Solving LMI Problems using MATLAB

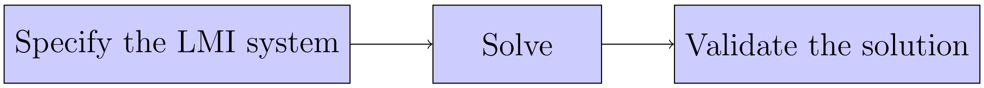

Figure 1: Steps to solve an LMI problem using MATLAB

Specify the LMI System:
Initialize the LMI system
```
setlmis([]);
```

Declare variables
```
lmivar(type, struct);
X = lmivar(type, struct);
[X, n, sX] = lmivar(type, struct);
```

`type=1` Symmetric matrices with a block-diagonal structure.

`struct` = R-by-2 matrix (if `X` has R blocks)

`struct(r,1)`: size of r^{th} block in `X`

`struct(r,2)`: type of r^{th} block (1 for full, 0 for scalar and -1 for zero block)

`type=2` Full m-by-n rectangular matrices; `struct` = [m n] in this case

`type=3` Other structures

Examples: Variable Declaration

- `X` is a 5-by-5 symmetric matrix
  `[X,n,sX] = lmivar(1,[5 1])`
- $$Z = \begin{bmatrix} \Phi_{3\times 3} & & & \\ & 0_{5\times 5} & & \\ & & \lambda & \\ & & & \delta I_{2\times 2} \end{bmatrix}$$

  `Z = lmivar(1,[3 1;5 -1;1 0;2 0])`
- `Y` is a 2-by-4 rectangular matrix
  `[Y,n,sY] = lmivar(2,[2 4])`
- $$M = \begin{bmatrix} 0 & -X \\ Y & 0 \end{bmatrix}$$

  `Z = lmivar(3, [zeros(5,4) -sX;sY zeros(2,5)])`

Define LMIs
```
lmiterm(termID, A, B, flag)
```
LMI terms are either outer factors, constant matrices or variable terms.

`termID` A 4-entry vector specifying the LMI number, term position and variable involved

`A, B` Left and right multipliers of a variable term

`flag` Optional; useful to specify conjugate expressions

Example: LMI Definition

Consider an LMI

$$\begin{bmatrix} A^T X + XA & B^T X \\ XB & -I \end{bmatrix} < N^T \begin{bmatrix} C^T C & 0 \\ 0 & D^T D \end{bmatrix} N$$

```
lmiterm([1 1 1 X], A',1,'s');
lmiterm([1 1 2 X], B',1);
lmiterm([1 2 2 0], -1);

lmiterm([-1 0 0 0], N);
lmiterm([-1 1 1 0], C'*C);
lmiterm([-1 2 2 0], D'*D);
```

Close the LMI system description
`LMIsys = getLMIs`

Invoke the solver:
`[tmin,xfeas] = feasp(LMIsys, ...)` for feasibility
`[copt,xopt] = mincx(LMIsys,c, ...)` for linear objective minimization

Validate the results:
```
evals = evallmi(LMIsys,decvars)
[lhs, rhs]=showlmi(LMIsys,evals)
```

Matlab Examples:

1.

$$\begin{bmatrix} -P & 0 \\ 0 & A^T P + PA \end{bmatrix} < 0$$

2.

$$\begin{aligned} \min \quad & \text{trace(X)} \\ \text{s.t.} \quad & \begin{bmatrix} A^T X + XA + Q & XB \\ B^T X & -I \end{bmatrix} < 0 \\ & X < 0 \end{aligned}$$

Introduction to YALMIP
YALMIP (Yet Another LMI Parser) is a modelling language for advanced modeling and solution of convex and nonconvex optimization problems. It is a free toolbox for MATLAB; can be downloaded from: http://users.isy.liu.se/johanl/yalmip/

- High-level description of the optimization problem
- Can use external solvers

Key Commands: `sdpvar, solvesdp`

$\mathcal{H}_\infty$ **Norm Computation**: Given $G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$, $\|G\|_\infty$ computation is equivalent to the LMI problem:

$$
\begin{aligned}
\min \quad & \gamma \\
\text{s.t.} \quad & \begin{bmatrix} A^T P + PA & PB & C^T \\ B^T P & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0 \\
& P > 0
\end{aligned}
$$