

# SPEKIFIKASI DAN VERIFIKASI SISTEM HIBRIDA

Belawati H. Widjaja\*



ID990000002

## ABSTRACT

**SPECIFICATION AND VERIFICATION OF HYBRID SYSTEMS.** Hybrid Systems are reactive systems which intermix between two components, discrete components and continuous components. The continuous components are usually called plants, subject to disturbances which cause the state variables of the systems changing continuously by physical laws and/or by the control laws. The discrete components can be digital computers, sensor and actuators controlled by programs. These programs are designed to select, control and supervise the behaviour of the continuous components. Specification and verification of hybrid systems has recently become an active area of research in both computer science and control engineering, many papers concerning hybrid system have been published. This paper gives a design methodology for hybrid systems as an example to the specification and verification of hybrid systems. The design methodology is based on the cooperation between two disciplines, control engineering and computer science. The methodology brings into the design of control loops and decision loops. The external behaviour of control loops are specified in a notation which is understandable by the two disciplines. The design of control loops which employed theory of differential equation is done by control engineers, and its correctness is also guaranteed analytically or experimentally by control engineers. The decision loops are designed in computing science based on the specifications of control loops. The verification of systems requirements can be done by computing scientists using a formal reasoning mechanism. For illustrating the proposed design, a problem of balancing an inverted pendulum which is a popular experiment device in control theory is considered, and the Mean Value Calculus is chosen as the formal notation for specifying the control loops and designing the decision loops.

## ABSTRAK

**SPEKIFIKASI DAN VERIFIKASI SISTEM HIBRIDA.** Sistem hibrida merupakan suatu sistem yang terdiri dari dua komponen, komponen diskret dan komponen kontinu. Komponen kontinu yang disebut *plant* dapat menyebabkan perubahan nilai-nilai dari variabel-variabel status (*state variables*) secara kontinu, tergantung pada hukum alam (*physical law*) dan hukum kendali (*control law*). Komponen diskretnya dapat berupa komputer, sensor dan *actuators* yang dikendalikan oleh program. Program-program ini dirancang untuk mengendalikan kelakuan dari komponen kontinunya. Spefikasi dan verifikasi sistem hibrida telah banyak dipelajari peneliti dan menghasilkan banyak publikasi. Pada tulisan ini diberikan suatu metodologi perancang suatu sistem hibrida berdasarkan kerja sama antar dua disiplin, yaitu disiplin *control engineering* dan disiplin *computer science*. Metodologi ini menghasilkan rancangan ikalan-ikalan kendali (*control loops*). Spefikasi kelakuan eksternal dari ikalan-ikalan kendali yang banyak memakai teori persamaan diferensial diserahkan kepada *control engineers* yang membuktikan kebenarannya ikalan-ikalan kendali tersebut secara analitik atau eksperimen. Sedangkan rancangan ikalan-ikalan keputusan diserahkan kepada *computing scientists* dan dirancang berdasarkan spefikasi

\* FMIPA Jurusan Matematika / Fakultas Ilmu Komputer - UI

dari ikalan-ikalan kendali tersebut. Begitu pula dengan verifikasi persyaratan sistem (*system requirements*) dapat dilakukan oleh *computing scientists* dengan menggunakan suatu mekanisme argumentasi formal (*formal reasoning mechanism*).

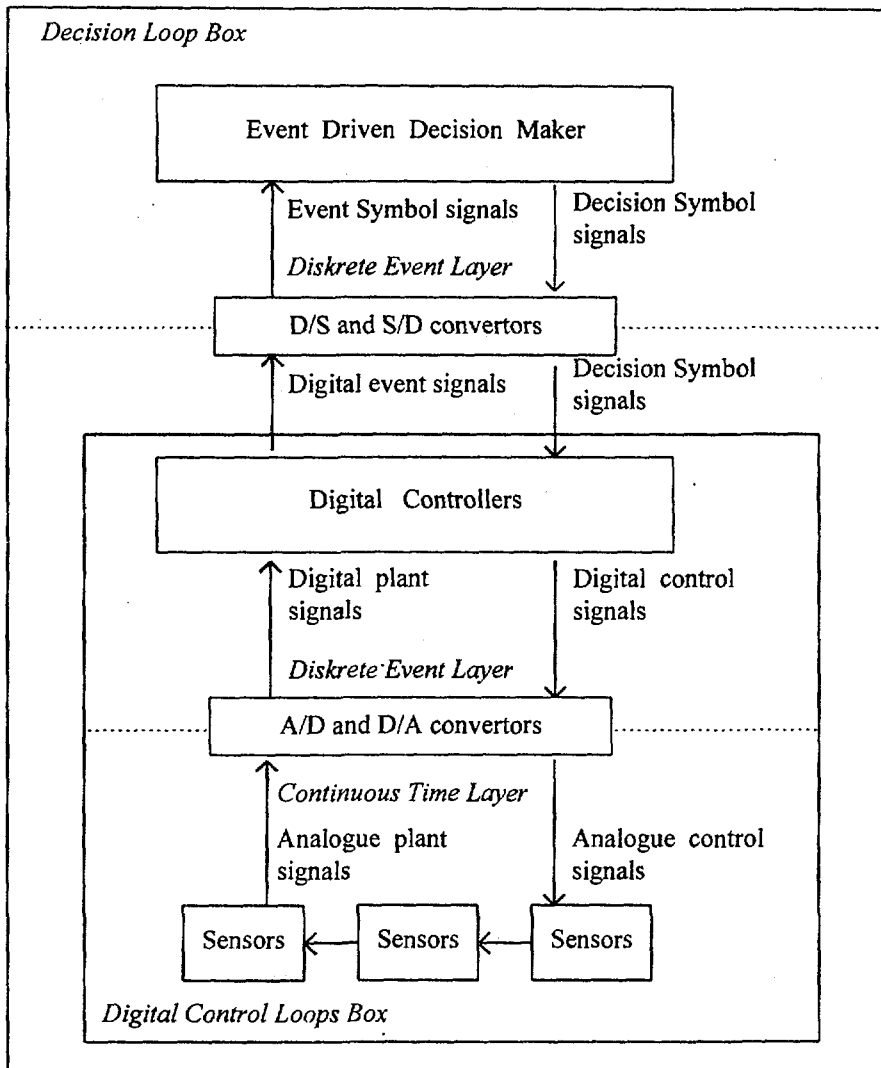
## PENDAHULUAN

Sistem hibrida merupakan suatu sistem yang terdiri dari dua komponen, komponen diskret dan komponen kontinu. Akhir-akhir ini, membuat rancangan atau model sistem hibrida (*hybrida systems*) dan implementasinya telah banyak menarik perhatian peneliti dibidang *control engineering*, *computer science* dan lainnya. Bagaimana sistem hibrida dianalisis? Bagaimana sistem hibrida disintesiskan? banyak pendekatan dilakukan oleh peneliti. Banyak penelitian telah memikirkan untuk memperluas isu-isu di bidang ilmu komputer tentang konkurensi (*concurrency*), *distributed computing* dan pengembangan dan verifikasi program ke sistem hibrida. Begitu pula isu-isu di bidang *control engineering*, misalnya, keteramatan (*observability*) dan keterkendalian (*controlability*) dapat diperluas kedalam sistem hibrida. Beberapa model untuk sistem hibrida telah diperkenalkan oleh beberapa peneliti. Manna dan Pnueli [16, 17] mengambil PTS (*Phase Transition Systems*) sebagai model sistem hibrida kemudian dikembangkan APTS (*Actor Phase Transition Systems*) sebagai model sistem kendali [19] Kapur, Henzinger, Manna dan Pnueli [10] memperkenalkan CPTS (*Concrete Phase Transition Systems*) sebagai sistem model hibrida. Alur, Courcoubetis, Henzinger, dan Ho [1], Alur, Courcoubetis, Halbwachs, Henzinger, Ho, Nicollin, Olivero, Sifakis dan Yovine [2], Norode dan Kohn [18] memperkenalkan *Finite Automata* sebagai model sistem hibrida. Kesten, Pnueli, Safakis dan Yovine [12] memperkenalkan *Integration Graphs* sebagai sistem model hibrida. Kesten dan Pnueli [11] memperkenalkan *Hybrid Statecharts* sebagai model sistem hibrida. Homman [8], Tehrani dan Stromberg [20] memperkenalkan model komposisi (*Composition Model*) untuk sistem hibrida. Fix dan Schneider [5] memperkenalkan **graf kendali** (*Control Graphs*) sebagai model sistem hibrida. Antsaklis, Stiver dan Lemmon [3], Lemmon, Stiver, Antsaklis [14], Widjaja, Chen, He dan Zhou [21] mengambil model *event-driven* untuk sistem hibrida. Beberapa bahasa spesifikasi (*specification language*) untuk sistem hibrida telah dipakai oleh beberapa peneliti. Manna dan Pnueli [17] memakai **logika temporal** (*Temporal Logic*) TL dengan jam (*Clock*) untuk spesifikasi dan verifikasi suatu sistem hibrida. Manna dan Pnueli [16], Maler, Manna dan Pnueli [15] memakai **logika temporal metrik** (*Metrik Temporal Logic*) MTL untuk spesifikasi dan verifikasi suatu sistem hibrida. Henzinger, Manna dan Pnueli [6], Kapur, Henzinger, Manna dan Pnueli [10] memakai **logika temporal hibrida** (*Hybrid Temporal Logic*) HTL untuk spesifikasi dan verifikasi suatu sistem hibrida. Lamport [13] memakai TLA+, yaitu bahasa spesifikasi berdasarkan TLA (*Temporal Logic of Action*) untuk spesifikasi dan

verifikasi suatu sistem hibrida. Zhou, Ravn dan Hansen [25], Yu, Wng, Zhou dan Pandya [22], Widjaja, Chen, He dan Zhou [21], Yu, Pandya, dan Sun [9] memakai **kalkulus Durasi** (*Duration Calculus*) atau perluasannya, *Extended Duration Calculus* dan *Mean Value Duration Calculus* untuk spesifikasi dan verifikasi sistem hibrida. He Jifeng [7] memperkenalkan perluasan dari **CSP** (*Communicating Sequential Processes*) sebagai suatu bahasa diskripsi formal untuk sistem hibrida. Pada tulisan ini diberikan suatu model *event-driven* dan memakai bahasa spesifikasi *Mean Value Duration Calculus* [24], yang merupakan perluasan dari *Duration Calculus* [23] untuk spesifikasi dan verifikasi sistem hibrida, dan diberikan suatu sistem metodologi perancang sistem hibrida berdasarkan kerja sama antar dua disiplin, yaitu disiplin *control engineering* dan disiplin *computer science*, dengan memberi suatu contoh klasik dalam teori kendali, yaitu bandul terbalik (*inverted pendulum*) [21].

## RANCANGAN KOOPERATIF UNTUK SISTEM HIBRIDA

Pada umumnya, sistem hibrida dapat dibagi atas tiga lapisan, seperti pada Gambar 1 [24]. Lapisan pertama adalah lapisan waktu kontinu (*continuous time layer*), yang terdiri dari *plant* yang dikendalikan, sensor-sensor untuk menangkap sinyal-sinyal yang dikeluarkan dari *plant* tersebut, dan *actuators* untuk mengendalikan sinyal-sinyal tersebut. Lapisan kedua adalah lapisan waktu diskret (*discrete time layer*), yang terdiri dari pengendali-pengendali digital untuk menerima sinyal-sinyal dari *plant* yang dikendalikan dan menyebarkan sinar kendali dan menghasilkan sinar kejadian. Lapisan pertama dan lapisan kedua ini berkomunikasi melalui pengubah (converter) A/D dan pengubah D/A. Pengubah A/D mengubah sinyal analog dari *plant* ke sinyal digital yang akan dikirim ke pengendali pada lapisan kedua. Pengubah D/A mengubah sinyal kendali digital dari si pengendali ke sinyal analog yang akan dikirim untuk mengendalikan *plant* tersebut.



Gambar 1. Arsitektur sistem hibrida

Lapisan ketiga adalah lapisan kejadian diskret (*discrete event layer*), yang terdiri dari pengambil-keputusan berdasarkan kejadian (*event-driven decision maker*). Pengambil-keputusan ini perlu apabila sistem hibridnya berinteraksi dengan lingkungan yang rumit. Pengambil-keputusan menerima sinyal kejadian simbolik dan mengirim sinyal perintah simbolik untuk menentukan pengendali-pengendali yang harus bereaksi agar dapat mengendalikan lingkungannya. Pengambil-keputusan berkomunikasi dengan lapisan kedua melalui pengubah D/S dengan pengubah S/D. Pengubah D/S menerima sinyal kejadian digital dari pengendali digital dan

mengubahnya menjadi sinyal kejadian simbolik, lalu diteruskan ke pengambil-keputusan. Pengubah S/D menerima sinyal perintah simbolik dari pengambil-keputusan menjadi sinyal perintah digital, lalu diteruskan ke si pengendali. Berdasarkan arsitektur yang dikemukakan diatas, sistem hibrida merupakan suatu sistem yang terbagi dalam ikalan-ikalan kendali (*control loops*) dan ikalan-ikalan keputusan (*decision loops*). Setiap ikalan pengendali mengandung satu pengendali untuk mengendalikan *plant* tersebut dalam suatu keadaan tertentu di sistem lingkungan itu. Rancangan suatu pengendali dapat dikerjakan dalam teori kendali dengan menggunakan analisis real atau simulasi atau kedua-duanya, dan sudah memberikan hasil-hasil yang gemilang, maka alangkah baiknya apabila rancangan pengendali dan validasinya diserahkan kepada pakarnya, *control engineers*. Terdapat dua jenis kelakuan dari setiap ikalan kendali, yang dapat diamati dari luar disebut kelakuan luar (*external behavior*) dan yang tidak dapat diamati disebut kelakuan dalam (*internal behavior*). Kelakuan luarnya sering kali dapat dispesifikasikan dalam bentuk yang dapat dimengerti oleh *control engineers* dan *computing scientists*, misalnya dalam bentuk aritmatika atau dalam bentuk analisis real yang sederhana, meskipun kelakuan dalamnya hanya dapat ditentukan oleh persamaan-persamaan diferensial biasa atau persamaan-persamaan diferensial parsial yang tidak mudah diselesaikan, termasuk hukum-hukum dinamikanya *plant* tersebut, begitu pula dengan hukum-hukum kendali dari pengendalinya. Dalam ikalan-ikalan keputusan, pengambil-keputusan setelah menerima sinyal-sinyal kejadian simbolik dari ikalan-ikalan kendali dan membuat komputasi, mengirim sinyal perintah simbolik untuk menentukan ikalan kendali berikutnya. Dengan demikian, pengambil-keputusan menentukan lintasan-lintasan (*trajectories*) dari ikalan-ikalan kendali yang ada berdasarkan kepada persyaratan (*requirement*) sistem yang diberikan dan lingkungannya. Bagian ini diserahkan kepada *computing scientists*. Persyaratan suatu sistem selalu merupakan sifat-sifat yang berkaitan dengan waktu-nyata, dari variabel-variabel keluaran (*output*) sistem tersebut. Sifat-sifat ini biasanya dapat dinyatakan dalam bentuk aritmatika atau dalam bentuk analisis real yang sederhana. Penentuan ikalan-ikalan keputusan, harus menjamin persyaratan sistem dapat dipenuhi dengan menggunakan sifat-sifat yang dipenuhi oleh kelakuan luar ikalan-ikalan kendali yang ada. dengan perkataan lain, persyaratan sistem harus dapat diturunkan dari lintasan-lintasan (yang ditentukan oleh pengambil-keputusan) dari spesifikasi luar (yang ditentukan oleh ikalan-ikalan kendali yang ada). Verifikasi ini dapat dilakukan secara formal yang tidak melibatkan teori persamaan diferensial atau sejenisnya. Hal ini diserahkan pula kepada *computing scientists*. Dengan demikian, rancangan keseluruhan sistem hibrida merupakan pembagian tugas antara *control engineers* dan *computing scientists*, maka disebut rancangan kooperatif untuk sistem hibrida (*Cooperative Design for Hybrid Systems*).

### Contoh: Bandul terbalik

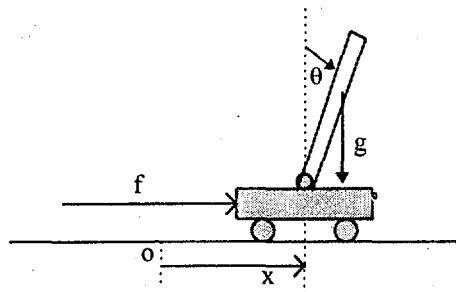
Untuk memperagakan usulan rancangan diatas, diambil sebuah contoh klasik yang sudah populer dibidang teori kendali, yaitu sebuah bandul terbalik. Sebuah bandul terbalik adalah sebuah batang yang salah satu ujungnya dilekatkan pada pusat sebuah kereta, seperti terlihat pada Gambar 2. Kereta itu dapat bergerak pada sebuah rel. Pergerakan kereta itu dikendalikan oleh sebuah kendali digital sedemikian agar akhirnya kereta itu dapat berhenti dengan batang tadi berdiri tegak pada tempatnya. Jika

$x(t)$  menyatakan jarak kereta dari sebuah titik pada rel tersebut pada waktu  $t$ ,  
 $x'(t)$  menyatakan kecepatan kereta pada waktu  $t$ ,  
 $x''(t)$  menyatakan percepatan kereta pada waktu  $t$ ,  
 $\theta(t)$  menyatakan sudut deviasi dari arah vertikal batang tersebut pada waktu  $t$ ,  
 $\theta'(t)$  menyatakan kecepatan sudut dari batang tersebut pada waktu  $t$ ,  
 $\theta''(t)$  menyatakan percepatan sudut dari batang tersebut pada waktu  $t$ ,

maka persyaratan sistem bandul terbalik ini dapat dinyatakan sebagai

$$(x'(t) \rightarrow 0) \wedge (\theta(t) \rightarrow 0) \wedge (\theta'(t) \rightarrow 0) \quad (1)$$

apabila  $t \rightarrow t_0$  untuk suatu  $t_0$  ( $t_0$  bisa  $\infty$ )



Gambar 2. Bandul terbalik (*Inverted pendulum*)

Untuk sistem ini, dalam lapisan waktu kontinu, berdasarkan hukum Newton, hukum dinamik dari bandul terbalik ini dapat digambarkan oleh persamaan-persamaan diferensial berikut:

$$\begin{aligned} (M+m)x'' + (m l \theta'' \cos \theta) - (m l \theta'^2 \sin \theta) &= f \\ (2 m l \theta'')/3 + m x'' \cos \theta &= m g \sin \theta \end{aligned}$$

di mana

$M$  adalah masa kereta,  $m$  adalah masa batang,  $\ell$  adalah panjang batang,  $g$  adalah gaya tarik bumi,  $f$  adalah gaya kendali.

Dalam lapisan waktu diskret, berdasarkan persyaratan sistem, *control engineers* menentukan dua buah hukum kendali:

- hukum D :  $f(t) = k_1 x'(t)$   $k_1$  konstan
- hukum R :  $f(t) = k_2 \theta(t) + k_3 \theta'(t)$   $k_2$  dan  $k_3$  konstan

Pada saat permulaan, dipakai hukum R. R akan memperbaiki  $\theta$  dan  $\theta'$ , sehingga derajat kebesarannya  $|\theta|$  dan  $|\theta'|$  lebih kecil dari derajat kebesarannya  $|x'|$ , yaitu terdapat  $\gamma$ , dengan  $0 < \gamma \ll 1$ , sehingga terpeuhi:

$$|\theta| + |\theta'| \leq \gamma |x'|$$

Setelah R dipakai hukum D, D akan memperlambat gerakan kereta dengan membuat sudut deviasi  $\theta$  membesar, yaitu terdapat  $\alpha > 0$  dan  $\beta > 0$  sehingga :

$$|\alpha\theta| + |\beta\theta'| \geq |x'|$$

D dan R masing-masing membentuk ikalan kendali. Akan dibuktikan bahwa dengan menggunakan kedua ikalan kendali tadi, bandul terbalik itu akan seimbang pada kedudukan vertikal dan keretanya berhenti bergerak.

Untuk menkoordinasi kedua ikalan kendali tadi, dibutuhkan pengambil-keputusan di lapisan kejadian diskret. Untuk itu didefinisikan dua buah kejadian (*event*),  $E_1$  dan  $E_2$ :

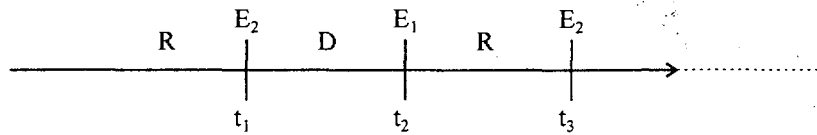
$$E_1 = |\alpha\theta| + |\beta\theta'| = |x'|$$
$$E_2 = |\theta| + |\theta'| = \gamma |x'|$$

Apabila kejadian  $E_1$  terjadi, pengambil-keputusan memberi sinyal pada ikalan kendali R untuk beraksi, apabila kejadian  $E_2$  terjadi, pengambil-keputusan memberi sinyal pada ikalan kendali D untuk beraksi. Berarti pengambil-keputusan menentukan lintasan-lintasan dari ikalan-ikalan kendali tersebut seperti berikut:

- Pada permulaan, hukum kendali R beraksi
- Beraksinya hukum R untuk jangka waktu tertentu, kejadian  $E_2$  akan muncul
- Apabila  $E_2$  muncul, berikan perintah agar hukum D beraksi
- Beraksinya hukum D untuk jangka waktu tertentu, kejadian  $E_1$  akan muncul
- Apabila  $E_1$  muncul, berikan perintah agar hukum R beraksi lagi

dan seterusnya. Di sini computing scientists tidak perlu dipusingkan oleh persamaan-persamaan diferensial diatas, begitu pula tidak dibutuhkan teori persamaan diferensial

untuk membuktikan terpenuhinya persyaratan sistem. Begitu pula, untuk membuktikan terpenuhinya persyaratan sistem secara formal bukan suatu pekerjaan yang mudah untuk *control engineers* jika lintasan-lintasan ikalan kendali menjadi rumit. Sedangkan kelakuan luar dari ikalan-ikalan kendalai itu dapat diamati dan dapat dinyatakan dengan analisis real yang sederhana dalam variabel-variabel  $x$ ,  $x'$ ,  $\theta$  dan  $\theta'$ .



Gambar 3. Lintasan ikalan-ikalan kendali

Perhatikan interval waktu  $[t_1, t_3]$  seperti pada Gambar 3,  $t_1$  menunjukkan waktu pada saat kejadian  $E_2$  muncul,  $t_3$  menunjukkan waktu pada saat kemunculan kejadian  $E_2$  berikutnya, dan  $(t_1 < t_2 < t_3)$  menunjukkan waktu pada saat kejadian  $E_1$  muncul diantara kemunculan dua buah kejadian  $E_2$  yang berurutan. Sifat-sifat yang terpenuhi pada potongan lintasan pada interval waktu  $[t_1, t_3]$  adalah

Terdapat  $d_1$  dengan  $0 \leq d_1 < 1$  sehingga untuk setiap interval  $[t_1, t_3]$ , berlaku

$$|x'(t_3)| \leq d_1 |x'(t_1)| \quad (2)$$

Terdapat  $d_2 > 0$  sehingga untuk setiap interval  $[t_1, t_3]$  dan  $t \in [t_1, t_3]$  berlaku

$$|x'(t)| \leq d_2 |x'(t_1)| \quad (3)$$

Terdapat  $d_3 > 0$ ,  $\alpha > 0$  dan  $\beta > 0$  sehingga untuk setiap interval  $[t_1, t_3]$  dan  $t \in [t_1, t_3]$  berlaku

$$|\alpha\theta(t)| + |\beta\theta'(t)| \leq d_3 |x'(t_2)| \quad (4)$$

(2) dan (3) menunjukkan mengecilnya kecepatan kereta, sedangkan (4) menunjukkan menurunnya nilai mutlak dari sudut deviasi  $\theta(t)$  dan kecepatan sudut  $\theta'(t)$ . Menunjukkan persyaratan sistem (1) dari pertaksamaan (2), (3) dan (4) dengan mengingat lintasan yang telah ditentukan dapat dilakukan dengan analisis real yang sederhana, tidak perlu melibatkan teori persamaan diferensial, hal ini dapat dilakukan oleh *computing scientists*. Sedangkan verifikasi terpenuhinya ketiga pertaksamaan oleh hukum-hukum kendali R dan D merupakan tugas *control engineers*.



## SPESIFIKASI DAN VERIFIKASI FORMAL

Di [21], spesifikasi dan verifikasi formal untuk sistem bandul terbalik itu dipakai *mean Value Duration Calculus*, yang merupakan perluasan dari *Duration Calculus*. Dalam *Mean Value Duration Calculus*, spesifikasi formalnya dinyatakan sebagai berikut:

$$\Box [D \vee R]^* \quad (5)$$

$$\Box [(D \Rightarrow \neg R) \wedge (R \Rightarrow \neg D)]^* \quad (6)$$

$$\Box (E_1 \Leftrightarrow \downarrow D \wedge \uparrow R) \quad (7)$$

$$\Box (E_2 \Leftrightarrow \downarrow D \wedge \uparrow R) \quad (7)$$

dimana

$$[P]^* = [P] \vee [ ]$$

(5), (6), (7) dan (8) mengambar spesifikasi yang diberikan oleh pengambil-keputusan. (5) menunjukkan kelengkapan dan (6) menunjukkan keekklusifan dari kedua ikalan kendali. (7) dan (8) menunjukkan perannya kedua kejadian  $E_1$  dan  $E_2$ .

Sifat-sifat (2), (3) dan (4) dari ikalan-ikalan kendali dapat diformulasikan sebagai berikut: Terdapat  $0 \leq d_1 < 1$ ,  $d_2 > 0$ ,  $d_3 > 0$ ,  $\alpha > 0$  dan  $\beta > 0$  sehingga

$$\Box ((E_2 \wedge |x'| \leq r); AE \Rightarrow EA; (E_2 \wedge |x'| \leq d_1 r)) \quad (9)$$

$$\Box ((E_2 \wedge |x'| \leq r); AE \Rightarrow E_2; (AE_2 \wedge [|x'| \leq d_2 r])) \quad (10)$$

$$\Box ([D]; (E_1 \wedge |x'| = u); [R]; E_2 \Rightarrow AE_2 \wedge [|\alpha\theta| + |\beta\theta'| \leq d_3 u]) \quad (11)$$

$$\text{dimana} \quad A = [D]; E_1; [R] \quad (12)$$

$$AE = A; E_1 \quad (13)$$

$$EA = E_1; A \quad (14)$$

untuk menyingkat, dipakai pula notasi berikut:

$$X^n = X; X; \dots; X \quad (n \text{ buah } X) \quad (15)$$

(9), (10) dan (11) merupakan spesifikasi untuk sistem bandul terbalik itu.

Misalkan  $\text{Traj}(n) = [D]; E_2; (AE)^n$  menyatakan bagian depan dari lintasan ikalan-ikalan kendali, maka persyaratan sistem (1) dapat dinyatakan sebagai berikut:

Untuk sembarang  $\varepsilon > 0$ , terdapat bilangan bulat  $N$  sehingga untuk setiap  $n > N$ , berlaku

$$\text{Req: Traj}(n) \Rightarrow (\text{Traj}(N); [(|x'| < \varepsilon) \wedge (|\theta| < \varepsilon) \wedge (|\theta'| < \varepsilon)]) \quad (16)$$

Untuk membuktikan Req, dibutuhkan lemma-lemma berikut:

**Lemma 1** Untuk  $n \geq 1$ , berlaku

$$\square((E_2 \wedge |x'| \leq r); (AE)^n \Rightarrow (EA)^n; (E_2 \wedge |x'| \leq d_1^n r)) \quad (17)$$

Dengan Menggunakan Lemma 1, dapat dibuktikan lemma berikut:

**Lemma 2** Untuk  $n \geq 1$ , berlaku

$$\square((E_2 \wedge |x'| \leq r); (AE)^n \Rightarrow (EA)^{n-1}; E_2; (AE \wedge \lceil |x'| \leq d_2 d_1^{n-1} r \rceil)) \quad (18)$$

**Lemma 3** Untuk  $n \geq 1$ , berlaku

$$\square((E_2 \wedge |x'| \leq r); (AE)^n \Rightarrow (EA)^{n-1}; E_2; (AE \wedge \lceil |\alpha\theta| + |\beta\theta'| \leq d_3 d_2 d_1^{n-1} r \rceil)) \quad (19)$$

Dengan menggunakan Lemma 2 dan Lemma 3, dapat dibuktikan Req.(16)

### **Teorema**

Untuk sembarang  $\varepsilon > 0$ , terdapat bilangan blat  $N$  sehingga untuk setiap  $n > N$ , berlaku

$$\text{Req: Traj}(n) \Rightarrow (\text{Traj}(N); \lceil (|x'| < \varepsilon) \wedge (|\theta| < \varepsilon) \wedge (|\theta'| < \varepsilon) \rceil)$$

## **PENUTUP**

Dalam tulisan ini diambil rancangan sistem hibrida berdasarkan kooperatif antara dua disiplin, *control engineering* dan *computing science*. Prinsip ini tidak dapat dilakukan dengan baik apabila sistem hibrida merupakan sistem yang *tightly coupled*, yaitu apabila ikalan-ikalan kendalinya tidak dapat dipisahkan dengan keputusan. Hal ini membuka peluang untuk diteliti. Di samping itu, seperti telah disinggung pada pendahuluan, bagaimana sistem hibrida analisis disintesisikan, bagaimana memverifikasi suatu sistem hibrida memenuhi persyaratan yang diinginkan menjadi tantangan bagi peneliti.

## DAFTAR PUSTAKA

1. RAJEEV ALUR, COSTAS COURCOUBETIS, THOMAS A.HENZINGER and PEI-HSIN HO, *Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems*, In *Hybrid Systems*, LNCS 736, Springer-Verlag, pp 209-229, 1993.
2. R.ALUR, C.COURCOUBETIS, N.HALBWACHS, T.A.HENZINGER, P.-H. HO, X.NICOLLIN, A.OLIVERO, J.SIFAKIS and S.YOVINE, *The Algorithmic Analysis of Hybrid Systems*, In proceedings of the 11 th International Conference on Analysis and Optimization of Discrete Event Systems, Lecture Notes in Control and Information Sciences 199, Springer-Verlag, pp 331-351. 1994.
3. PANOS J.ANTSKLIS, JAMES A.STIVER and MICHAEL LEMMON, *Hybrid Systems Modeling and Autonomous Control Systems* LNCS 736, Springer-Verlag, pp 366-392, 1993.
4. WILLIAM J.BENCZE and GENE F.FRANKLIN, *A Separation Principle for Hybrid Control Systems Design*, In proceedings of the 1994 IEEE/IFAC Symposium on Computer-Aided Control System Design, March 7-9, 1994 in Tuscon, Arizona.
5. LIMOR FIX and FRED B.SCHNEIDER, *Hybrid Verification by Exploiting the Environment* In *Formal Techniques in Real-Time and Fault-Tolerant Systems* LNCS 863, Springer-Verlag, pp I - 18, 1994.
6. THOMAS A.HENZINGER, ZOHAR MANNA and AMIR PNUELI, *Toward Refining Temporal Specifications into Hybrid Systems* In *Hybrid Systems* LNCS 736, Springer-Verlag, pp 60-76, 1993.
7. JIFENG HE, *From CSP to Hybrid Systems* In *A Classical Mind Essays in Honour of C.A.R. Hoare* Prentice-Hall, pp 171 - 190, 1994.
8. JOZEF HOOMAN, *A Compositional Approach to the Design of Hybrid Systems* In *Hybrid Systems* LNCS 736, Springer-Verlag, pp 121 - 148, 1993.
9. YU HUIQUN, PARITOSH K.PANDYA and SUN YONGQIANG, *A Calculus for Hybrid Sampled Data Systems* In *Formal Techniques in Real-Time and Fault Tolerant Systems* LNCS 863, Springer-Verlag, pp 716-737, 1994.
10. ARJUN KAPUR, THOMAS A.HENZINGER, ZOHAR MANNA and AMIN PNUELI. *Proving Safety of Hybrid Systems* In *Formal Techniques in Real-Time and Fault-Tolerant Systems* LNCS 863, Springer-Verlag, pp 431 -454, 1994.

11. Y.KESTEN and A.PNUELI, *Timed and Hybrid Statechard and their Textual Representation*, In *Formal Techniques in Real-Time and Fault Tolerant Systems* LNCS 571, Springer-Verlag, pp 591-620, 1992.
12. Y.KESTEN, A.PNUELI, J.SAFAKIS and S.YOVINE, *Integration Graphs: A Class of Decidable Hybrid Sytems* In *Hybrid Systems* LNCS 736, Springer-Verlag, pp 179-208, 1993.
13. LESLIE LAMPORT, *Hybrid Systems in TLA<sup>+</sup>* In *Hybrid Systems* LNCS 736, Springer-Verlag, pp 77- 102, 1993.
14. M.LEMMON, J.A.STIVER and P.J.ANTSAKLIS, *Event Identification and Intelligent Hybrid Control* In *Hybrid Systems* LNCS 736, Springer-Verlag, pp 268-296, 1993. .
15. ODED MALER, ZOHAN MANNA and AMIR PNUELI, *From Timed to Hybrid Systems* In proceedings of the REX Workshop in *Real-Time: Theory in Practice* LNCS 600, Springer-Verlag, pp 447-484, 1992.
16. ZOHAR MANNA and AMIR PNUELI, *Models for Reactivity* Acta Informatica 30, pp 609-678, 1993
17. ZOHAR MANNA and AMIR PNUELI, *Verifying Hybrid Systems* In *Hybrid systems*, LNCS 736, Springer-Verlag, pp 4-35, 1993
18. ANIL NERODE and WOLF KOHN, *Models for Hybrid Systems : Automata Topologies Controllability Observability* In *Hybrid Systems* LNCS 736 Springer-Verlag, pp 317-356, 1993.
19. HENNY B.SIPMA and ZOHAR MANNA, *Specification and Verification of Controlled Systems*, In *Formal Techniques in Real-Time and Fault Tolerant Systems* LNCS 863, Springer-Verlag, pp 641-659, 1994.
20. SIMIN NADJM-TEHRANI and JAN-ERIK STROMBERG, *Fronl Physical Modelling to Compositional Models of Hybrid Systems* pp 583-604.
21. BELAWATI H.WIDJAJA, CHEN ZONGJI, HE WEIDONG and ZHOU CHAOCHEN, *A Cooperative Design for Hybrid Systems* UNU/IIST Report No.36, January 1995, UNU/IIST, P.O.Box 3058, Macao, presented in the European Workshop on Real-Time and Hybrid Systems, May 51 - June 2, 1995 in Grenoble, France.
22. YU XINYAO, WANG JI, ZHOU CHAOCHEN and PARITOSH K. PANDYA, *Formal Design of Hybrid Systems* In *Formal Techniques in Real-Time and fault-Tolerant Systems* LNCS 863, Springer-Verlag, pp 738-755, 1994.

23. ZHOU CHAOCHEN, C.A.R.HOARE and ANDER P.RAVN, *A Calculus ad Durations*, Information Processing Letters, 40, 5, pp 269-276, 1991.
24. ZHOU CHAOCHEN and LI XIAOSHAN, *A Mean Value Calculus of Durations* UNU/IIST Report No.5, March 1993, published in *A Classical Mind Essays in Honour of C.A.R.Hoare*, Prentice Hall, pp 431 -451, 1994.
25. ZHOU CHAOCHEN, ANDER P.RAWN and MICHAEL R.HANSEN, *An Extended Duration Calculus for Hybrid Real-Time Systems In Hybrid Systems* LNCS 736, Springer-Verlag, pp 36-59, 1993.

**NEXT PAGE(S)  
left BLANK**