

# Scalable Robustness Analysis Using Integral Quadratic Constraints

Anders Rantzer

LCCC Linnaeus Center, Lund University, Sweden



## A Grand Challenge



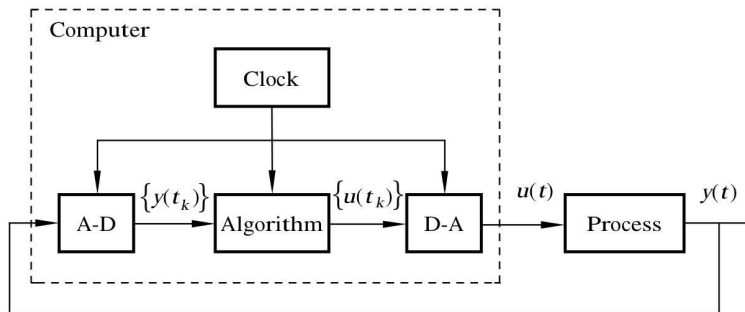
A control system should be delivered with

- 1 A specification of closed loop **requirements**
- 2 A network of interconnected **process models** (including controller hardware)
- 3 A **controller code**
- 4 A **certificate** proving that code and processes together meet the requirements. Validation of certificates must **scale linearly** with the number of interconnected components.

Is this possible?



## A Standard Setup

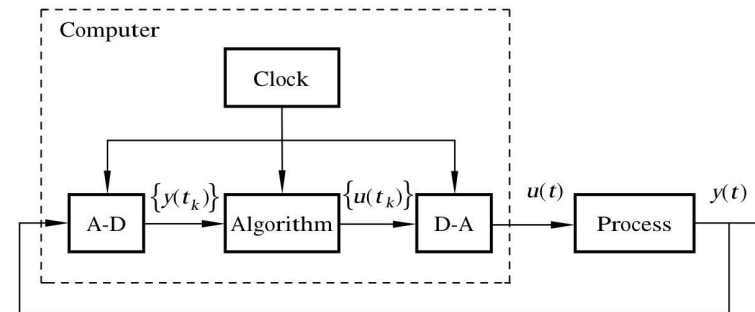


For quadratic requirements, linear process model and linear control algorithm, verification is straightforward...

... but is it scalable?



## A Standard Setup

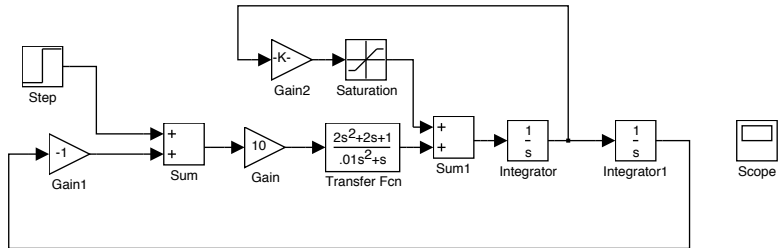


For quadratic requirements, linear process model and linear control algorithm, verification is straightforward...

... but is it scalable?



## A servo with friction



Simulations show stability.

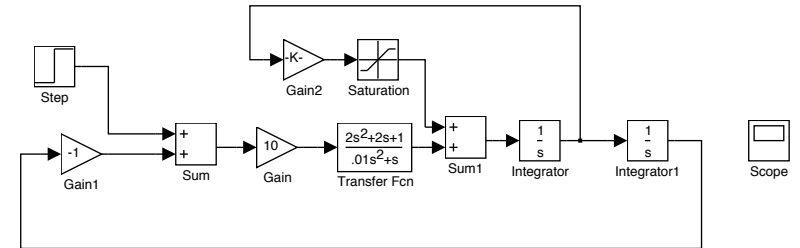
The circle criterion can *prove* stability.

But what if the feedback controller induces time delays?

:



## A servo with friction



Simulations show stability.

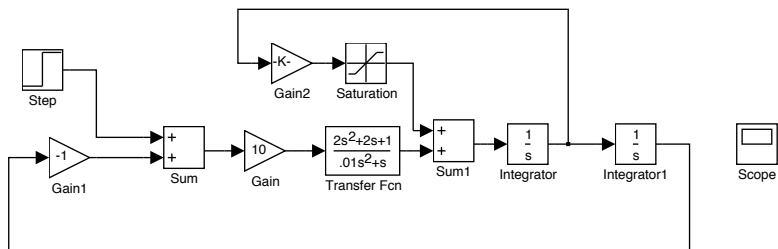
The circle criterion can *prove* stability.

But what if the feedback controller induces time delays?

:



## A servo with friction



Simulations show stability.

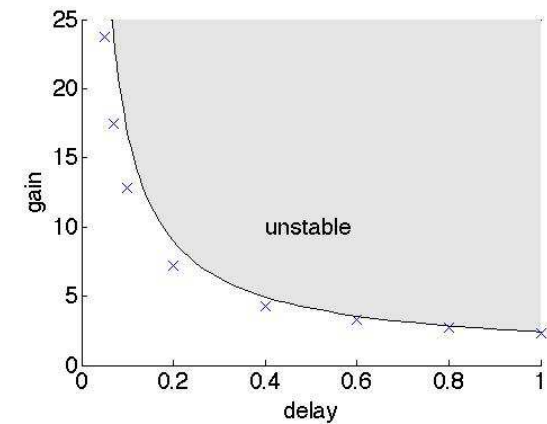
The circle criterion can *prove* stability.

But what if the feedback controller induces time delays?

:

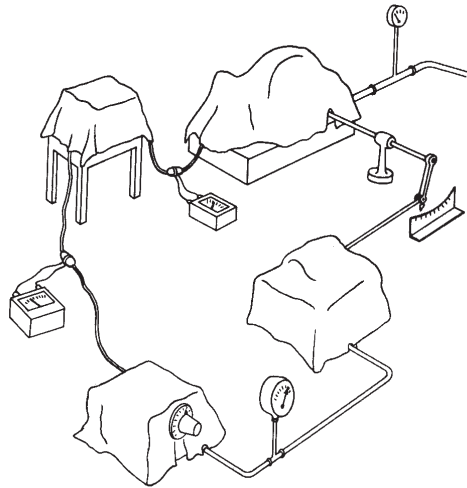


## Stability by simulation



Every cross represents a stable simulation.

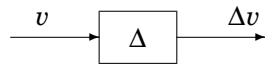
But what about in between?



- Integral Quadratic Constraints
- A Matlab tool for verification
- Matrix Decomposition
- Making IQC Analysis Scalable
- Conclusions



## Integral Quadratic Constraint



The (possibly nonlinear) operator  $\Delta$  on  $\mathbf{L}_2^m[0, \infty)$  is said to satisfy the IQC defined by  $\Pi$  if

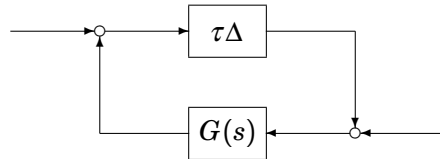
$$\int_{-\infty}^{\infty} \begin{bmatrix} \widehat{v}(i\omega) \\ (\widehat{\Delta v})(i\omega) \end{bmatrix}^* \Pi(i\omega) \begin{bmatrix} \widehat{v}(i\omega) \\ (\widehat{\Delta v})(i\omega) \end{bmatrix} d\omega \geq 0$$

for all  $v \in \mathbf{L}_2[0, \infty)$ .

$\Delta$ structure	$\Pi(i\omega)$	Condition
$\Delta$ passive	$\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$	
$\ \Delta(i\omega)\  \leq 1$	$\begin{bmatrix} x(i\omega)I & 0 \\ 0 & -x(i\omega)I \end{bmatrix}$	$x(i\omega) \geq 0$
$\delta \in [-1, 1]$	$\begin{bmatrix} X(i\omega) & Y(i\omega) \\ Y(i\omega)^* & -X(i\omega) \end{bmatrix}$	$X = X^* \geq 0$ $Y = -Y^*$
$\delta(t) \in [-1, 1]$	$\begin{bmatrix} X & Y \\ Y^T & -X \end{bmatrix}$	
$\Delta(s) = e^{-\theta s} - 1$	$\begin{bmatrix} x(i\omega)\rho(\omega)^2 & 0 \\ 0 & -x(i\omega) \end{bmatrix}$	$\rho(\omega) = 2 \max_{ \theta  \leq \theta_0} \sin(\theta\omega/2)$



# IQC Stability Theorem



Let  $G(s)$  be stable and proper and let  $\Delta$  be causal.

For all  $\tau \in [0, 1]$ , suppose the loop is well posed and  $\tau\Delta$  satisfies the IQC defined by  $\Pi(i\omega)$ . If

$$\begin{bmatrix} G(i\omega) \\ I \end{bmatrix}^* \Pi(i\omega) \begin{bmatrix} G(i\omega) \\ I \end{bmatrix} < 0 \quad \text{for } \omega \in [0, \infty]$$

then the feedback system is input/output stable.

:



# Underlying Math Problem



Given a number of symmetric matrices, find a convex combination that is positive definite!

$$\begin{pmatrix} * & * & * \\ * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * \end{pmatrix} = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}$$

:



# S-procedure



The inequality

$$\sigma_0(h) \leq 0$$

follows from the inequalities

$$\sigma_1(h) \geq 0, \dots, \sigma_n(h) \geq 0$$

if there exist  $\tau_1, \dots, \tau_n \geq 0$  such that

$$\sigma_0(h) + \sum_k \tau_k \sigma_k(h) \leq 0 \quad \forall h$$

:



# Outline

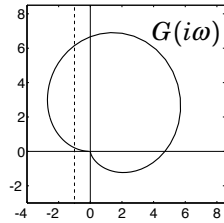
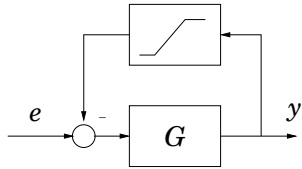


- Integral Quadratic Constraints
- **A Matlab tool for verification**
- Matrix Decomposition
- Making IQC Analysis Scalable
- Conclusions

:



## A Matlab toolbox for system analysis



```
>> abst_init_iqc;
>> G = tf([10 0 0],[1 2 2 1]);
>> e = signal
>> w = signal
>> y = -G*(e+w)
>> w==iqc_monotonic(y)
>> iqc_gain_tbx(e,y)
```

:

```
>> iqc_gui('fricSYSTEM')
```

extracting information from fricSYSTEM ...

```
scalar inputs: 5
states:       10
simple q-forms: 7
```

```
LMI #1  size = 1  states: 0
LMI #2  size = 1  states: 0
LMI #3  size = 1  states: 0
LMI #4  size = 1  states: 0
LMI #5  size = 1  states: 0
```

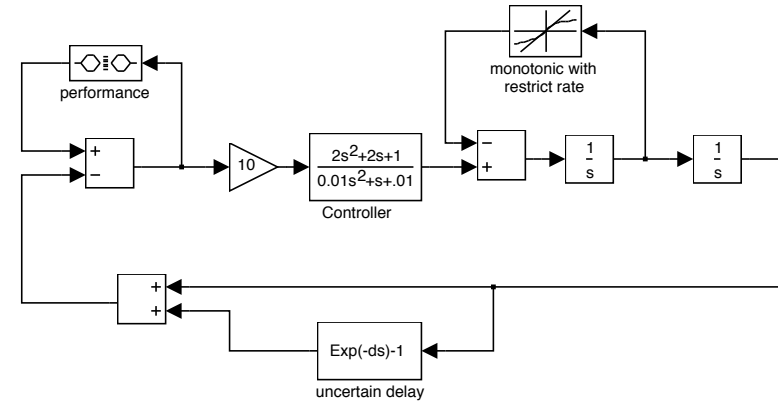
Solving with 62 decision variables ...

```
ans = 4.7139
```

:



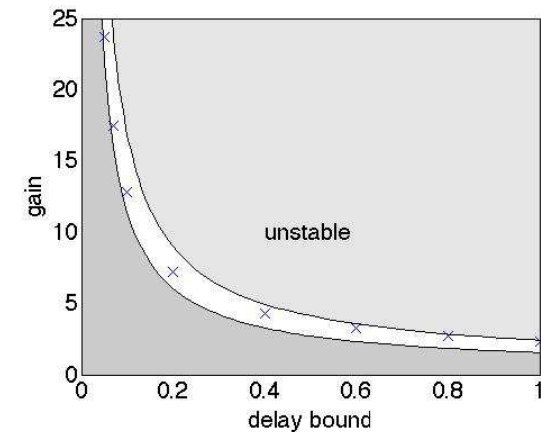
## An analysis model defined graphically



:



## Verification by IQCs



IQCs prove stability below the lower line.

:







# Outline



- Integral Quadratic Constraints
- A Matlab tool for verification
- Matrix Decomposition
- **Making IQC Analysis Scalable**
- Conclusions



# S-procedure for IQC Analysis



Find  $\tau_1, \dots, \tau_n \geq 0$  such that  $\sigma_0(h) + \sum_k \tau_k \sigma_k(h)$  becomes negative semi-definite:

$$\begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \tau_1 \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \tau_2 \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}$$



# Decomposing IQC Analysis



Find  $\tau_1, \dots, \tau_n \geq 0$  such that  $\sigma_0(h) + \sum_k \tau_k \sigma_k(h)$  has a negative semi-definite decomposition:

$$\begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \tau_1 \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \tau_2 \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}$$

Distributed certificates!

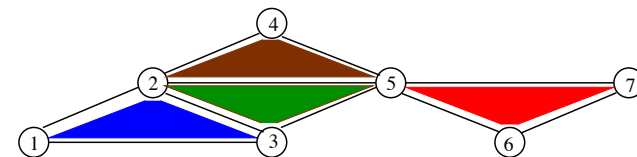


# Chordal Decompositions



Cholesky factors inherit the sparsity structure of the symmetric matrix if and only if the sparsity pattern corresponds to a "chordal" graph.

$$\begin{pmatrix} * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \end{pmatrix} = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}$$

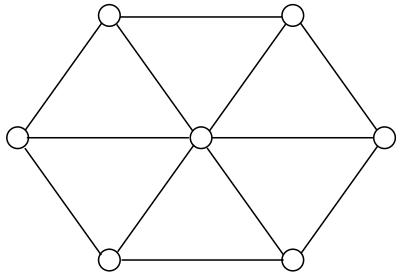


[Blair & Peyton, An introduction to chordal graphs and clique trees, 1992]





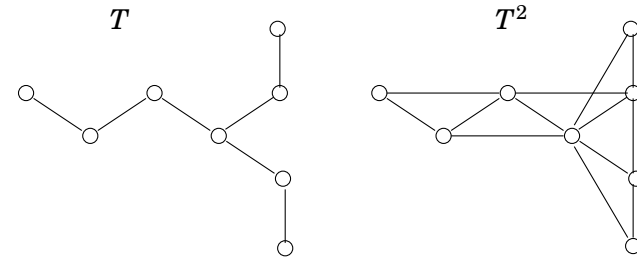
## Example: Non-chordal graph



## Example: Chordal graphs



If  $T$  is a tree, then  $T^k$  is chordal for every  $k \geq 1$ .



## Network congestion control



Maximize  $\sum_i U_i(x)$  over  $x_i \geq 0$  subject to  $\sum_i R_{li}x_i \leq c_l$

Alternatively:  $\min_{p_l \geq 0} \max_{x_i \geq 0} \sum_i [U_i(x_i) - \sum_l p_l (R_{li}x_i - c_l)]$

A model for Internet dynamics can look like this:

$$\dot{x}_i(t) = k_i x_i(t) \left( 1 - \frac{\sum_l R_{li} p_l(t - \tau_{li})}{U'_i(x_i(t))} \right)$$

$$\beta_l \dot{p}_l(t) + p_l(t) = \sum_i x_i(t - \tau_{il})$$

Scalable stability conditions by Low, Paganini, Doyle, Papachristodolou, Vinnicombe, Lestas, Pates, ...

Is there a connection to scalable IQC analysis?



## Network congestion control

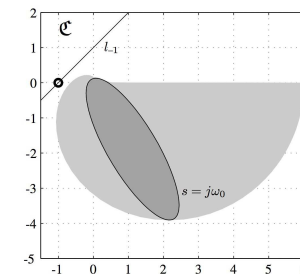


Yes!

[Pates/Vinnicombe 2012]:

Separate the ellipse

$\left\{ \frac{z^* L(i\omega) z}{z^* z} : z \in \mathbb{C}^n \right\}$  from  $-1$



$$\begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} + \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}$$

Sources for conservatism: fixed decomposition  
fixed separating hyperplane

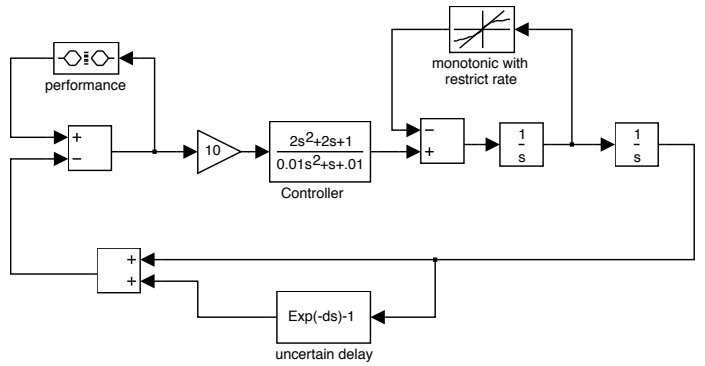


# Outline

- Integral Quadratic Constraints
- A Matlab tool for verification
- Matrix Decomposition
- Making IQC Analysis Scalable
- Conclusions



# Distributed Verification



[Feron (2010)]: “The credible autocoder produces not only a target code that implements control-system specifications but also documents the target code with its properties and their proofs.”

Line 8 is informative since it forces  $y_c$  to be bounded by one in absolute value. This information can be extracted in a more compact way by using the `max` function. In the next section, we will see how this information is used to generate the postcondition  $\{x \in \mathcal{E}_9, y_c^2 \leq 1\}$ . However, the postcondition  $\{x \in \mathcal{E}_9, y_c^2 \leq 1\}$  is complicated to propagate forward. Instead, consider the weaker annotation

$$\left\{ \begin{bmatrix} x \\ y_c \end{bmatrix} \in \mathcal{E}_9 \right\}, \mathcal{O}_9 = \begin{bmatrix} \mu P & 0_{5 \times 1} \\ 0_{1 \times 2} & 1 - \mu \end{bmatrix} \quad (13)$$

where  $\mu$  is an arbitrary parameter between zero and one. In the rest of this section,  $\mu$  is assigned the value 0.9991 found above during the controller stability analysis. The weakest annotation is stated as follows: `propagate(9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000`

Asking what annotation (13) becomes after passing through line 9 is equivalent to asking what the image of an ellipsoid through a linear mapping is. Since this image is also an ellipsoid in this case a one-dimensional segment, a valid assertion in the postcondition of line 9 is  $y_c^2 \leq [C, D, Q_c^{-1} C, D, J]^T$ , while keeping the remaining assertions identical. The bound on  $y_c$  is a straightforward consequence of the fact that  $\max_{y_c^2 \leq 1} y_c^2 = \sqrt{C^T C}$ , for the positive-definite matrix  $C$ . The resulting triple is therefore

$$\left\{ \begin{bmatrix} x \\ y_c \end{bmatrix} \in \mathcal{E}_9 \right\}, y_c = Cx + xC + Dc + y_c c$$

$$\left\{ \begin{bmatrix} x \\ y_c \end{bmatrix} \in \mathcal{E}_9, y_c^2 \leq [C, D, Q_c^{-1} C, D, J]^T \right\}$$

which propagates the set containing  $x$  and  $y_c$ , while providing a bound on the output variable  $y_c$ . Next, line 10 transforms the ellipsoid  $\mathcal{E}_9$  into the ellipsoid  $\mathcal{E}_8$ , where

$$P = [(A, B, Q_c^{-1} C, B, J)^T]^{-1} \quad (14)$$

Thus line 10 leads to the triple

$$\left\{ \begin{bmatrix} x \\ y_c \end{bmatrix} \in \mathcal{E}_8, y_c^2 \leq [C, D, Q_c^{-1} C, D, J]^T \right\}$$

$$10: \quad xc = Ac + xc + Bc + y_c c$$

$$K_8 \in \mathcal{E}_8, y_c^2 \leq [C, D, Q_c^{-1} C, D, J]^T$$

Line 11 merely writes the output of the controller to the appropriate memory location for further processing. Since it is not used afterwards, it can, but does not have to, be dropped from the triple. This is leading to the triple  $\{x \in \mathcal{E}_8, y_c^2 \leq [C, D, Q_c^{-1} C, D, J]^T\}$ . Lines 12 and 13 do not bring new information

**TABLE 3** Annotated Matlab program, forward constraint propagation. Although the annotation process requires global understanding of the program semantics, verifying the correctness of the annotations can be done using verification techniques, but automated verification is possible and much faster.

```

(100)
(101)
(102)
(103)
(104)
(105)
(106)
(107)
(108)
(109)
(110)
(111)
(112)
(113)
(114)
(115)
(116)
(117)
(118)
(119)
(120)
(121)
(122)
(123)
(124)
(125)
(126)
(127)
(128)
(129)
(130)
(131)
(132)
(133)
(134)
(135)
(136)
(137)
(138)
(139)
(140)
(141)
(142)
(143)
(144)
(145)
(146)
(147)
(148)
(149)
(150)
(151)
(152)
(153)
(154)
(155)
(156)
(157)
(158)
(159)
(160)
(161)
(162)
(163)
(164)
(165)
(166)
(167)
(168)
(169)
(170)
(171)
(172)
(173)
(174)
(175)
(176)
(177)
(178)
(179)
(180)
(181)
(182)
(183)
(184)
(185)
(186)
(187)
(188)
(189)
(190)
(191)
(192)
(193)
(194)
(195)
(196)
(197)
(198)
(199)
(200)
(201)
(202)
(203)
(204)
(205)
(206)
(207)
(208)
(209)
(210)
(211)
(212)
(213)
(214)
(215)
(216)
(217)
(218)
(219)
(220)
(221)
(222)
(223)
(224)
(225)
(226)
(227)
(228)
(229)
(230)
(231)
(232)
(233)
(234)
(235)
(236)
(237)
(238)
(239)
(240)
(241)
(242)
(243)
(244)
(245)
(246)
(247)
(248)
(249)
(250)
(251)
(252)
(253)
(254)
(255)
(256)
(257)
(258)
(259)
(260)
(261)
(262)
(263)
(264)
(265)
(266)
(267)
(268)
(269)
(270)
(271)
(272)
(273)
(274)
(275)
(276)
(277)
(278)
(279)
(280)
(281)
(282)
(283)
(284)
(285)
(286)
(287)
(288)
(289)
(290)
(291)
(292)
(293)
(294)
(295)
(296)
(297)
(298)
(299)
(300)
(301)
(302)
(303)
(304)
(305)
(306)
(307)
(308)
(309)
(310)
(311)
(312)
(313)
(314)
(315)
(316)
(317)
(318)
(319)
(320)
(321)
(322)
(323)
(324)
(325)
(326)
(327)
(328)
(329)
(330)
(331)
(332)
(333)
(334)
(335)
(336)
(337)
(338)
(339)
(340)
(341)
(342)
(343)
(344)
(345)
(346)
(347)
(348)
(349)
(350)
(351)
(352)
(353)
(354)
(355)
(356)
(357)
(358)
(359)
(360)
(361)
(362)
(363)
(364)
(365)
(366)
(367)
(368)
(369)
(370)
(371)
(372)
(373)
(374)
(375)
(376)
(377)
(378)
(379)
(380)
(381)
(382)
(383)
(384)
(385)
(386)
(387)
(388)
(389)
(390)
(391)
(392)
(393)
(394)
(395)
(396)
(397)
(398)
(399)
(400)
(401)
(402)
(403)
(404)
(405)
(406)
(407)
(408)
(409)
(410)
(411)
(412)
(413)
(414)
(415)
(416)
(417)
(418)
(419)
(420)
(421)
(422)
(423)
(424)
(425)
(426)
(427)
(428)
(429)
(430)
(431)
(432)
(433)
(434)
(435)
(436)
(437)
(438)
(439)
(440)
(441)
(442)
(443)
(444)
(445)
(446)
(447)
(448)
(449)
(450)
(451)
(452)
(453)
(454)
(455)
(456)
(457)
(458)
(459)
(460)
(461)
(462)
(463)
(464)
(465)
(466)
(467)
(468)
(469)
(470)
(471)
(472)
(473)
(474)
(475)
(476)
(477)
(478)
(479)
(480)
(481)
(482)
(483)
(484)
(485)
(486)
(487)
(488)
(489)
(490)
(491)
(492)
(493)
(494)
(495)
(496)
(497)
(498)
(499)
(500)
(501)
(502)
(503)
(504)
(505)
(506)
(507)
(508)
(509)
(510)
(511)
(512)
(513)
(514)
(515)
(516)
(517)
(518)
(519)
(520)
(521)
(522)
(523)
(524)
(525)
(526)
(527)
(528)
(529)
(530)
(531)
(532)
(533)
(534)
(535)
(536)
(537)
(538)
(539)
(540)
(541)
(542)
(543)
(544)
(545)
(546)
(547)
(548)
(549)
(550)
(551)
(552)
(553)
(554)
(555)
(556)
(557)
(558)
(559)
(560)
(561)
(562)
(563)
(564)
(565)
(566)
(567)
(568)
(569)
(570)
(571)
(572)
(573)
(574)
(575)
(576)
(577)
(578)
(579)
(580)
(581)
(582)
(583)
(584)
(585)
(586)
(587)
(588)
(589)
(590)
(591)
(592)
(593)
(594)
(595)
(596)
(597)
(598)
(599)
(600)
(601)
(602)
(603)
(604)
(605)
(606)
(607)
(608)
(609)
(610)
(611)
(612)
(613)
(614)
(615)
(616)
(617)
(618)
(619)
(620)
(621)
(622)
(623)
(624)
(625)
(626)
(627)
(628)
(629)
(630)
(631)
(632)
(633)
(634)
(635)
(636)
(637)
(638)
(639)
(640)
(641)
(642)
(643)
(644)
(645)
(646)
(647)
(648)
(649)
(650)
(651)
(652)
(653)
(654)
(655)
(656)
(657)
(658)
(659)
(660)
(661)
(662)
(663)
(664)
(665)
(666)
(667)
(668)
(669)
(670)
(671)
(672)
(673)
(674)
(675)
(676)
(677)
(678)
(679)
(680)
(681)
(682)
(683)
(684)
(685)
(686)
(687)
(688)
(689)
(690)
(691)
(692)
(693)
(694)
(695)
(696)
(697)
(698)
(699)
(700)
(701)
(702)
(703)
(704)
(705)
(706)
(707)
(708)
(709)
(710)
(711)
(712)
(713)
(714)
(715)
(716)
(717)
(718)
(719)
(720)
(721)
(722)
(723)
(724)
(725)
(726)
(727)
(728)
(729)
(730)
(731)
(732)
(733)
(734)
(735)
(736)
(737)
(738)
(739)
(740)
(741)
(742)
(743)
(744)
(745)
(746)
(747)
(748)
(749)
(750)
(751)
(752)
(753)
(754)
(755)
(756)
(757)
(758)
(759)
(760)
(761)
(762)
(763)
(764)
(765)
(766)
(767)
(768)
(769)
(770)
(771)
(772)
(773)
(774)
(775)
(776)
(777)
(778)
(779)
(780)
(781)
(782)
(783)
(784)
(785)
(786)
(787)
(788)
(789)
(790)
(791)
(792)
(793)
(794)
(795)
(796)
(797)
(798)
(799)
(800)
(801)
(802)
(803)
(804)
(805)
(806)
(807)
(808)
(809)
(810)
(811)
(812)
(813)
(814)
(815)
(816)
(817)
(818)
(819)
(820)
(821)
(822)
(823)
(824)
(825)
(826)
(827)
(828)
(829)
(830)
(831)
(832)
(833)
(834)
(835)
(836)
(837)
(838)
(839)
(840)
(841)
(842)
(843)
(844)
(845)
(846)
(847)
(848)
(849)
(850)
(851)
(852)
(853)
(854)
(855)
(856)
(857)
(858)
(859)
(860)
(861)
(862)
(863)
(864)
(865)
(866)
(867)
(868)
(869)
(870)
(871)
(872)
(873)
(874)
(875)
(876)
(877)
(878)
(879)
(880)
(881)
(882)
(883)
(884)
(885)
(886)
(887)
(888)
(889)
(890)
(891)
(892)
(893)
(894)
(895)
(896)
(897)
(898)
(899)
(900)
(901)
(902)
(903)
(904)
(905)
(906)
(907)
(908)
(909)
(910)
(911)
(912)
(913)
(914)
(915)
(916)
(917)
(918)
(919)
(920)
(921)
(922)
(923)
(924)
(925)
(926)
(927)
(928)
(929)
(930)
(931)
(932)
(933)
(934)
(935)
(936)
(937)
(938)
(939)
(940)
(941)
(942)
(943)
(944)
(945)
(946)
(947)
(948)
(949)
(950)
(951)
(952)
(953)
(954)
(955)
(956)
(957)
(958)
(959)
(960)
(961)
(962)
(963)
(964)
(965)
(966)
(967)
(968)
(969)
(970)
(971)
(972)
(973)
(974)
(975)
(976)
(977)
(978)
(979)
(980)
(981)
(982)
(983)
(984)
(985)
(986)
(987)
(988)
(989)
(990)
(991)
(992)
(993)
(994)
(995)
(996)
(997)
(998)
(999)
(1000)

```

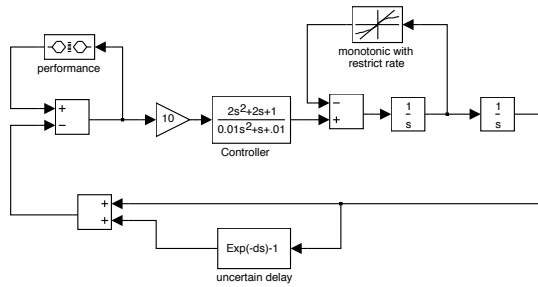
## From Control Systems to Control Software

INTEGRATING LYAPUNOV-THEORETIC PROOFS WITHIN CODE

ERIC FERON



# Summary



IQC analysis scales using positive definite decompositions !

$$\begin{pmatrix}
 * & * & * & * \\
 * & * & * & * \\
 * & * & * & * \\
 * & * & * & *
 \end{pmatrix}
 =
 \begin{pmatrix}
 * & * & * \\
 * & * & * \\
 * & * & *
 \end{pmatrix}
 +
 \begin{pmatrix}
 * & * & * \\
 * & * & * \\
 * & * & *
 \end{pmatrix}
 +
 \begin{pmatrix}
 * & * & * \\
 * & * & * \\
 * & * & *
 \end{pmatrix}
 +
 \begin{pmatrix}
 * & * & * \\
 * & * & * \\
 * & * & *
 \end{pmatrix}$$

Scalability comes from monotonicity.