
Python Anesthesia Simulator

Release 0.0.4

Bob Aubouin-Pairault

Apr 28, 2023

CONTENTS:

1	src.python_anesthesia_simulator package	1
1.1	Submodules	1
1.2	src.python_anesthesia_simulator.simulator module	1
1.3	src.python_anesthesia_simulator.pk_models module	6
1.4	src.python_anesthesia_simulator.pd_models module	9
1.5	src.python_anesthesia_simulator.disturbances module	15
1.6	src.python_anesthesia_simulator.metrics module	16
	Bibliography	19
	Python Module Index	21
	Index	23

SRC.PYTHON_ANESTHESIA_SIMULATOR PACKAGE

1.1 Submodules

1.2 src.python_anesthesia_simulator.simulator module

```
class Patient(patient_characteristic: list, co_base: float = 6.5, map_base: float = 90, model_propo: str =  
    'Schnider', model_remi: str = 'Minto', ts: float = 1, hill_param: Optional[list] = None,  
    random_PK: bool = False, random_PD: bool = False, co_update: bool = False, save_data_bool:  
    bool = True)
```

Bases: object

Define a Patient class able to simulate Anesthesia process.

Parameters

Patient_characteristic: list

Patient_characteristic = [age (yr), height(cm), weight(kg), gender(0: female, 1: male)]

co_base

[float, optional] Initial cardiac output. The default is 6.5L/min.

map_base

[float, optional] Initial Mean Arterial Pressure. The default is 90mmHg.

model_propo

[str, optional] Name of the Propofol PK Model. The default is 'Schnider'.

model_remi

[str, optional] Name of the Remifentanyl PK Model. The default is 'Minto'.

ts

[float, optional] Sampling time (s). The default is 1.

BIS_param

[list, optional] Parameter of the BIS model (Propo Remi interaction) list [C50p_BIS, C50r_BIS, gamma_BIS, beta_BIS, E0_BIS, Emax_BIS]. The default is None.

random_PK

[bool, optional] Add uncertainties in the Propofol and Remifentanyl PK models. The default is False.

random_PD

[bool, optional] Add uncertainties in the BIS PD model. The default is False.

co_update

[bool, optional] Turn on the option to update PK parameters thanks to the CO value. The default is False.

save_data_bool

[bool, optional] Save all interns variable at each sampling time in a data frame. The default is True.

Attributes

age

[float] Age of the patient (yr).

height

[float] Height of the patient (cm).

weight

[float] Weight of the patient (kg).

gender

[bool] 0 for female, 1 for male.

co_base

[float] Initial cardiac output (L/min).

map_base

[float] Initial mean arterial pressure (mmHg).

ts

[float] Sampling time (s).

model_propo

[str] Name of the propofol PK model.

model_remi

[str] Name of the remifentanil PK model.

hill_param

[list] Parameter of the BIS model (Propo Remi interaction) list [C50p_BIS, C50r_BIS, gamma_BIS, beta_BIS, E0_BIS, Emax_BIS].

random_PK

[bool] Add uncertainties in the Propodfol and Remifentanil PK models.

random_PD

[bool] Add uncertainties in the BIS PD model.

co_update

[bool] Turn on the option to update PK parameters thanks to the CO value.

save_data_bool

[bool] Save all interns variable at each sampling time in a data frame.

lbm

[float] Lean body mass (kg).

propo_pk

[CompartmentModel] 6-comparments model for Propofol.

remi_pk

[CompartmentModel] 5-comparments model for Remifentanil.

nore_pk

[CompartmentModel] 1-comparments model for Norepinephrine.

bis_pd
[BIS_model] Surface-response model for bis computation.

tol_pd
[TOL_model] Hierarchical model for TOL computation.

hemo_pd
[Hemo_PD_model] Hemodynamic model for CO and MAP computation.

data
[pd.DataFrame] Dataframe containing all the intern variables at each sampling time.

bis
[float] Bispectral index (%).

tol
[float] Tolerance of laryngospie probability (0-1).

co
[float] Cardiac output (L/min).

map
[float] Mean arterial pressure (mmHg).

blood_volume
[float] Blood volume (L).

Methods

<i>blood_loss</i> ([fluid_rate])	Actualize the patient parameters to mimic blood loss.
<i>find_bis_equilibrium_with_ratio</i> (bis_target)	Find the input of Propofol and Remifentanil to meet the BIS target at the equilibrium with a fixed ratio between drugs rates.
<i>find_equilibrium</i> (bis_target, tol_target, ...)	Find the input to meet the targeted outputs at the equilibrium.
<i>full_sim</i> ([u_propo, u_remi, u_nore])	Simulate the patient model with the given inputs.
<i>init_dataframe</i> ()	Initilize the dataframe variable.
<i>initialized_at_given_input</i> ([u_propo, ...])	Initialize the patient Simulator at the given input as an equilibrium point.
<i>initialized_at_maintenance</i> (bis_target, ...)	Initialize the patient model at the equilibrium point for the given output value.
<i>one_step</i> ([u_propo, u_remi, u_nore, ...])	Simulate one step time of the patient.
<i>save_data</i> ([inputs])	Save all current intern variable as a new line in self.dataframe.

blood_loss(*fluid_rate: float = 0*)

Actualize the patient parameters to mimic blood loss.

Parameters

fluid_rate

[float, optional] Fluid rates from blood volume (mL/min), negative is bleeding while positive is a transfusion. The default is 0.

Returns

None.

find_bis_equilibrium_with_ratio(*bis_target*: float, *rp_ratio*: float = 2) → tuple[float, float]

Find the input of Propofol and Remifentanil to meet the BIS target at the equilibrium with a fixed ratio between drugs rates.

Solve the optimization problem:

$$J = (bis - bis_{target})^2$$

Where *bis* is the BIS computed from the pharmacodynamic model. And with the constraints:

$$u_{propo} = u_{remi} * rp_{ratio}$$

$$A_{propo}x_{propo} + B_{propo}u_{propo} = 0$$

$$A_{remi}x_{remi} + B_{remi}u_{remi} = 0$$

Parameters

bis_target

[float] BIS target (%).

rp_ratio

[float] remifentanil over propofol rates ratio. The default is 2.

Returns

u_propo

[float:] Propofol infusion rate (mg/s).

u_remi

[float:] Remifentanil infusion rate (μg/s).

find_equilibrium(*bis_target*: float, *tol_target*: float, *map_target*: float) → tuple[float, float, float]

Find the input to meet the targeted outputs at the equilibrium.

Solve the optimization problem to find the equilibrium input for BIS - TOL:

$$\min_{C_{p,es}, C_{r,es}} \frac{||BIS_{target} - BIS||^2}{100^2} + ||TOL_{target} - TOL||^2$$

Then compute the concentration of Noradrenaline to meet the MAP target.

Finally, compute the input of Propofol, Remifentanil and Noradrenaline to meet the targeted concentration.

Parameters

bis_target

[float] BIS target (%).

tol_target

[float] TOL target ([0, 1]).

map_target:float

MAP target (mmHg).

Returns

u_propo

[float:] Propofol infusion rate (mg/s).

u_remi

[float:] Remifentanil infusion rate (μg/s).

u_nore

[float:] Norepinephrine infusion rate (μg/s).

full_sim(*u_propo: Optional[list] = None, u_remi: Optional[list] = None, u_nore: Optional[list] = None*)

Simulate the patient model with the given inputs.

Parameters

u_propo

[list] Propofol infusion rate (mg/s).

u_remi

[list] Remifentanyl infusion rate ($\mu\text{g/s}$).

u_nore

[list] Norepinephrine infusion rate ($\mu\text{g/s}$).

Returns

pandas.Dataframes

Dataframe with all the data.

init_dataframe()

Initilize the dataframe variable.

initialized_at_given_input(*u_propo: float = 0, u_remi: float = 0, u_nore: float = 0*)

Initialize the patient Simulator at the given input as an equilibrium point.

For each drug, the equilibrium state is computed from the input. Then this state is used to initialize each drug pharmacokinetic model.

Parameters

u_propo

[float, optional] Propofol infusion rate (mg/s). The default is 0.

u_remi

[float, optional] Remifentanyl infusion rate ($\mu\text{g/s}$). The default is 0.

u_nore

[float, optional] Norepinephrine infusion rate ($\mu\text{g/s}$). The default is 0.

Returns

None.

initialized_at_maintenance(*bis_target: float, tol_target: float, map_target: float*) \rightarrow tuple[float, float, float]

Initialize the patient model at the equilibrium point for the given output value.

Parameters

bis_target

[float] BIS target (%).

rass_target

[float] RASS target ([0, -5]).

map_target:float

MAP target (mmHg).

Returns

u_propo

[float:] Propofol infusion rate (mg/s).

u_remi

[float:] Remifentanyl infusion rate ($\mu\text{g/s}$).

u_nore

[float:] Norepinephrine infusion rate ($\mu\text{g/s}$).

one_step(*u_propo*: float = 0, *u_remi*: float = 0, *u_nore*: float = 0, *blood_rate*: float = 0, *dist*: list = [0, 0, 0], *noise*: bool = True) → tuple[float, float, float, float]

Simulate one step time of the patient.

Parameters

u_propo

[float, optional] Propofol infusion rate (mg/s). The default is 0.

u_remi

[float, optional] Remifentanyl infusion rate ($\mu\text{g/s}$). The default is 0.

u_nore

[float, optional] Norepinephrine infusion rate ($\mu\text{g/s}$). The default is 0.

blood_rate

[float, optional] Fluid rates from blood volume (mL/min), negative is bleeding while positive is a transfusion. The default is 0.

Dist

[list, optional] Disturbance vector on [BIS (%), MAP (mmHg), CO (L/min)]. The default is [0]*3.

noise

[bool, optional] bool to add measurement noise on the outputs. The default is True.

Returns

bis

[float] Bispectral index(%).

co

[float] Cardiac output (L/min).

map

[float] Mean arterial pressure (mmHg).

tol

[float] Tolerance of Laryngoscopy index (0-1).

save_data(*inputs*: list = [0, 0, 0])

Save all current intern variable as a new line in self.dataframe.

1.3 src.python_anesthesia_simulator.pk_models module

class CompartmentModel(*Patient_characteristic*: list, *lbm*: float, *drug*: str, *model*: Optional[str] = None, *ts*: float = 1, *random*: bool = False, *x0*: Optional[list] = None, *opiate*=True, *measurement*='arterial')

Bases: object

PKmodel class modelize the PK model of propofol or remifentanyl drug. Simulate the drug distribution in the body.

Use a 6 compartement model for propofol, a 5 compartement model for remifentanyl, and a 1 compartement model for norepinephrine. The model is a LTI model. The state vector is the concentration of the drug in each compartement.

Parameters

Patient_characteristic: list

Patient_characteristic = [age (yr), height(cm), weight(kg), gender(0: female, 1: male)]

lbm

[float] lean body mass index.

drug

[str] can be “Propofol”, “Remifentanyl” or “Norepinephrine”.

model

[str, optional] Could be “Schnider” [1], “Marsh_initial”[2], “Marsh_modified”[3], “Shuttler”[4] or “Elefeld”[5] for Propofol. “Minto”[6], “Elefeld”[7] for Remifentanyl. only “Beloeil”[8] for Norepinephrine. The default is “Minto” for Remifentanyl and “Schnider” for Propofol.

ts

[float, optional] Sampling time, in s. The default is 1.

random

[bool, optional] bool to introduce uncertainties in the model. The default is False.

x0

[list, optional] Initial concentration of the compartement model. The default is `np.ones([4, 1])*1e-4`.

opiate

[bool, optional] For Elefeld model for propofol, specify if there is a co-administration of opiate (Remifentanyl) in the same time. The default is True.

measurement

[str, optional] For Elefeld model for propofol, specify the measurement place for blood concentration. Can be either ‘arterial’ or ‘venous’. The default is ‘arterial’.

References

[1], [2], [3], [4], [5], [6], [7], [8]

Attributes

ts

[float] Sampling time, in s.

drug

[str] can be “Propofol”, “Remifentanyl” or “Norepinephrine”.

A_init

[list] Initial value of the matrix A.

B_init

[list] Initial value of the matrix B.

v1

[float] Volume of the first compartement.

continuous_sys

[control.StateSpace] Continuous state space model.

discrete_sys
[control.StateSpace] Discrete state space model.

x
[list] State vector.

y
[list] Output vector (hypnotic effect site concentration).

Methods

<code>full_sim(u)</code>	Simulate PK model with a given input.
<code>one_step(u)</code>	Simulate one step of PK model.
<code>update_param_CO(CO_ratio)</code>	Update PK coefficient with a linear function of Cardiac output value.
<code>update_param_blood_loss(v_ratio)</code>	Update PK coefficient to mimic a blood loss.

full_sim(*u: list*) → list
Simulate PK model with a given input.

Parameters

u
[list] Infusion rate (mg/s for Propofol, µg/s for Remifentanyl and Norepinephrine).

Returns

numpy array
List of the states value during the simulation. (µg/mL for Propofol and ng/mL for Remifentanyl and Norepinephrine).

one_step(*u: float*) → list
Simulate one step of PK model.

$$x^+ = Ax + Bu$$

Parameters

u
[float] Infusion rate (mg/s for Propofol, µg/s for Remifentanyl).

Returns

numpy array
Actual effect site concentration (µg/mL for Propofol and ng/mL for Remifentanyl and Norepinephrine).

update_param_CO(*CO_ratio: float*)
Update PK coefficient with a linear function of Cardiac output value.

Parameters

CO
[float] Ratio of Current CO relatively to initial CO.

Returns

None.

update_param_blood_loss(*v_ratio*: float)

Update PK coefficient to mimic a blood loss.

Update the blood volume compartment

Parameters

v_ratio

[float] blood volume as a fraction of init volume, 1 mean no loss, 0 mean 100% loss.

Returns

None.

1.4 src.python_anesthesia_simulator.pd_models module

class BIS_model(*hill_model*: str = 'Bouillon', *hill_param*: Optional[list] = None, *random*: bool = False)

Bases: object

Surface Response model to link Propofol and Remifentanil blood concentration to BIS.

equation:

$$BIS = E0 + Emax * \frac{U^\gamma}{1 + U^\gamma}$$

$$U = \frac{U_p + U_r}{1 - \beta\theta + \beta\theta^2}$$

$$U_p = \frac{C_{p,es}}{C_{p,50}}$$

$$U_r = \frac{C_{r,es}}{C_{r,50}}$$

$$\theta = \frac{U_p}{U_r + U_p}$$

Parameters

hill_model

[str, optional] Only 'Bouillon' [9] is available. Ignored if hill_param is specified.

hill_param

[list, optional] Parameter of the Hill model (Propo Remi interaction) list [c50p_BIS, c50r_BIS, gamma_BIS, beta_BIS, E0_BIS, Emax_BIS]:

- c50p_BIS : Concentration at half effect for propofol effect on BIS (µg/mL)
- c50r_BIS : Concentration at half effect for remifentanil effect on BIS (ng/mL)
- gamma_BIS : slope coefficient for the BIS model,
- beta_BIS : interaction coefficient for the BIS model,
- E0_BIS : initial BIS,
- Emax_BIS : max effect of the drugs on BIS.

The default is None.

random

[bool, optional] Add uncertainties in the parameters. Ignored if Hill_cruv is specified. The default is False.

ts

[float, optional] Sampling time, in s. The default is 1.

References

[9]

Attributes

c50p

[float] Concentration at half effect for propofol effect on BIS ($\mu\text{g/mL}$).

c50r

[float] Concentration at half effect for remifentanil effect on BIS (ng/mL).

gamma

[float] slope coefficient for the BIS model.

beta

[float] interaction coefficient for the BIS model.

E0

[float] initial BIS.

E_{max}

[float] max effect of the drugs on BIS.

hill_param

[list] Parameter of the Hill model (Propo Remi interaction) list [c50p_BIS, c50r_BIS, gamma_BIS, beta_BIS, E0_BIS, Emax_BIS]

c50p_init

[float] Initial value of c50p, used for blood loss modelling.

Methods

<code>compute_bis(c_es_propo, c_es_remi)</code>	Compute BIS function from Propofol and Remifentanil effect site concentration.
<code>inverse_hill(BIS[, c_es_remi])</code>	Compute Propofol effect site concentration from BIS and Remifentanil Effect site concentration.
<code>plot_surface()</code>	Plot the 3D-Hill surface of the BIS related to Propofol and Remifentanil effect site concentration.
<code>update_param_blood_loss(v_ratio)</code>	Update PK coefficient to mimic a blood loss.

compute_bis(*c_es_propo*: float, *c_es_remi*: float) → float

Compute BIS function from Propofol and Remifentanil effect site concentration.

Parameters

cep

[float] Propofol effect site concentration $\mu\text{g/mL}$.

cer

[float] Remifentanil effect site concentration ng/mL

Returns**BIS**

[float] Bis value.

inverse_hill(*BIS: float, c_es_remi: float = 0*) → float

Compute Propofol effect site concentration from BIS and Remifentanil Effect site concentration.

Parameters**BIS**

[float] BIS value.

cer

[float, optional] Effect site Remifentanil concentration (ng/mL). The default is 0.

Returns**cep**

[float] Effect site Propofol concentration (µg/mL).

plot_surface()

Plot the 3D-Hill surface of the BIS related to Propofol and Remifentanil effect site concentration.

update_param_blood_loss(*v_ratio: float*)

Update PK coefficient to mimic a blood loss.

Update the c50p parameters thanks to the blood volume ratio.

Parameters**v_loss**

[float] blood volume as a fraction of init volume, 1 mean no loss, 0 mean 100% loss.

Returns**None.**

class Hemo_PD_model(*nore_param: Optional[list] = None, propo_param: Optional[list] = None, remi_param: Optional[list] = None, random: bool = False, co_base: float = 6.5, map_base: float = 90*)

Bases: object

Modelize the effect of Propofol, Remifentanil, Norepinephrine on Mean Arterial Pressure and Cardiac Output.

Use the addition of sigmoid curve to model the effect of each drugs on MAP and CO. The following articles are used to define the parameters of the model:

- Norepinephrine to MAP: [10]
- Noepinephrine to CO: [11]
- Propofol to MAP: [12]
- Propofol to CO: [13]
- Remifentanil to MAP: [14]
- Remifentanil to CO: [15]

Parameters**nore_param**

[list, optional] List of hill curve parameters for Norepinephrine action [Emax_map, c50_map, gamma_map, Emax_co, c50_co, gamma_co]. The default is None.

propo_param

[list, optional] List of hill curve parameters for Propofol action [emax_SAP, emax_DAP, c50_map_1, c50_map_2, gamma_map_1, gamma_map_2, Emax_co, c50_co, gamma_co]. The default is None.

remi_param

[list, optional] List of hill curve parameters for Relifentanil action [Emax_map, c50_map, gamma_map, Emax_co, c50_co, gamma_co]. The default is None.

random

[bool, optional] Add uncertainties in the parameters. The default is False.

co_base: float, optional

Baseline Cardiac output (L/min). The default is 6.5 L/min.

map_base: float, optional

Baseline mean arterial pressure (mmHg). The default is 90mmHg.

References

[10], [11], [12], [13], [14], [15]

Attributes

co_base

[float] Baseline cardiac output.

map_base

[float] Baseline mean arterial pressure.

emax_nore_map

[float] Maximal effect of Norepinephrine on MAP.

c50_nore_map

[float] Concentration of Norepinephrine that produce half of the maximal effect on MAP.

gamma_nore_map

[float] Slope of the sigmoid curve for Norepinephrine effect on MAP.

emax_nore_co

[float] Maximal effect of Norepinephrine on CO.

c50_nore_co

[float] Concentration of Norepinephrine that produce half of the maximal effect on CO.

gamma_nore_co

[float] Slope of the sigmoid curve for Norepinephrine effect on CO.

emax_propo_SAP

[float] Maximal effect of Propofol on SAP.

emax_propo_DAP

[float] Maximal effect of Propofol on DAP.

emax_propo_co

[float] Maximal effect of Propofol on CO.

c50_propo_map_1

[float] Concentration of Propofol that produce half of the maximal effect on MAP.

c50_propo_map_2

[float] Concentration of Propofol that produce half of the maximal effect on MAP.

gamma_propo_map_1

[float] Slope of the sigmoid curve for Propofol effect on MAP.

gamma_propo_map_2

[float] Slope of the sigmoid curve for Propofol effect on MAP.

c50_propo_co

[float] Concentration of Propofol that produce half of the maximal effect on CO.

gamma_propo_co

[float] Slope of the sigmoid curve for Propofol effect on CO.

emax_remi_map

[float] Maximal effect of Remifentanyl on MAP.

emax_remi_co

[float] Maximal effect of Remifentanyl on CO.

c50_remi_map

[float] Concentration of Remifentanyl that produce half of the maximal effect on MAP.

gamma_remi_map

[float] Slope of the sigmoid curve for Remifentanyl effect on MAP.

c50_remi_co

[float] Concentration of Remifentanyl that produce half of the maximal effect on CO.

gamma_remi_co

[float] Slope of the sigmoid curve for Remifentanyl effect on CO.

map

[float] Mean arterial pressure.

co

[float] Cardiac output.

Methods

<code>compute_hemo(c_es_propo, c_es_remi, c_es_nore)</code>	Compute current MAP and CO using addition of hill curv, one for each drugs.
---	--

compute_hemo(*c_es_propo*: list, *c_es_remi*: float, *c_es_nore*: float) → tuple[float, float]

Compute current MAP and CO using addition of hill curv, one for each drugs.

Parameters**c_es_propo**

[list] Propofol concentration on both hemodynamic effect site concentration µg/mL.

c_es_remi

[float] Remifentanyl hemodynamic effect site concentration µg/mL.

c_es_nore

[float] Norepinephrine hemodynamic effect site concentration µg/mL.

Returns**map**

[float] Mean arterial pressure (mmHg), without blood loss.

co

[float] Cardiac output (L/min), without blood loss.

class TOL_model(*model*: str = 'Bouillon', *model_param*: Optional[list] = None, *random*: bool = False)

Bases: object

Hierarchical model to link druf effect site concentration to Tolerance of Laryngoscopy.

The equation are:

$$postopioid = preopioid * \left(1 - \frac{C_{r,es}^{\gamma_r}}{C_{r,es}^{\gamma_r} + (C_{r,50}preopioid)^{\gamma_r}} \right)$$

$$TOL = \frac{C_{p,es}^{\gamma_p}}{C_{p,es}^{\gamma_p} + (C_{p,50}postopioid)^{\gamma_p}}$$

Parameters

model

[str, optional] Only 'Bouillon'[9] is available. Ignored if model_param is specified. The default is 'Bouillon'.

model_param

[list, optional] Model parameters, model_param = [c50p, c50r, gammaP, gammaR, Preopioid intensity]. The default is None.

random

[bool, optional] Add uncertainties in the parameters. Ignored if model_param is specified. The default is False.

References

[9]

Attributes

c50p

[float] Concentration at half effect for propofol effect on BIS (µg/mL).

c50r

[float] Concentration at half effect for remifentanil effect on BIS (ng/mL).

gamma_p

[float] Slope of the Hill function for propofol effect on TOL.

gamma_r

[float] Slope of the Hill function for remifentanil effect on TOL.

pre_intensity

[float] Preopioid intensity.

Methods

<code>compute_tol(c_es_propo, c_es_remi)</code>	Return TOL from Propofol and Remifentanil effect site concentration.
<code>plot_surface()</code>	Plot the 3D-Hill surface of the BIS related to Propofol and Remifentanil effect site concentration.

compute_tol(*c_es_propo*: float, *c_es_remi*: float) → float

Return TOL from Propofol and Remifentanil effect site concentration.

Compute the output of the Hirarchical model to predict TOL from Propofol and Remifentanil effect site concentration. TOL = 1 mean very relaxed and will tolerate laryngoscopy while TOL = 0 mean fully awake and will not tolerate.

Parameters

cep

[float] Propofol effect site concentration µg/mL.

cer

[float] Remifentanil effect site concentration ng/mL

Returns

TOL

[float] TOL value.

plot_surface()

Plot the 3D-Hill surface of the BIS related to Propofol and Remifentanil effect site concentration.

fsig(*x*, *c50*, *gam*)

1.5 src.python_anesthesia_simulator.disturbances module

compute_disturbances(*time*: float, *dist_profil*: str = 'realistic', *start_step*: float = 600, *end_step*: float = 1200) → list

Give the value of the disturbance profil for a given time.

Parameters

time

[float] Time: in seconde.

dist_profil

[str, optional] disturbance profil, can be: 'realistic', 'simple', 'step' or "null". The default is 'realistic'.

start_step

[float, optional] start time of the step disturbance (seconds). The default is 600s.

end_step

[float, optional] End time of the step disturbance (seconds). The default is 1200s.

Returns

list

dist_bis, dist_map, dist_co: respectively the additive disturbance to add to the BIS, MAP and CO signals.

1.6 src.python_anesthesia_simulator.metrics module

compute_control_metrics(time: list, bis: list, phase: str = 'maintenance', start_step: float = 600, end_step: float = 1200)

Compute metrics for closed loop anesthesia.

This function compute the control metrics initially proposed in [1].

Parameters

time

[list] List of time value.

bis

[list] List of BIS value over time.

ts

[float, optional] Sampling time in second. The default is 1.

phase

[str, optional] Control phase, can be “maintenance”, “induction” or “total”. The default is ‘maintenance’.

start_step: float, optional

Start time of the step disturbance, for maintenance and total phase. The default is 600s.

end_step: float, optional

End time of the step disturbance, for maintenance and total phase. The default is 1200s.

Returns

TT

[float] Observed time-to-target (in minute) required for reaching first time the target interval of [55,45] BIS values

BIS_NADIR: float

for “induction” or “total” phase. The lowest observed BIS value during induction phase

ST10: float

for “induction” or “total” phase. Settling time (in minute) on the reference BIS value, defined within ± 5 BIS(i.e., between 45 and 55 BIS)and stay within this BIS range

ST20: float

for “induction” or “total” phase. Settling time (in minute) on the reference BIS value, defined within ± 10 BIS(i.e., between 40 and 60 BIS) and stay within this BIS range

US: float

for “induction” or “total” phase. Undershoot, defined as the BIS value that exceeds the limit of the defined BIS interval, namely, the 45 BIS value.

TTp

[float] Time to target (in minute) after the positive step disturbance.

BIS_NADIRp: float

for “maintenance” or “total” phase. Minimum BIS value after the positive step disturbance.

TTpn: float

for “maintenance” or “total” phase. Time to target (in minute) after the negative step disturbance.

BIS_NADIRn: float

for “maintenance” or “total” phase. Maximum BIS value after the negative step disturbance.

References

[1]

BIBLIOGRAPHY

- [1] T. W. Schnider et al., “The Influence of Age on Propofol Pharmacodynamics,” *Anesthesiology*, vol. 90, no. 6, pp. 1502–1516., Jun. 1999, doi: 10.1097/00000542-199906000-00003.
- [2] B. Marsh, M. White, N. morton, and G. N. C. Kenny, “Pharmacokinetic model Driven Infusion of Propofol in Children,” *BJA: British Journal of Anaesthesia*, vol. 67, no. 1, pp. 41–48, Jul. 1991, doi: 10.1093/bja/67.1.41.
- [3] M. M. R. F. Struys et al., “Comparison of Plasma Compartment versus Two Methods for Effect Compartment-controlled Target-controlled Infusion for Propofol,” *Anesthesiology*, vol. 92, no. 2, p. 399, Feb. 2000, doi: 10.1097/00000542-200002000-00021.
- [4] J. Schüttler and H. Ihmsen, “Population Pharmacokinetics of Propofol: A Multicenter Study,” *Anesthesiology*, vol. 92, no. 3, pp. 727–738, Mar. 2000, doi: 10.1097/00000542-200003000-00017.
- [5] D. J. Eleveld, P. Colin, A. R. Absalom, and M. M. R. F. Struys, “Pharmacokinetic–pharmacodynamic model for propofol for broad application in anaesthesia and sedation” *British Journal of Anaesthesia*, vol. 120, no. 5, pp. 942–959, mai 2018, doi:10.1016/j.bja.2018.01.018.
- [6] C. F. Minto et al., “Influence of Age and Gender on the Pharmacokinetics and Pharmacodynamics of Remifentanyl: I. Model Development,” *Anesthesiology*, vol. 86, no. 1, pp. 10–23, Jan. 1997, doi: 10.1097/00000542-199701000-00004.
- [7] D. J. Eleveld et al., “An Allometric Model of Remifentanyl Pharmacokinetics and Pharmacodynamics,” *Anesthesiology*, vol. 126, no. 6, pp. 1005–1018, juin 2017, doi: 10.1097/ALN.0000000000001634.
- [8] H. Beloeil, J.-X. Mazoit, D. Benhamou, and J. Duranteau, “Norepinephrine kinetics and dynamics in septic shock and trauma patients,” *BJA: British Journal of Anaesthesia*, vol. 95, no. 6, pp. 782–788, Dec. 2005, doi: 10.1093/bja/aei259.
- [9] T. W. Bouillon et al., “Pharmacodynamic Interaction between Propofol and Remifentanyl Regarding Hypnosis, Tolerance of Laryngoscopy, Bispectral Index, and Electroencephalographic Approximate Entropy,” *Anesthesiology*, vol. 100, no. 6, pp. 1353–1372, Jun. 2004, doi: 10.1097/00000542-200406000-00006.
- [10] H. Beloeil, J.-X. Mazoit, D. Benhamou, and J. Duranteau, “Norepinephrine kinetics and dynamics in septic shock and trauma patients,” *BJA: British Journal of Anaesthesia*, vol. 95, no. 6, pp. 782–788, Dec. 2005, doi: 10.1093/bja/aei261.
- [11] X. Monnet, J. Jabot, J. Maizel, C. Richard, and J.-L. Teboul, “Norepinephrine increases cardiac preload and reduces preload dependency assessed by passive leg raising in septic shock patients” *Critical Care Medicine*, vol. 39, no. 4, p. 689, Apr. 2011, doi: 10.1097/CCM.0b013e318206d2a3.
- [12] C. Jelezcov, M. Lavielle, J. Schüttler, and H. Ihmsen, “Pharmacodynamic response modelling of arterial blood pressure in adult volunteers during propofol anaesthesia,” *BJA: British Journal of Anaesthesia*, vol. 115, no. 2, pp. 213–226, Aug. 2015, doi: 10.1093/bja/aeu553.

- [13] J. E. Fairfield, A. Dritsas, and R. J. Beale, “HAEMODYNAMIC EFFECTS OF PROPOFOL: INDUCTION WITH 2.5 MG KG-1,” *British Journal of Anaesthesia*, vol. 67, no. 5, pp. 618–620, Nov. 1991, doi: 10.1093/bja/67.5.618.
- [14] J. F. Standing, G. B. Hammer, W. J. Sam, and D. R. Drover, “Pharmacokinetic–pharmacodynamic modeling of the hypotensive effect of remifentanyl in infants undergoing cranioplasty,” *Pediatric Anesthesia*, vol. 20, no. 1, pp. 7–18, 2010, doi: 10.1111/j.1460-9592.2009.03174.x.
- [15] C. Chanavaz et al., “Haemodynamic effects of remifentanyl in children with and without intravenous atropine. An echocardiographic study,” *BJA: British Journal of Anaesthesia*, vol. 94, no. 1, pp. 74–79, Jan. 2005, doi: 10.1093/bja/aei293.
- [9] T. W. Bouillon et al., “Pharmacodynamic Interaction between Propofol and Remifentanyl Regarding Hypnosis, Tolerance of Laryngoscopy, Bispectral Index, and Electroencephalographic Approximate Entropy,” *Anesthesiology*, vol. 100, no. 6, pp. 1353–1372, Jun. 2004, doi: 10.1097/00000542-200406000-00006.
- [1] C. M. Ionescu, R. D. Keyser, B. C. Torrico, T. D. Smet, M. M. Struys, and J. E. Normey-Rico, “Robust Predictive Control Strategy Applied for Propofol Dosing Using BIS as a Controlled Variable During Anesthesia,” *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 9, pp. 2161–2170, Sep. 2008, doi: 10.1109/TBME.2008.923142.

PYTHON MODULE INDEX

S

`src.python_anesthesia_simulator.disturbances,`
 [15](#)
`src.python_anesthesia_simulator.metrics,` [16](#)
`src.python_anesthesia_simulator.pd_models,` [9](#)
`src.python_anesthesia_simulator.pk_models,` [6](#)
`src.python_anesthesia_simulator.simulator,` [1](#)

INDEX

B

BIS_model (class in *src.python_anesthesia_simulator.pd_models*), 9

blood_loss() (*Patient* method), 3

C

CompartmentModel (class in *src.python_anesthesia_simulator.pk_models*), 6

compute_bis() (*BIS_model* method), 10

compute_control_metrics() (in module *src.python_anesthesia_simulator.metrics*), 16

compute_disturbances() (in module *src.python_anesthesia_simulator.disturbances*), 15

compute_hemo() (*Hemo_PD_model* method), 13

compute_tol() (*TOL_model* method), 15

F

find_bis_equilibrium_with_ratio() (*Patient* method), 3

find_equilibrium() (*Patient* method), 4

fsig() (in module *src.python_anesthesia_simulator.pd_models*), module, 15

full_sim() (*CompartmentModel* method), 8

full_sim() (*Patient* method), 4

H

Hemo_PD_model (class in *src.python_anesthesia_simulator.pd_models*), 11

I

init_dataframe() (*Patient* method), 5

initialized_at_given_input() (*Patient* method), 5

initialized_at_maintenance() (*Patient* method), 5

inverse_hill() (*BIS_model* method), 11

M

module

src.python_anesthesia_simulator.disturbances, 15

src.python_anesthesia_simulator.metrics, 16

src.python_anesthesia_simulator.pd_models, 9

src.python_anesthesia_simulator.pk_models, 6

src.python_anesthesia_simulator.simulator, 1

O

one_step() (*CompartmentModel* method), 8

one_step() (*Patient* method), 6

P

Patient (class in *src.python_anesthesia_simulator.simulator*), 1

plot_surface() (*BIS_model* method), 11

plot_surface() (*TOL_model* method), 15

S

save_data() (*Patient* method), 6

src.python_anesthesia_simulator.disturbances

src.python_anesthesia_simulator.metrics module, 16

src.python_anesthesia_simulator.pd_models module, 9

src.python_anesthesia_simulator.pk_models module, 6

src.python_anesthesia_simulator.simulator module, 1

T

TOL_model (class in *src.python_anesthesia_simulator.pd_models*), 14

U

update_param_blood_loss() (*BIS_model* method), 11

update_param_blood_loss() (*CompartmentModel* method), 8

update_param_CO() (*CompartmentModel* method), 8