2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

A Novel Method to Support the Formulation of Control-Oriented Reduced-Order Models of Thermal Power Generation Processes

Francesco Casella[1] and Matteo Luigi De Pascali[1]

Abstract— **The energy transition will require innovative sustainable thermal power generation systems to supplement intermittent renewable power sources. Accurate dynamic models of these plants can be built using process simulation tools, but they are often way too complex to be directly used for model-based control design. This paper presents a novel method that starts from results obtained from the accurate dynamic model and helps selecting the minimal number of state variables for writing first-principle control-oriented simplified models, assessing the trade-off between complexity and accuracy. Two test cases are presented: a super-critical CO_2 boiler-turbine system, and the innovative SOS-CO_2 system, which combines a closed recuperated CO_2 Brayton cycle with a fuel cell.**

I. INTRODUCTION

The forthcoming energy transition, spurred by the urgent need to curb anthropogenic CO_2 emissions, will have a dramatic impact on the electrical power generation sector, which is responsible for a third of those emissions [1]. Most of the new installed power generation in the next 30 years will be intermittent solar photovoltaic and wind power [2], leaving to thermal generation processes an increased burden in the balancing and stabilization of power grids.

After several decades in which the dominant technologies in controllable power generation (coal-fired, combined-cycle, hydro, and nuclear fission) remained basically the same, this will likely generate a very strong interest in the design and installation of innovative and sustainable thermal power generation systems, involving, e.g., green hydrogen, carbon capture and storage, fuel cells, innovative thermodynamic cycles, etc. These plants will need to operate in a highly dynamic regime, to counteract the variability of intermittent renewable sources, likely requiring advanced, model-based control systems for their operation.

As of today, powerful and highly flexible object-oriented, equation-based modelling and simulation methodologies and tools are available for this purpose, such as, for example, those based on the Modelica language [3], as well as libraries of reusable component models for thermal power generation processes, such as [4], [5], [6]. These tools allow to build detailed first-principles dynamic process models by an equation-based modular approach, typically resulting in models with tens of thousands of equations and hundreds of state variables, which are too complex to be used directly for model-based control design and are typically used to perform simulation studies.

[1]Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Milan, Italy (francesco.casella@polimi.it, matteoluigi.depascali@polimi.it)

As discussed in [7], such detailed *knowledge models* can be complemented by control-oriented, low-order, highly simplified first-principles *interpretation models*, which can explain the fundamental control-relevant dynamics (though not as accurately as *knowledge models*) and can be useful to develop control strategies and model-based controllers. For traditional steam thermal power plants, such models emerged between the 70s and the 90s of the last century through a trial-and-error process, in which *interpretation models* were proposed by making educated guesses about suitable simplifications and by validating them against the results of *knowledge models*, see again [7] for some examples. Unfortunately, innovative systems being proposed today are often much more complex and tightly integrated, making this process difficult and in dire need of help from systematic methods.

There are many techniques available to obtain "black-box" reduced models, e.g. balanced reduction for linearized models, or the identification of nonlinear models from training data generated by detailed simulation models. Unfortunately, all these models suffer from a fundamental limitation, i.e., their internal variables and equations have no physical interpretation, which makes them unsuitable as *interpretation models* and can also be dangerous if they are used outside their training range.

The ideal situation is when one can derive a simplified first-principle models by introducing drastic simplifying assumptions, but still employing equations that have a physical justification (balance equations, fluid equations of state, turbomachinery maps, heat transfer correlations, etc.).

The goal of this paper is to present a novel method to help deriving such *interpretation models*, applicable to innovative power generation process where there is little or no prior experience to guide the reduced-order modelling process. Given a reference frequency value, the method indicates which state variables and which time-varying variables must be included in the simplified model in order to achieve a sufficiently accurate estimation of the frequency response at that frequency. Based on this information, it is much easier for an expert modeller to derive the simplified model, even though this process still requires human skill and expertise.

The paper is structured as follows: the state and variable selection algorithm is illustrated in Section II and demonstrated on two example cases: a supercritical CO_2 boiler-turbine model in Section III and the novel SOS-CO_2 plant, which combines a closed recuperated CO_2 cycle with a fuel cell, in Section IV. Section V concludes the paper with final remarks and future work discussion.

979-8-3503-1632-2/24/$31.00 ©2024 IEEE

II. THE STATE AND VARIABLE SELECTION METHOD

The starting point of the process is a detailed, equation-based dynamic model of the process, which could be built using a Modelica tool, using suitable component libraries. The model is originally formulated as a system of differential-algebraic equations; it is then automatically transformed by the Modelica tool into state-space form (see [8] for further details):

$$\dot{x}(t) = f(x(t), u(t)) \tag{1}$$

$$y(t) = g(x(t), u(t)) \tag{2}$$

where x is the vector of state variables, u is the vector of input variables, and y is the vector of output variables. Executable code is generated to compute functions $f(x,u)$ and $g(x,u)$, internally using numerical solvers to handle algebraic loops that cannot be resolved symbolically.

The Modelica tool can also solve the steady-state initialization problem:

$$0 = f(\bar{x}, \bar{u}) \tag{3}$$

$$\bar{y} = g(\bar{x}, \bar{u}) \tag{4}$$

by either specifying the steady state inputs $\bar{u}$ or outputs $\bar{y}$ and using an iterative Newton-Raphson solver. The tool can then differentiate equations (1)-(2), either numerically by finite differences or by means of automatic differentiation techniques, thus producing the A,B,C,D matrices of the LTI linearized model valid around that equilibrium:

$$\Delta\dot{x}(t) = A\Delta x(t) + B\Delta u(t) \tag{5}$$

$$\Delta y(t) = C\Delta x(t) + D\Delta u(t), \tag{6}$$

where $\Delta x(t) = x(t) - \bar{x}$, $\Delta u(t) = u(t) - \bar{u}$, $\Delta y(t) = y(t) - \bar{y}$.

When tackling a novel thermal generation process for the first time, one can start exploring the dynamic behaviour of the plant model around a certain steady-state, e.g. the design conditions or some off-design operating points, by initializing the system on that steady state and then simulating the response of (1)-(2) to small step changes to the inputs.

In many cases, one observes responses that could be approximated, over a certain time horizon, by a low-order system. By reasoning in the frequency domain, one could approximate that dynamic response with a reduced-order system, whose linearization has approximately the same frequency response $G(j\omega^*)$ at the frequency ω^* of interest for control, e.g., the cross-over frequency ω_c if one want to design PID controllers using loop-shaping techniques.

In order to derive a reduced-order model, one needs to figure out what are the relevant physical phenomena (e.g. mass or energy balances) and what are the corresponding variables (e.g., pressures or temperatures) that are relevant to approximate $G(j\omega^*)$. In general, three situations can arise:

1) a state variable Δx_j and its corresponding dynamic state equation, typically corresponding to some balance equation, are relevant to determine $G(j\omega^*)$ with sufficient accuracy;
2) a state variable Δx_j is relevant to determine $G(j\omega^*)$ with sufficient accuracy, but the dynamic behaviour introduced by its state equation is negligible, so its state equation can be approximated by a static one, following a singular perturbation approach [9];
3) a state variable Δx_j does not contribute significantly to $G(j\omega^*)$, which means that the corresponding state variable x_j of the original model can be considered as a constant when deriving the reduced model.

Given a nonlinear process model (1)-(2), a reference equilibrium point $(\bar{x}, \bar{u}, \bar{y})$, a pair of scalar input and output variables u_r, y_s, and a reference frequency ω^*, the proposed method aims at finding a minimal set of variables of type 1. and 2. that will need to be part of a first-principles simplified model that reproduces the dynamic behaviour at frequency ω^* with sufficient accuracy. In most cases, this physical model will also be able to reproduce the dynamic behaviour around other equilibrium points, though this cannot be guaranteed a priori and needs to be checked a posteriori. The method goes through four main steps.

A. Step 1: Ranking relevant states

During the first step, a simple heuristic algorithm progressively removes state variables from the linearized model (5)-(6), eventually producing a sequence of reduced-order linearized models M_k with increasingly larger state-space size, each characterized by a percentage error

$$e_{k,\%} = 100 \cdot \frac{|G_k(j\omega^*) - G(j\omega^*)|}{|G(j\omega^*)|}. \tag{7}$$

The result of this step can be summarized as in Table I, which can be interpreted as follows: a model that only includes x_3 as a state variable will have an 81.3% relative error in $G_1(j\omega^*)$; a model that includes x_3 and x_1 will have a 42.5% error in $G_2(j\omega^*)$; a model that includes x_3, x_1, and x_4 will have an 8.9% error in $G_3(j\omega^*)$, etc. The error on the last row is always zero, as it corresponds to the degenerate case of a reduced model identical to the original one.

The algorithm is described by the pseudo-code in Alg. 1. First, the frequency response at frequency ω^* is computed for the original linearized system (5)-(6), considering the single input-output pair u_r, y_s. An empty list *RM* is instantiated to collect the indices of the states in matrix A that will be removed during the iterations, and an empty list of percentage errors is initialized containing a zero element.

While the number of removed states is lower than the total number of states, the main iteration cycle is repeated. First, the rows and columns of the system matrix collected

TABLE I: Example result of Step 1 of the method

State	Error [%]
x_3	81.3
x_1	42.5
x_4	8.9
x_5	2.5
x_6	0.4
x_2	0.0

Algorithm 1

```
n = size(A,1)
B = B[:, r]
C = C[s, :]
H = freqResp(A, B, C, D, ω*)
RM = [ ] # List of removed states
ERR = [0] # List of the corresponding errors
while len(RM) < n do
    # i are the indices of the states in matrix A
    A(RM.i, RM.i) = [ ]
    B(RM.i, :) = [ ]
    C(:, RM.i) = [ ]
    E = +Inf
    for j in the remaining states do
        A* = A([1 : j−1, j+1 : end], [1 : j−1, j+1 : end])
        B* = B([1 : j−1, j+1 : end], :)
        C* = C(:, [1 : j−1, j+1 : end])
        H* = freqResp(A*, B*, C*, D, ω*)
        e = abs((H − H*)/H)
        if e < E then
            E = e
            k = j
        end if
    end for
    RM = [RM, k]
    ERR = [ERR, E]
end while
RM = RM[end : −1 : 1]
ERR = ERR[end : −1 : 1]
```

in the list RM are eliminated. Then, a for loop is executed on the remaining states, removing them one at a time; the removed state that leads to the smallest relative error in the computation of $G(j\omega^*)$ is selected and stored in the list RM, alongside with the corresponding error e.

At the end of the algorithm, the two lists RM and ERR are reversed and displayed as in Table I. Note that the obtained ranking is not necessarily the optimal one: it may as well happen that some combination of reduced-order states different from the ones that are obtained by taking the first k rows of Table I leads to a smaller error for the same number of state variables. On the other hand, looking for these optimal configurations would lead to a combinatorial explosion of combinations to be tested, which is not feasible. The proposed greedy algorithm is a simple heuristics for tackling this combinatorial problem, which however proved to work well in several tested examples.

Also note that there is no guarantee that the values of the error in the second column of Table I are monotonically decreasing. For example, in some cases it may well happen that a certain physical phenomenon is described by the combined effect of two states x_j and x_k; removing both could increase the error by, say, 10%, while removing only one may create an imbalanced model with a much larger error.

A final important remark is due at this point: when modelling thermal power generation systems, 1D distributed-parameter models are often employed, e.g., for the modelling of heat exchangers. In this case, the state variable of the original partial differential equation (PDE), e.g., a temperature distribution $T(x,t)$, is discretized into an *array* of n scalar variables, e.g., $T_j(t),\ j = 1,...,n$, using methods such as finite volumes, finite elements, or finite differences. Accurate simulation models usually require moderately high values of n, e.g., 10 or 20. It is apparent how the influence on $G(j\omega^*)$ of a single element T_j of such an array will be quite small in case of large n; in this case, the proposed heuristics would progressively remove all those individual variables one by one, because their small contribution leads to a small error when removed from the system, until none is left, even though the original variable of the PDE has a significant impact in determining $G(j\omega^*)$.

Therefore, each array of state variables, which typically comes from the discretization of a 1D model, will be handled atomically, i.e., it will be removed as a whole from the system matrix instead of one element at a time. This also substantially reduces the number of iterations required for large systems, which may have in excess of 1000 individual state variables, but not more than a few dozens if the 1D-discretized variables are considered as atomic entities.

Regarding the chemical part of the models, it often involves arrays of compositions that may become state variables of the models because of mass balance equations; in the scope of this work, these arrays were also considered as atomic entities for the simplification process, i.e., the chemical part of the model is either fully retained or fully neglected.

B. Step 2: Deciding the reduced-order model size

Once Step 1 is complete, a result of the analysis such as Table I will be available. At this point the expert modeller will have to take a decision on where to draw the line between the relevant and irrelevant states. This involves a trade off between the complexity of the model and its accuracy. Controllers with a moderate level of robustness can usually tolerate errors up to $20-30\%$ without too much trouble, but this ultimately depends on how the reduced model will be used for the model-based control design.

The outcome of this second step will be a list of relevant state variables corresponding to the first k elements of Table I and the corresponding relative error of the frequency response $G_k(j\omega^*)$.

C. Step 3: Further order reduction by singular perturbation

Once a decision has been made in Step 2 about the trade-off between the size of the reduced model and its accuracy around frequency ω^*, it is still possible to further reduce the order of the system without sacrificing the accuracy too much, by means of singular perturbations [9]. The idea is that one can partition the state vector x in two sub-vectors x_r and z, then neglect the dynamic term in the state equations

TABLE II:
Example result of Step 3 of the method

State	Error [%]
x_3	42.8
x_1	15.3
x_4	8.9

for z, i.e., write:

$$\Delta\dot{x}_r(t) = A_{11}\Delta x_r(t) + A_{12}\Delta z(t) + B_1\Delta u(t) \quad (8)$$

$$0 = A_{21}\Delta x_r(t) + A_{22}\Delta z(t) + B_2\Delta u(t) \quad (9)$$

$$\Delta y(t) = C_1\Delta x(t) + C_2\Delta z(t) + D\Delta u(t), \quad (10)$$

which can then be solved as

$$\Delta\dot{x}_r(t) = A^*\Delta x_r(t) + B^*\Delta u(t) \quad (11)$$

$$\Delta y(t) = C^*\Delta x(t) + D^*\Delta u(t), \quad (12)$$

where

$$A^* = A_{11} - A_{12}A_{22}^{-1}A_{21} \quad (13)$$

$$B^* = B_1 - A_{12}A_{22}^{-1}B_2 \quad (14)$$

$$C^* = C_1 - C_2A_{22}^{-1}A_{21} \quad (15)$$

$$D^* = D - C_2A_{22}^{-1}B_2 \quad (16)$$

Algorithm 2, whose inputs are the reduced-order A,B,C,D matrices obtained from the previous step and the frequency ω^*, is similar to Algorithm 1 and can be described by the following pseudo-code:

Algorithm 2

$H = \text{freqResp}(A,B,C,D,\omega^*)$
$RM = [\ ]$ # List of removed states
$ERR = [\text{E}_{\text{step1}}]$ # List of the corresponding errors
while $len(RM) < n$ **do**
  E = +Inf
  for j in the remaining states **do**
    Re-order the state x so the j-th state comes last
    Compute A^*,B^*,C^*,D^* by singular perturbations
    $H^* = \text{freqResp}(A^*,B^*,C^*,D,\omega^*)$
    $e = abs((H-H^*)/H)$
    if $e < E$ **then**
      $E = e$
      $k = j$
      $A_r = A^*$, $B_r = B^*$, $C_r = C^*$, $D_r = D^*$
    end if
  end for
  $A = A_r, B = B_r, C = C_r, D = D_r$
  $RM = [RM,k]$
  $ERR = [ERR,E]$
end while
$RM = RM[end:-1:1]$
$ERR = ERR[end:-1:1]$

Assuming the decision in Step 2 was to retain states x_3,x_1,x_4, the end result of the algorithm could be as shown in Table II, which can interpreted as follows: a model that keeps x_3 only as state and eliminates x_1,x_4 via singular perturbation leads to a 42.8% error on $G_1(j\omega^*)$; a model that keeps x_3,x_1 as states and eliminates x_4 via singular perturbation leads to a 15.3% error on $G_2(j\omega^*)$; a model which keeps x_3,x_1,x_4 as states leads to an 8.9% relative error on $G_3(j\omega^*)$.

D. Step 4: Deciding further reduction of the model via singular perturbation

Based on the results shown in Table II, the expert modeller will then once again weigh the trade-off between complexity and accuracy and decide which of the states selected in Step 2 should be retained as such in the reduced-order models, and which one should instead be considered as quasi-static.

E. Step 4: Writing the first-principles reduced model

Based on the results obtained in Steps 2 and 4, the modeller will then write a reduced-order model that can assume all the states eliminated in Step 2 as constants, all the states eliminated in Step 4 as time-varying variables, governed by algebraic equations, and all the surviving state variables as states, governed by differential state equations.

F. Implementation details

The Python language and the Scipy package [10] were used in this work; the open-source code is available on a GitHub repository [11].

As the proposed method is potentially applicable to large system models, possibly counting over a thousand state variables, extra care must be exercised to ensure an efficient and numerically robust implementation.

First of all, if the original physical process model is written in Modelica, its variables are always written using SI variables, which can make the corresponding model very badly scaled; the power output of a large power generation system could have order of magnitude 10^9 W, while control valve flow coefficients could be around 10^{-4} m^2, thus spanning 13 orders of magnitude. In order to avoid bad matrix conditioning due to scaling issues, it is recommended to scale all input, output, and state variables, so that they all have order of magnitude around unity.

Scypy's *freqResp()* function, if supplied the A,B,C,D matrices of the linearized system, converts the matrix representation into its zero-pole-gain form before evaluating the frequency response. When the size of the system grows, this method scales badly, performing inefficient calculations which may also fail. The following function should be used instead, which scales well also for system of order greater than one thousand.

```
def freqResp(A,B,C,D,w):
    jw = 1j*w
    I = numpy.identity(len(A))
    F = jw*I-A
    G = scipy.linalg.solve(F, B)
    fr = numpy.matmul(C,G) + D
    return fr
```

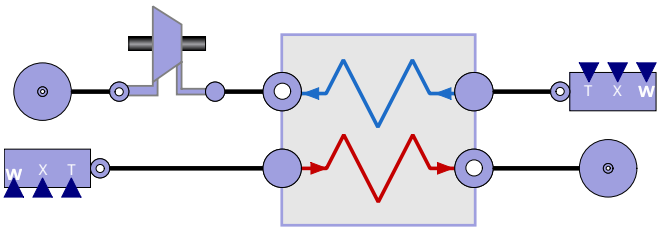

Fig. 1: The Modelica diagram of the sCO2 boiler + turbine.

TABLE III:
w_{hot} VS P_{el} @ 0.003 rad/s

State	Error [%]
wall.T[6]	100.00
coldside.p	45.01
coldside.T[6]	37.24
hotside.T[6]	9.38
hotside.p	0.00
coldside.X[6,5]	0.00
hotside.X[6,5]	0.00

TABLE IV:
w_{hot} VS P_{el}: type 2 states

State	Error [%]
wall.T[6]	11.36
coldside.T[6]	9.05
hotside.T[6]	6.72
coldside.p	9.38

III. EXAMPLE 1: SCO_2 BOILER AND TURBINE

In this section, a simplified model inspired by the supercritical CO_2 plant presented in [12] is analyzed. It comprises a heat exchanger representing the plant boiler and a turbine as shown in Fig. 1. The hot side fluid is an exhaust gas entering the boiler at 1400 K; the cold side fluid is pure CO_2 getting heated from 720 K to 890 K. The state variables of the detailed process model are the temperatures T, the pressures p and the mass fractions array X of each discretization volume of the heat exchanger and the temperatures T of the discretized metal walls separating the channels. The inputs of the system are the mass flow rates entering the two channels of the boiler with a fixed temperature and composition, while the output is the electrical power generated by the turbine. The Modelica model employed to represent the system features six finite volumes for each channel and for the wall, for a total number of differential-algebraic equations around 2100 and 80 scalar state variables.

The application of a mass flow rate step change on the hot side inlet changes the power output mainly through thermal phenomena, since more thermal power is available to power the turbine, but it first needs to be trasmitted to the cold side through the wall thermal inertia. The perturbed system reaches a new steady state condition with a time constant $\tau_1 \approx 300\,\mathrm{s}$, as shown by the red curve in Fig. 2.

The application of a mass flow rate step to the cold side causes instead a faster response of the turbine power with a time constant $\tau_2 \approx 30\,\mathrm{s}$, likely linked to the coupling between the channel pneumatic capacitance and the turbine admittance, followed by a slower settling transient (green curve of Fig. 3).

Once the time scale of interests of the two responses have been identified, the system can be analyzed with the proposed algorithm, focusing on the frequencies related to the aforementioned time constants.

A. Hot flow rate perturbation

Table III shows the output of Step 1 of the method for the input-output pair hot mass flow rate–power output. The analysis is made at a frequency $\omega^* = 1/\tau_1 \approx 0.003\,\mathrm{rad/s}$. Starting from the bottom of the table, a first pretty obvious result is that the states related to the compositions do not play any role, since the inlet composition is not changed and there are no chemical reactions along the heat exchanger channels. The hot side pressure can also be taken as a constant in the reduced model, without leading to a large error.

As a result of Step 2, a reasonable compromise is to keep as states the three temperatures and the cold side pressure, since this leads to an error of about 10%.

The result of Step 3 is shown in Table IV. Starting from the bottom, if all four states are kept as such, the error is the same as in Step 2, about 10%. If the singular perturbation transformation is applied to the cold side pressure state and the first three variables are kept as states, the error is actually slightly reduced. This is perfectly sensible, as the effect of the dynamic mass accumulation term in the cold side mass balance, which mainly depends on the cold side pressure derivative, is actually negligible if compared to the huge thermal inertia that dominates this transient. The same thing holds for the derivatives of the hot and cold side temperatures, which correspond to the dynamic energy storage terms of the two fluids, which are negligible compared to the energy storage in the wall, and thus can be removed from the model as states. Only keeping the wall temperature as a state variable, and the other two temperature and the the cold side pressure as variables not associated to any storage, leads to an error of 11%, which is acceptable.

We now consider four simplified models, all considering constant compostions. The first one, not reported here for space constraints, includes differential equations with the states corresponding to the first four rows of Table III: $T_{w,j}$, p_c, $T_{c,j}$, and $T_{h,j}$, $j = 1,\ldots,N$, with $N = 6$. These equations are the dynamic mass and energy balances for the six volumes of the cold side, coupled to the turbine equations, the dynamic energy balance for the six volumes of the heat exchanger walls, and the dynamic energy balance for the six volumes of the hot side, assuming constant hot side pressure. The overall model has 33 differential-algebraic equations and 19 states. The step response of this first model is shown by the orange dashed curve of Fig. 2, which is quite a good approximation.

Considering the six hot side temperatures as constant would lead the response shown by the yellow curve, which is a quite bad approximation, in accordance to the estimated 37% error reported in the third line of Table III for that case. This confirms the soundness of the decision taken in Step 2.

The results of Step 3 summarized in Table IV suggest that if we only consider $T_{w,j}$ as states and $T_{c,j}$, $T_{h,j}$, and p_c as time varying variables, we could still achieve a good accuracy, with an error of about 11%. Since the only surviving states are the wall temperatures, the only dynamic equations we are allowed to write are the energy balance equations on the

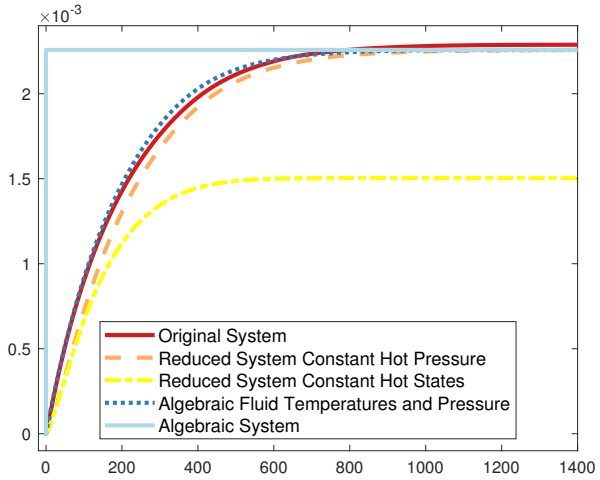

Fig. 2: Power output responses of the simplified system after a hot mass flow rate step through subsequent iterations.

wall volumes, while the mass and energy balance equations on the hot and cold fluid volumes, which depend on the corresponding time-varying pressures and temperatures, will be static ones, without any accumulation term.

To this aim, we can define the additional following time-varying variables: the hot mass flow rate input w_h, the thermal power flows $Q_{c,j}$ and $Q_{h,j}$, the turbine flow rate w_t, and the turbine power P_t. We then define the following constants: the cold flow rate input w_c, the mass and specific heat capacity M_w and c_w of the walls, the specific heat capacity of the cold and hot fluids $c_{v,c}$, $c_{p,c}$ and $c_{p,h}$, the cold and hot side inlet temperatures $T_{c,0}$ and $T_{h,0}$, the nominal heat transfer coefficients γ_c and γ_h on the cold and hot sides, the Stodola coefficient K_t, the cold fluid gas constant R_c, and the turbine outlet pressure p_o. The reduced model then becomes:

$$0 = w_c - w_t \tag{17}$$

$$0 = w_c c_{p,c}(T_{c,j} - T_{c,j+1}) - Q_{c,j} \tag{18}$$

$$0 = w_h c_{p,h}(T_{h,j} - T_{h,j+1}) - Q_{h,j} \tag{19}$$

$$\frac{M_w c_w}{N}\frac{dT_{w,j}}{dt} = Q_{c,j} + Q_{h,N-j+1} \tag{20}$$

$$Q_{c,j} = \gamma_c \left(\frac{w_c}{w_{c,n}}\right)^{\beta} \left(\frac{T_{c,j} + T_{c,j-1}}{2} - T_{w,j}\right) \tag{21}$$

$$Q_{h,j} = \gamma_h \left(\frac{w_h}{w_{h,n}}\right)^{\beta} \left(\frac{T_{h,j} + T_{h,j-1}}{2} - T_{w,N-j+1}\right) \tag{22}$$

$$w_t = K_t \frac{p_c}{\sqrt{R_c T_{c,N}}} \sqrt{1 - \left(\frac{p_o}{p_c}\right)^2} \tag{23}$$

$$P_t = w_t c_p \eta c_p T_{c,N} \left(1 - \left(\frac{p_o}{p_c}\right)^{c_p/c_v}\right), \tag{24}$$

with $j = 1,\ldots,N = 6$. This model only has 33 differential-algebraic equations and 6 state variables, compared to the 2100 differential algebraic equations and 80 scalar state variables of the original model; yet, its step response, shown by the dotted blue line in Fig. 2, is a very good approximation of the response of the original model, as suggested by the outcome of Step 4 of the proposed method.

TABLE V:
w_{cold} VS P_{el} @ 0.03 rad/s

State	Error [%]
coldside.p	16.01
coldside.T[6]	3.21
wall.T[6]	0.07
hotside.T[6]	0.00
hotside.p	0.00
coldside.X[6,5]	0.00
hotside.X[6,5]	0.00

Of course if we also got rid of the $T_{w,j}$ states via singular perturbation, we would have obtained a completely static model, whose response, shown in light blue in Fig. 2 is a quite bad approximation of the original one over this time/frequency scale.

B. Cold flow rate perturbation

The analysis is performed at frequency $\omega^* = 1/\tau_2 \approx 0.03\,\text{rad/s}$. In this case, the outcome of Step 2, shown in Table V, is pretty clear: a model that only considers the cold side pressure p_c as state variable and all other variables as constants can lead to an acceptable error of 16%. In this case there is no need to proceed with Step 3, since this would lead to a completely static model.

The reduced-order first-principle model can only have one differential equation in the variable p_c, and consider all other states as constants; this will of course be the cold side fluid mass balance equation, written under the assumption of constant temperatures and compositions, which is then coupled with the turbine equations:

$$\sum_1^N \frac{V_c}{N R_c T_{c,j}} \frac{dp_c}{dt} = w_c - w_t \tag{25}$$

$$w_t = K_t \frac{p_c}{\sqrt{R_c T_{c,N}}} \sqrt{1 - \left(\frac{p_o}{p_c}\right)^2} \tag{26}$$

$$P_t = w_t c_p \eta c_p T_{c,N} \left(1 - \left(\frac{p_o}{p_c}\right)^{c_p/c_v}\right), \tag{27}$$

where V_c is the volume of the boiler cold channels.

In this case, the reduced-order model only has 3 differential-algebraic equations and 1 state, but yet, as shown in Fig. 3, it reproduces the system step response quite well for the first 30-50 seconds, which roughly correspond to the selected value of ω^*; over this time horizon, the wall temperature can be assumed as a constant. On a longer time scale, thermal phenomena kick in: since the cold mass flow was increased, but the hot mass flow was not, the cold side outlet temperature decreases over the same time scale of the previous case, causing a slight reduction in the power output. However, this slower dynamics is not too relevant if one wants to close a feedback loop with crossover frequency around 0.03 rad/s.

If we eliminated the p_c state in Step 3 by singular perturbation, i.e, by replacing the left-hand-side of Equation (25) with zero, we would obtain a static model, whose response is

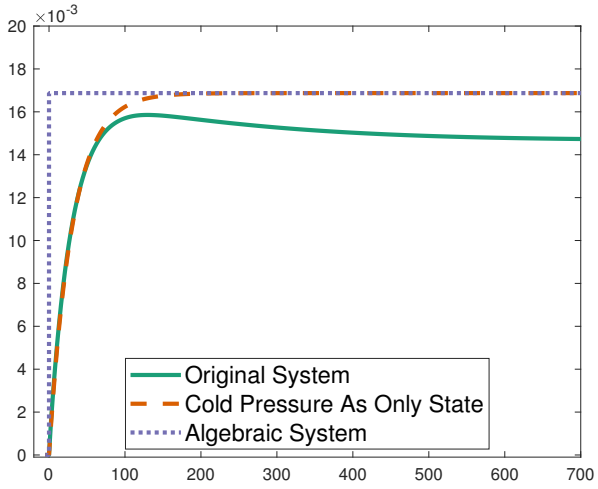

Fig. 3: Responses of the simplified system after a cold mass flow rate step through subsequent iterations.

shown by the dotted line in Fig. 3, which is not acceptable if we are interested in a time horizon of a few tens of seconds.

IV. EXAMPLE 2: SOS-CO_2 CYCLE

In this section the algorithm is applied to the SOS-CO_2 cycle [13]. This power plant is an innovative technology featuring an oxy-combustion closed recuperated Brayton cycle and a solid-oxide fuel cell. Through a train of compressors (CPR), intercoolers (IC) and water condensers (WC), the recycle mass flow rate, mainly composed by CO_2 is compressed and split into the moderator and the oxidant flows. Both streams are pre-heated in regenerators (R and O) and the oxidant flow, mixed with pure oxygen, oxidizes the injected fuel first in a solid-oxide fuel cell and then in the oxy-combustor (COM). The flue gases are expanded in a turbine and serve as hot flows in the regenerators. The excess CO_2 flow produced is extracted from the recycle flow and is sent to a compression and purification unit for permanent underground storage.

The model of this thermal power generation process was coded in the Modelica language (see Fig. 4) and features more than 110000 equations and about 1400 scalar state variables. The prohibitive size and the complexity of the model makes the formulation of control-oriented models very difficult. Useful indications may then be obtained by processing the plant model with the proposed algorithm.

Table VI presents the result of the analysis considering as input-output pair the flow rate of compressor CPR-4 and the net plant power output, i.e., the turbine and fuel cell power output minus compressor consumption. The inspected frequency is $0.07\,\mathrm{rad/s}$, since it is a reasonable crossover frequency for a power output controller.

It is immediately possible to recognize three groups of states. The pressures of the components in the low pressure side of the cycle (first 6 rows of the table) are essential to capture correctly the dynamic phenomena, in particular, the intercooler pressures, since they determine the overall

TABLE VI:
w_{CPR4} VS P_{el} @ $0.07\,\mathrm{rad/s}$

State	Error [%]
IC-1.p	100.00
O.hotside.p[10]	100.00
R.hotside.p[10]	86.90
IC-2.p	77.89
IC-3.p	71.61
IC-4.p	42.44
O.hotside.T[10,4]	23.00
IC-5.p	23.00
R.coldside.p[10]	22.89
COM.p	22.89
FC.anode.p[10]	22.89
FC.anode.X[10,10]	22.89
FC.cathode.p[10]	25.67
R.hotside.T[10,4]	10.20
O.coldside.p[10]	15.16
COM.fluegas.T	11.55
O.coldside.T[10,4]	8.76
COM.fluegas.X[5]	5.85
IC-2.X[5]	5.06
O.hotside.X[10,4,10]	5.02
IC-1.X[5]	3.77
FC.anode.T[10]	1.50
FC.cathode.T[10]	1.66
R.hotside.X[10,4,10]	2.19
FC.cathode.X[10,10]	0.45
R.coldside.T[10,4]	1.97
R.wall.T[10,4]	0.82
O.wall.T[10,4]	0.73
IC-3.X[5]	1.43
FC.wall.T[10]	0.66
IC-4.X[5]	0.24
IC-5.X[5]	0.21
R.coldside.X[10,4,10]	0.19
O.coldside.X[10,4,10]	0.19
FC.anodePlate.T[10]	0.07
FC.cathodePlate.T[10]	0.00

TABLE VII:
w_{CPR4} VS P_{el}: Type 2 states

State	Error [%]
R.hotside.p[10]	74.85
R.coldside.p[10]	53.20
FC.anode.p[10]	41.61
O.hotside.p[10]	33.03
O.coldside.p[10]	27.85
IC-3.p	24.20
IC-5.p	19.89
IC-2.p	16.31
IC-4.p	14.12
FC.cathode.p[10]	11.65
IC-1.p	9.97
COM.fluegas.p	9.10
R.hotside.T[10,4]	8.78
O.hotside.T[10,4]	8.70
FC.anode.X[10,10]	8.70
COM.fluegas.T	8.71
O.coldside.T[10,4]	8.76

condensed water flow leaving the cycle from the condensers; this flow rate influences the mass flow rate through the downstream compressors, hence their consumption, but also the turbine expansion ratio, hence the turbine power output.

The next 8-10 states comprise some of the most relevant states of the high pressure side and the temperatures of the regenerators hot sides. Their influence is less important since temperatures act on a longer time scale and many of the pressures are not directly influenced by the mass flow rate variation due to the presence of compressor CPR-5, which follows a mass flow rate setpoint.

All the remaining state variables, in particular the fluid compositions, have a marginal impact on the accuracy of the reduced-order model.

As in the previous examples, Steps 3 and 4 can further reduce the order of the simplified model. Assuming that the decision of Step 2 is to retain the first 17 states (note that many of those states are arrays, so the number of scalar state variables is much larger), Step 3 leads to Table VII. This clearly shows that the temperature states may be converted into algebraic variables by singular perturbation without increasing the approximation error, leading to reduced models that only consider 12 pressure state variables and still provide an error less than 10% in the frequency response $G(j\omega^*)$

Simplified models based on these results are currently

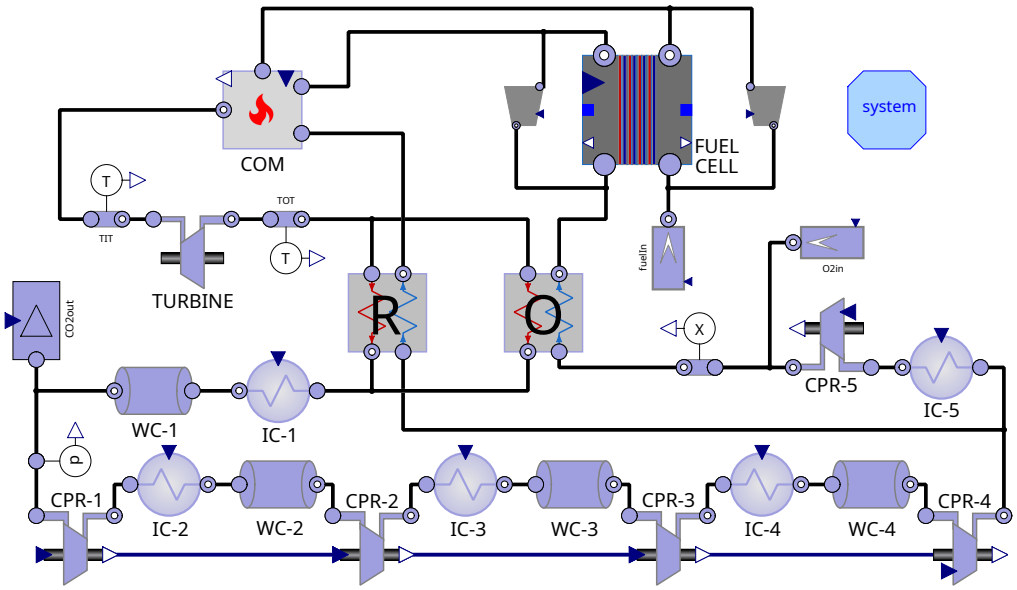

Fig. 4: The SOS-CO_2 cycle Modelica process diagram.

under development and will be presented in future works.

V. CONCLUSION

In this work the authors propose a novel method to support the formulation of control-oriented reduced models of thermal power generation processes, based on two heuristic algorithms and two expert decision steps.

The procedure starts from a detailed, accurate simulation model of the process, an indication of a steady-state operating point, and a frequency of interest. It then provides a categorization of its states into three classes: essential state variables, states that may be converted into algebraic variables and states that might be considered as constants. The expert modeller can then use these suggestions as a guide to formulate first-principle reduced-order models for control purposes, which capture the control-relevant dynamics in a white-box fashion.

The proposed method was tested on two different systems: a conceptual model of a sCO_2 boiler-turbine system and the (very large) model of the innovative SOS-CO_2 power generation system. The results of the tests are consistent with the expected result in the first case and seem very promising in the second case despite the large size of the accurate dynamic model of the system, though they certainly need to be tested on more use cases to confirm the general validity of the proposed approach. The Python code of the proposed method and the Modelica code of the simplified models is available on the linked GitHub repository [11].

A simple extension of the proposed method is to consider the frequency response approximation error over a finite frequency range, instead of a single value, and/or around multiple operating points instead of one, to ensure that the choice of states will lead to accurate enough models over a wider operating range than the currently assessed one.

ACKNOWLEDGMENT

This work is supported by the Italian Ministry of University and Research under the PONRI program (Programma Operativo Nazionale "Ricerca e Innovazione" 2014-2020).

REFERENCES

[1] M. Pathak *et al.*, "Climate change 2022: Mitigation of climate change. technical summary contribution of Working Group III to the sixth assessment report of the intergovernmental panel on climate change," Tech. Rep.

[2] IEA, "World Energy Outlook 2023," https://www.iea.org/reports/world-energy-outlook-2023, 2023.

[3] S. E. Mattsson, H. Elmqvist, and M. Otter, "Physical system modeling with Modelica," *Control Engineering Practice*, vol. 6, no. 4, pp. 501–510, 1998.

[4] F. Casella and A. Leva, "Object-oriented modelling & simulation of power plants with modelica," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 7597–7602.

[5] A. Vojacek *et al.*, "Status of the ClaRa Library: Detailed transient simulation of complex energy systems," in *Proc. 15th International Modelica Conference*, Aachen, Germany, Oct 9–11 2023.

[6] D. B. Baligh El-Hefni, *Modeling and Simulation of Thermal Power Plants with ThermoSysPro: A Theoretical Introduction and a Practical Guide*. Springer, 2019.

[7] C. Maffezzoni, "Boiler-turbine dynamics in power-plant control," *Control Engineering Practice*, vol. 5, no. 3, pp. 301–312, 1997.

[8] F. E. Cellier and E. Kofman, *Continuous system simulation*. Springer Science & Business Media, 2006.

[9] P. V. Kokotovic, R. E. O'Malley Jr, and P. Sannuti, "Singular perturbations and order reduction in control theory—an overview," *Automatica*, vol. 12, no. 2, pp. 123–132, 1976.

[10] P. Virtanen *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[11] M. De Pascali and F. Casella, "ProcModRed," https://github.com/looms-polimi/ProcModRed, 2024.

[12] D. Alfani, M. Astolfi, M. Binotti, S. Campanari, F. Casella, and P. Silva, "Multi Objective Optimization of Flexible Supercritical CO2 Coal-Fired Power Plants," in *ASME Turbo Expo 2019: Turbomachinery Technical Conference and Exposition*. American Society of Mechanical Engineers Digital Collection, 2019.

[13] R. Scaccabarozzi, M. Gatti, S. Campanari, and E. Martelli, "Solid oxide semi-closed CO2 cycle: A hybrid power cycle with 75% net efficiency and zero emissions," *Applied Energy*, vol. 290, p. 116711, 2021.