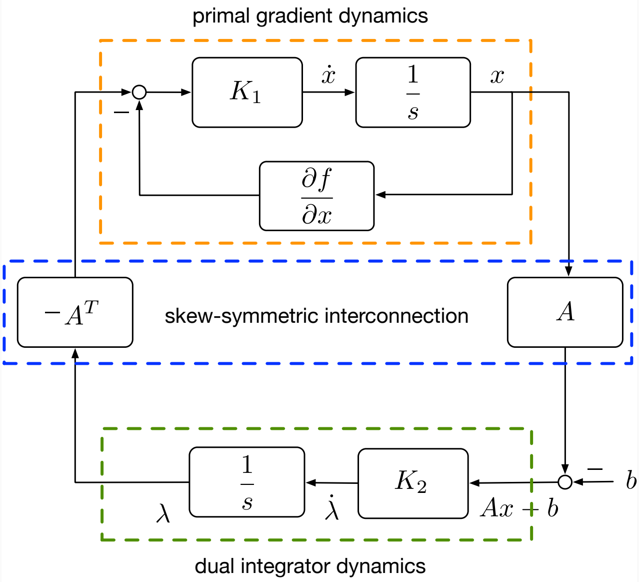
$$\dot{\lambda} = K_2 (Ax - b)$$

- **idea:** make primal dynamics faster than dual dynamics

$\Rightarrow \; \|K_1\| \gg \|K_2\|$

- **limit case**: $\|K_1\| \to \infty$

Block-diagram of primal-dual saddle-point flow



$$\dot{x} = -K_1 \frac{\partial f(x)}{\partial x} - K_1 A^\top \lambda$$

$$\dot{\lambda} = K_2 (Ax - b)$$

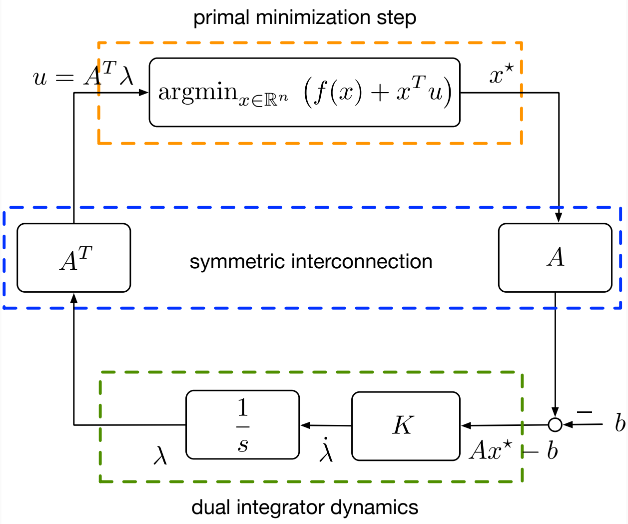
- **idea:** make primal dynamics faster than dual dynamics

$\Rightarrow \quad \|K_1\| \gg \|K_2\|$

- **limit case** : $\|K_1\| \to \infty$

$\Rightarrow$ primal loop is algebraic

Block-diagram of dual ascent

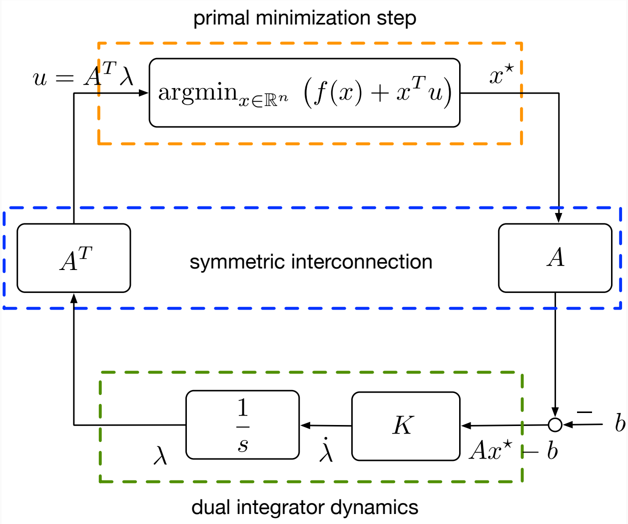


$$x^\star = \text{argmin}_{x\in\mathbb{R}^n} \left(f(x) + x^\top A^\top \lambda\right)$$

$$\dot{\lambda} = K\left(Ax^\star(\lambda) - b\right)$$

1. primal minimization

Block-diagram of dual ascent

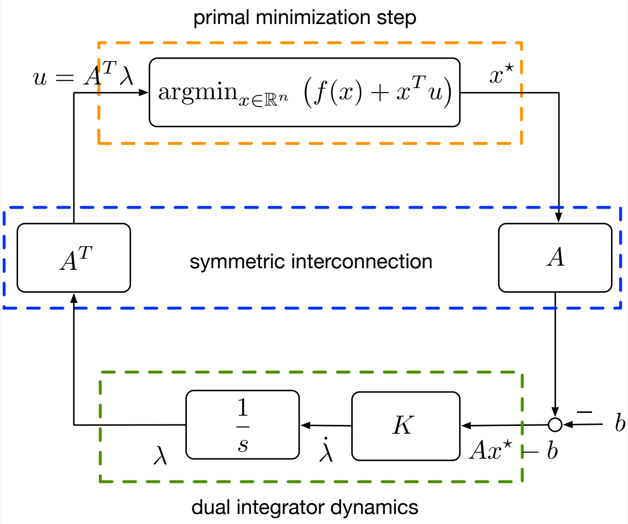


$$x^\star = \mathrm{argmin}_{x\in\mathbb{R}^n}\ (f(x) + x^\top A^\top \lambda)$$

$$\dot\lambda = K(Ax^\star(\lambda) - b)$$

1. primal minimization
2. dual integral control penalizing constraint violation

Block-diagram of dual ascent



$$x^\star = \mathrm{argmin}_{x\in\mathbb{R}^n}\left(f(x) + x^\top A^\top\lambda\right)$$

$$\dot{\lambda} = K\left(Ax^\star(\lambda) - b\right)$$

1. primal minimization
2. dual integral control penalizing constraint violation
3. symmetric interconnection

Example with LQ Problem

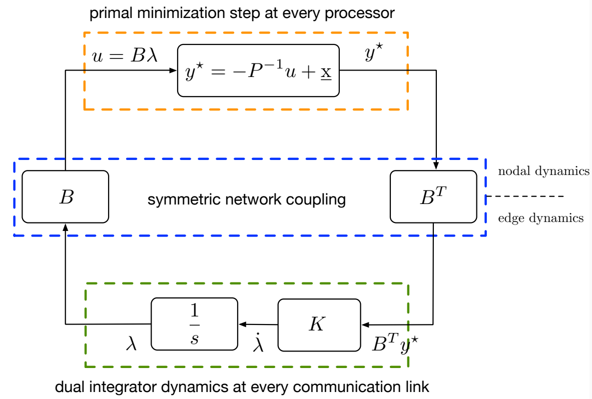
Example with LQ Problem

distributed dual ascent with incidence constraint

Distributed dual ascent with incidence constraint

Distributed dual ascent with incidence constraint

Block-diagram of distributed dual ascent

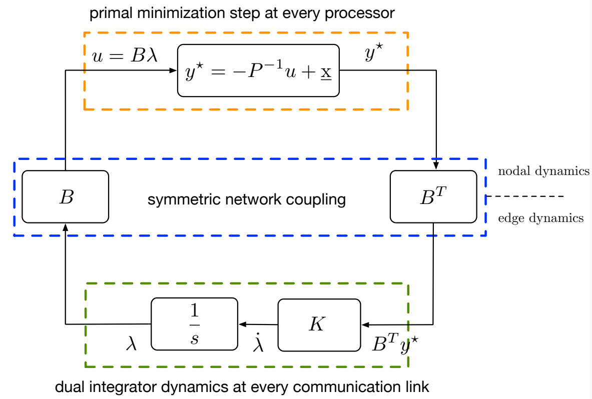


1. primal minimization

$$y_i^\star = \operatorname{argmin}_{y_i} \left(\frac{1}{2}(y_i - \underline{x}_i)^\top P_i (y_i - \underline{x}_i) + y_i^\top (B\lambda)_i \right) \quad , \quad \dot{\lambda} = K B^\top y^\star(\lambda)$$

primal update on **nodes** (processors) — dual updates on **edges** (routers)

Block-diagram of distributed dual ascent



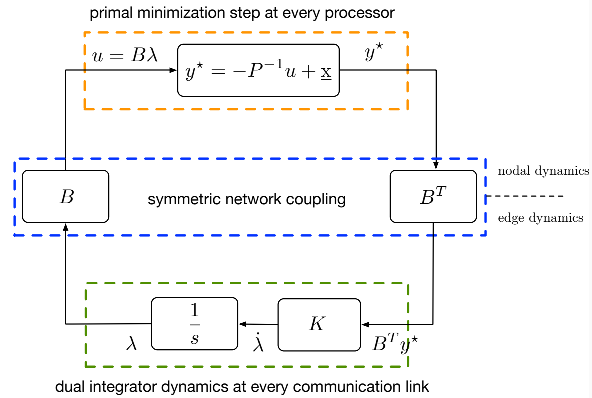
1. primal minimization
2. dual integral control penalizing violation of incidence constraint

$$y_i^\star = \operatorname{argmin}_{y_i} \left(\frac{1}{2}(y_i - \underline{x}_i)^\top P_i (y_i - \underline{x}_i) + y_i^\top (B\lambda)_i \right) \quad , \quad \dot{\lambda} = K B^\top y^\star(\lambda)$$

primal update on **nodes** (processors)

dual updates on **edges** (routers)

Block-diagram of distributed dual ascent

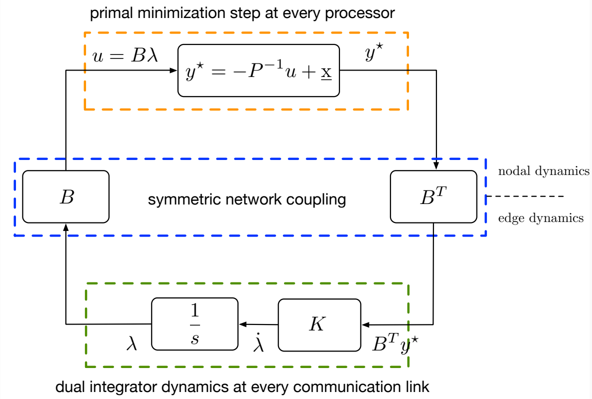


1. primal minimization
2. dual integral control penalizing violation of incidence constraint
3. symmetric network coupling via B & $B^\top$

$$y_i^\star = \mathrm{argmin}_{y_i} \left(\frac{1}{2}(y_i - \underline{x}_i)^\top P_i (y_i - \underline{x}_i) + y_i^\top (B\lambda)_i \right) \quad , \quad \dot{\lambda} = K B^\top y^\star(\lambda)$$

primal update on **nodes** (processors) dual updates on **edges** (routers)

Block-diagram of distributed dual ascent



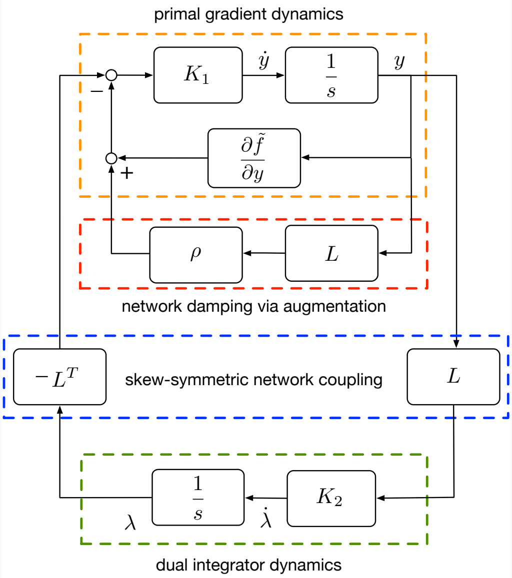
1. primal minimization
2. dual integral control penalizing violation of incidence constraint
3. symmetric network coupling via B & $B^\top$
4. **incidence constraint** node & edge updates

$$y_i^\star = \operatorname{argmin}_{y_i} \left(\frac{1}{2}(y_i - \underline{x}_i)^\top P_i (y_i - \underline{x}_i) + y_i^\top (B\lambda)_i \right) \quad , \quad \dot{\lambda} = K B^\top y^\star(\lambda)$$

primal update on **nodes** (processors) dual updates on **edges** (routers)

distributed gradient descent

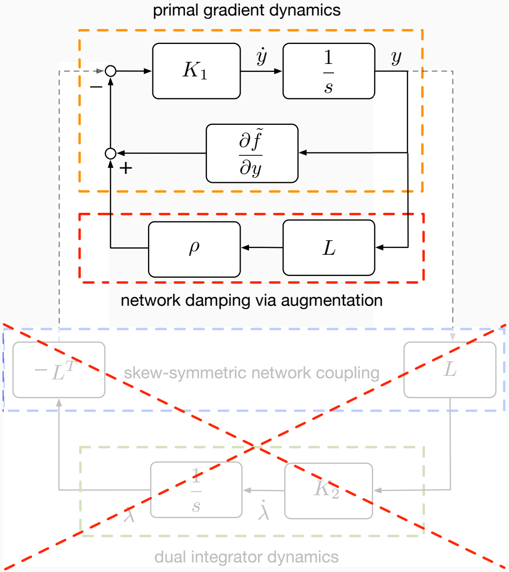
Block-diagram of distributed primal-dual saddle-point flow



$$\begin{aligned}\dot{y} &= -K_1 \frac{\partial \tilde{f}(y)}{\partial y} \\ &\quad - \rho K_1 L y - K_1 L \lambda \\ \dot{\lambda} &= +K_2 L y\end{aligned}$$

1. **idea I:** take dual update gain to zero: $K_2 = 0$

Block-diagram of distributed primal-dual saddle-point flow

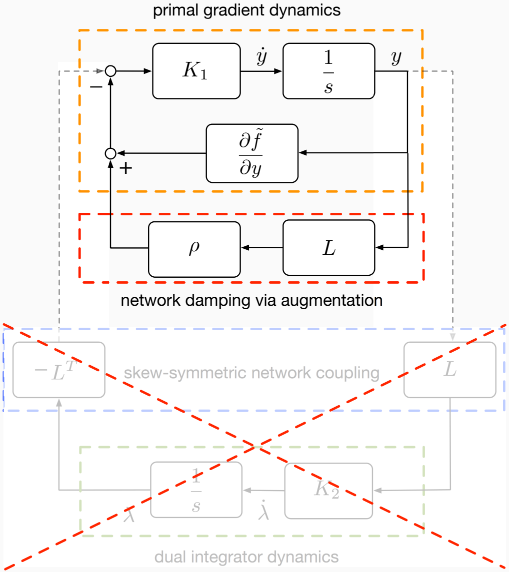


$$\dot{y} = -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - \rho K_1 L y \cancel{- K_1 L \lambda}$$

$$\cancel{\dot{\lambda} = +K_2 L y}$$

1. **idea I:** take dual update gain to zero: $K_2 = 0$

Block-diagram of distributed primal-dual saddle-point flow



$$\dot{y} = -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - \rho K_1 L y \cancel{- K_1 L \lambda}$$

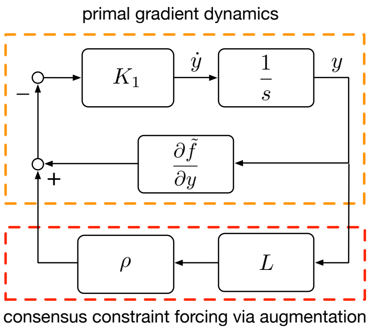
$$\cancel{\dot{\lambda} = + K_2 L y}$$

1. **idea I:** take dual update gain to zero: $K_2 = 0$

$\Rightarrow$ thus, we are left with a gradient descent for the **modified objective**

$$\underset{y}{\text{minimize}}\ \tilde{f}(y) + \frac{\rho}{2} \cdot y^\top L y$$

Block-diagram of distributed gradient descent

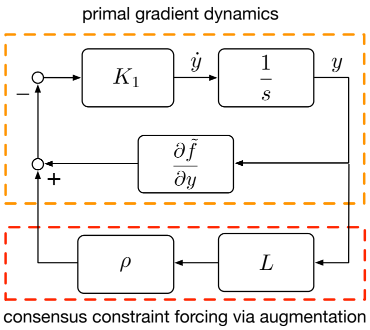


$$\dot{y} = -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - \rho K_1 L y$$

$\Rightarrow$ gradient descent for **modified objective**

$$\underset{y}{\text{minimize}}\ \tilde{f}(y) + \frac{\rho}{2} \cdot y^\top L y$$

Block-diagram of distributed gradient descent



$$\dot{y} = -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - \rho \, K_1 L y$$

$\Rightarrow$ gradient descent for **modified objective**

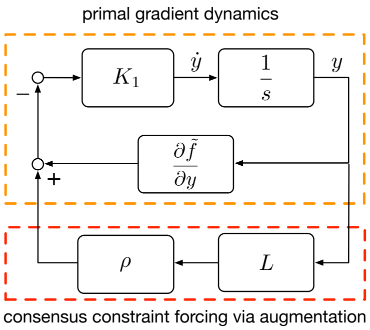
$$\underset{y}{\text{minimize}} \; \tilde{f}(y) + \frac{\rho}{2} \cdot y^\top L y$$

2. **idea II:** make augmentation gain ρ "much larger" than K_1 to enforce consensus

$$L y^\star \approx \mathbf{0}$$

at the optimal solution

Block-diagram of distributed gradient descent



$$\dot{y} = -K_1 \frac{\partial \tilde{f}(y)}{\partial y} - \rho \, K_1 L y$$

$\Rightarrow$ gradient descent for **modified objective**

$$\underset{y}{\text{minimize}} \; \tilde{f}(y) + \frac{\rho}{2} \cdot y^\top L y$$

1. even better: make the gains time-varying so that consensus dominates asymptotically …

2. **idea II:** make augmentation gain ρ "much larger" than K_1 to enforce consensus

$$L y^\star \approx \mathbf{0}$$

at the optimal solution

For a more comprehensive course on optimization, check out the course **“Large Scale Distributed Optimization”**

Thanks for your attention!

We hope to see you in person next fall!