

Yeniden Yapılandırılabilir Üretim Sistemleri için Kontrolcu Tasarımı

Ayşe Nur Sülek^{1,*}, Kıvanç Uğur Afşar¹ ve Klaus Werner Schmidt²

¹Çankaya Üniversitesi, Elektronik ve Haberleşme Mühendisliği, 06810 Yenimahalle, Ankara, Türkiye

²Çankaya Üniversitesi, Mekatronik Mühendisliği, 06810 Yenimahalle, Ankara, Türkiye

*Corresponding author: anursulek@cankaya.edu.tr

Özet. Değişen ürün pazarlarına ve türlerine uyum sağlamak için, hızlı yeniden yapılandırılabilir üretim sistemleri (YÜS) günümüz üretim teknolojisine önemli bir çözüm sunmaktadır. Bu makalede, yeniden yapılanmanın hızlı olması ve kontrolcülerin kolayca gerçekleştirilmesi için yeniden yapılandırılabilir üretim sistemlerinin (YÜS) kontrolü ele alınmaktadır. Makale kapsamında, bir konfigürasyon sistemin istenen bir olası operasyonu anlamına gelmektedir.

Kullandığımız yöntem YÜS'lerin ayrık olaylı sistem (AOS) modeli üzerine kurulmuştur ve sistemin her bir konfigürasyonu için kontrolcü tasarımı ile başlar. Daha sonra bu kontrolcü temel operasyonuna göre dört farklı biçimde genişletilir. Her kontrolcü dört biçimden birini uygulamaktadır: kendi konfigürasyonunu aktif olarak çalıştırır (biçim 1), kendi konfigürasyonunu bitirir ve aktif olmayan duruma geçer (biçim 2), aktif olmayan durumda bekler (biçim 3) ya da yeniden aktif olur (biçim 4). Oluşturduğumuz algoritma yalnızca bir konfigürasyonun aktif olmasını, diğer bütün konfigürasyonların ise aktif olmamasını sağlar. Ayrıca konfigürasyonlar arasında sınırlı bir sürede geçiş yapabilmeyi sağlar. Yaklaşımımız ile her bir konfigürasyon için kontrolcü hazırlandıktan sonra, çalışan sisteme kolaylıkla yeni bir konfigürasyon eklenebilir. Geliştirilen yöntem küçük bir üretim sistemi örneğiyle açıklanmıştır.

Anahtar Kelimeler. Ayrık olaylı sistemler, üretim sistemleri, gözetimli kontrol, yeniden yapılandırma.

Abstract. The rapid reconfiguration of manufacturing systems is an important issue in today's manufacturing technology in order to adjust the production to varying product demands and types. In this paper, we study the control of reconfigurable manufacturing systems (RMSs) with the aim of fast reconfiguration and an easy controller implementation. In the scope of this paper, a configuration denotes one possible desired operation of the system.

Received September 7, 2012; accepted February 25, 2013.

Bu makale, 26-27 Nisan 2012 tarihlerinde Çankaya Üniversitesi'nin Ankara Merkez yerleşkesinde yapılmış olan 5. Çankaya Üniversitesi Mühendislik ve Teknoloji Sempozyumu'nda sunulan ve sadece geniş bildiri özeti bölümü hakem sürecinden geçerek bu sempozyum kitapçığında yayımlanan bir makalenin revize edilmiş şekli olup Sempozyum Değerlendirme Komitesi tarafından yayımlanmak üzere Çankaya University Journal of Science and Engineering dergisine gönderilmesi önerilmiş ve derginin bağımsız hakem değerlendirmeleri sonucunda yayıma kabul edilmiştir.

Our method is based on a discrete event system (DES) model of an RMS and starts with a controller design for each separate configuration of the system. This controller is then extended such that its basic operation assumes four different modes. Each controller is either actively performing its configuration (mode 1), completing its configuration and becoming inactive (mode 2), staying inactive (mode 3) or becoming active again (mode 4). Our algorithmic design guarantees that if a certain configuration is active, then all other configurations are inactive. Furthermore, it allows to switch between configurations in a bounded time. Since one controller is designed for each configuration, the approach makes it easy to add new configurations to a running system. The developed method is illustrated by a small manufacturing system example.

Keywords. Discrete event systems, manufacturing systems, supervisory control, reconfiguration. algorithm.

1. Giriş

Girişken rekabet, ürün geliştirmedeki hızlı değişimler ve çağdaş üretim sistemlerindeki imalat teknolojisi *yeniden yapılandırılabilir üretim sistemlerini* (YÜS) öne çıkarmaktadırlar [1, 2, 3, 4]. Yeniden yapılandırılabilir üretim sistemleri; yeni imalat modellerinin piyasaya sürülmesini, imalat kapasitesinin düzenlenmesini, yeni özelliklerin ve teknolojilerin entegrasyonunu ve çeşitli imalat miktarlarının adaptasyonunu hızlı ve kolay bir şekilde gerçekleştirmektedirler.

Bu makalenin konusu, kontrol stratejilerin tasarımı ve bu tasarımın sorunsuz bir şekilde değişik *konfigürasyonlar* arasında kullanılabilmesidir. Yaklaşımımız Ramadge ve Wonham'ın *supervisory control theory for discrete event systems* adlı çalışmasına dayanmaktadır [5]. Öncelikle her bir yeniden yapılandırılabilir üretim sisteminin sıralı faaliyetini sonlu durum modeli ile göstermekteyiz. Bunu takiben, her konfigürasyon için dört değişik biçimde *yeniden yapılandırma kontrolcüsü* oluşturmayı sağlayan bir algoritma geliştirmekteyiz. Bu yeniden yapılandırma kontrolcüsü dört farklı biçimde çalışabilir: biçim1: aktif olarak konfigürasyonun çalışması, biçim2: konfigürasyonunu tamamlayıp çalışmayan bekleme durumuna geçmesi, biçim3: çalışmadığı bekleme durumunda kalması ve biçim4: konfigürasyonun yeniden çalışmaya başlaması. Yeniden yapılandırma kontrolcüleri eşzamanlı bir şekilde çalışmak için tasarlanmıştır: bir yeniden yapılandırma kontrolcüsü (geçerli olan konfigürasyonun kontrolcüsü) aktif durumda çalışırken, diğer bütün yeniden yapılandırma kontrolcüleri (kendi konfigürasyonları aktif olmak için seçilene kadar) aktif değildir. Ayrıca, kontrolcüler sınırlı bir zaman için konfigürasyonların birbirleri arasında değişimlerine izin verir. Her bir konfigürasyon için ayrı bir yeniden yapılandırma kontrolcüsü kullandığımız için YÜS'e yeni bir konfigürasyon eklemek kolaylaşmıştır. Tek gerekli

olan, yeni konfigürasyon için bir yeniden yapılandırma kontrolcüsü oluşturmak ve bunu zaten var olan yeniden yapılandırma kontrolcüleri ile eşzamanlı bir şekilde çalıştırmaktır. Geliştirilmiş olan yöntem için bir üretim sistem örneği sunulmaktadır.

Bu konuyla ilgili literatürdeki araştırma sınırlıdır. [6]'da ayrık olaylı sistemler için konfigürasyonların ayrı ayrı gereklilikleri ele alınmıştır. Fakat konfigürasyon değişikliklerinin bir geçiş zamanı olmadan yapıldığı varsayılmıştır. Bizim yöntemimizde bu tür kısıtlamalar yoktur. [7, 8]'de YÜS'ler için kontrolcülerin *etkin uygulamaları* çalışılmıştır fakat kontrolcü tasarımı sunulmamaktadır. [8]'deki yaklaşım, yeniden yapılandırmadaki Petri net kontrolcüleri için YÜS'lerin yapı bloklarını kullanmıştır. [9]'deki çalışma, hibrid dinamik sistemler için formüle edilmiştir ancak YÜS'ler için de kullanılabilir. Önceki çalışmalardan sadece [10] yeniden yapılandırılabilir makine araçlarının (Reconfigurable Machine Tool - RMT) farklı konfigürasyonları için sistematik kontrolcü tasarımı için bir yöntem önermektedir. Makalede sistemin yeniden konfigürasyonu için yapılması gerekenlerin algoritmik çözümü için kurallar açıklanmıştır. Yeniden yapılandırma kontrolcüsü hesabı için bir yapılandırma algoritması belirtmemiştir ve her bir konfigürasyon için yeni bir yeniden yapılandırma kontrolcüsünün tasarımı içermemektedir.

Makalemizin organizasyonu şu şekildedir: 2. bölüm'de ayrık olaylı sistemlerin gözetimli kontrol için temel gösterimleri açıklanmıştır. Üzerine çalışılan problem motive edici bir örnekle tanımlanmıştır ve 3. bölümde problem formülize edilmiştir. 4. bölümde yeniden yapılandırma kontrolcüsünün yapılandırma algoritması açıklanmıştır ve son olarak 5. bölüm elde edilen sonuçları göstermektedir.

2. Temeller

Bu bölümde gözetimli kontrol teorisi (supervisory control theory) için temel notasyonlar açıklanmaktadır [11, 12].

2.1. Ayrık olaylı sistemler (Discrete event systems – DES). Sonlu bir olay alfabesi Σ için tüm sonlu dizilerin (string) kümesi Σ^* olsun. $s_1, s_2 \in \Sigma^*$ dizilerinin ardarda eklenmesi (concatenation) $s_1s_2 \in \Sigma^*$ ve s_1 'in s 'in bir *önek* (*prefix*) olması durumu $s_1 \leq s$ ile gösterilir. Boş dizi (empty string) $\epsilon \in \Sigma^*$, için $s\epsilon = \epsilon s = s$ doğrudur. Σ üzerinde bir *dil* (*language*) bir altküme $L \subseteq \Sigma^*$ olarak tanımlanır. L 'nin *önek kapalılığı* (*prefix closure*) $\bar{L} := \{s_1 \in \Sigma^* | \exists s \in L \text{ s.t. } s_1 \leq s\}$ olarak tanımlanır. Buna göre bir dil L , $L = \bar{L}$ sağladığında *önek kapalı* (*prefix closed*) olur.

Bir DES sonlu otomat $G = (X, \Sigma, \delta, x_0, X_m)$ olarak modellenir. Bu modelde X sonlu *durum* kümesini; Σ sonlu *olay* alfabetini; $\delta : X \times \Sigma \rightarrow X$ kısmi (partial) *durum değiştirme fonksiyonunu* (partial function); $x_0 \in X$ *ilk durumu* ve $X_m \subseteq X$ *işaretli durumlar* (marked states) kümesini gösterir. $\delta, (x, \sigma)$ 'da tanımlı ise $\delta(x, \sigma)!$ yazılabilir. Durum değiştirme fonksiyonu δ kısmi bir fonksiyon olan $X \times \Sigma^*$ 'a standart şekilde genişletilebilir. $|X|$ durum sayısını gösterir.

G 'nin davranışı kendisinin *kapalı dili* (closed language) olan $L(G) := \{s \in \Sigma^* | \delta(x_0, s)!\}$ tarafından tanımlanır. G için *işaretli dil* (marked language) $L_m(G) := \{s \in L(G) | \delta(x_0, s) \in X_m\}$ olarak tanımlanır. $\overline{L_m(G)} = L(G)$ ise G *engelsiz* (nonblocking) olma özelliğine sahiptir. İki otomat G_1 ve G_2 için *eşderleme* (synchronous composition) $G_1 || G_2$ bu otomatların eşzamanlı çalışmalarını tanımlar. Eşçarpım operasyonun formal tanımı için [11]'e bakınız.

$G = (X, \Sigma, \delta, x_0, X_m)$ ve $G' = (X', \Sigma, \delta', x'_0, X'_m)$ için sonlu durum otomatları olsun. G', G' 'nin *altotomatu* (subautomaton) olsun, $G' \sqsubseteq G$ ile gösterilsin. Bu durumda ya G' boş otomattır ($X' = \emptyset$) ya da $x'_0 = x_0, X' \subseteq X$ ve bütün $x \in X'$ ve $\sigma \in \Sigma$ için $\delta'(x, \sigma)! \Rightarrow \delta'(x, \sigma) = \delta(x, \sigma)$ doğrudur [13].

2.2. Gözetimli kontrol (Supervisory control). Gözetimli kontrol kapsamında *kontrol edilebilir* (controllable) ve *kontrol edilemez* (uncontrollable) olaylar sırasıyla (Σ_c) ve (Σ_u) olarak gösterilebilir ve $\Sigma = \Sigma_c \dot{\cup} \Sigma_u$ olarak tanımlanır. Alfabe Σ üzerinde bir otomat $S = (Q, \Sigma, \nu, q_0, Q_m)$ olsun. S, Σ_c 'deki olayları etkisizleştirebiliyorsa (disable) bir sistem G için bir *kontrolcu* olabilir. Bu durumda her $s \in L(G) \cap L(S)$ ve $s\sigma \in L(G)$ olan $\sigma \in \Sigma_u$ için $s\sigma \in L(S)$ sağlanmalıdır. Bu takdirde $G || S$ kapalı devre sistemini gösterir.

Bir $K \subseteq L_m(G)$ dili $L(G) \subseteq \Sigma^*$ ve Σ_u için kontrol edilebilir olarak tanımlanabilmesi için $\overline{K}\Sigma_u \cap L(G) \subseteq \overline{K}$ olması gerekir. $L_m(S) = K$ şartını sağlayan bir kontrolcu S olabilmesi için gerek ve yeter şart K 'nın $L(G)$ ve Σ_u için kontrol edilebilir olmasıdır [5]. K 'nın $L(G)$ ve Σ_u için kontrol edilebilir olmadığı durumda kontrolcu K için *en yüksek mertebede kontrol edilebilen alt dili* (supremal controllable sublanguage) gerçekleştirecektir. $L_m(S || G) = \sup C(K, L(G), \Sigma_u)$ olarak belirtilir. $\sup C(K, L(G), \Sigma_u) \neq \emptyset$ ise kontrolcunun engelsiz (nonblocking) ve en yüksek mertebeden izin veren (maximally permissive) olması garantilenir [5]. Bir kontrolcu S, G 'nin alt otomatu ise ($S \sqsubseteq G$), $S G$ için bir *durum geri beslemeli kontrolcudur* (state-feedback supervisor).

2.3. Çekim (Attraction). $G = (X, \Sigma, \delta, x_0, X_m)$ bir otomat olsun. $X' \subseteq X$ 'in G içinde bir *değişmez küme* (*invariant set*) olarak tanımlanması için X' içinde bir durumdan hiç bir değişmenin bu kümeden çıkmaması gerekir. Buna göre $\forall x \in X'$ ve $\sigma \in \Sigma$ için $\delta(x, \sigma) \in X'$ olmalıdır.

[14]'de bulunan *güçlü çekim* (*strong attraction*) tanımına göre. $A \subseteq X' \subseteq X$ ve A, X' G 'de değişmez kümeler olsun. Bu durumda A 'nın, G 'deki X' için bir *güçlü çekici* (*strong attractor*) olması için şartlar:

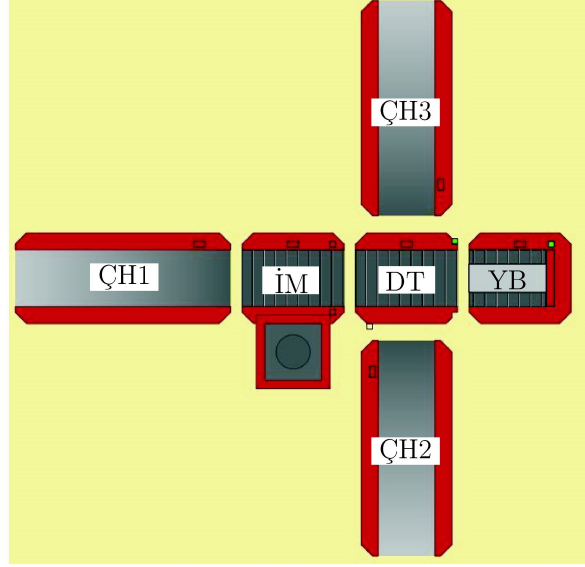
- G 'nin altotomatının durum kümesinin $X' \setminus A$ olması ve döngüsüz (acyclic) olması
- $\forall x \in X', \delta(x, u) \in A$ Sağlayacak bir $u \in \Sigma^*$ olması

$\Sigma_u \subseteq \Sigma$ bir kontrol edilemez olaylar kümesi olsun. Eğer bir Σ_u 'ya göre durum geribeslemeli kontrolcü $S \subseteq G$ varsa A *zayıf çekici* (*weak attractor*) G de X' için olarak tanımlanır. G 'de X' için A zayıf çekici olacak şekilde durum geribeslemeli kontrolcü S hesaplaması [15]'de incelenmektedir. G 'de X' için A güçlü çekici olacak şekilde *en az kısıtlamalı optimal kontrolcü* (*minimally restrictive optimal supervisor*) S varlığı gösterilmektedir. S 'nin optimal olması daha küçük yakınsama zamanlı başka bir S' kontrolcusunun olmamasını, en az kısıtlamalı olması ise aynı yakınsama zamanına sahip bir S' kontrolcüsü için $S' \subseteq S$ olması demektir.

3. Problem Tanımı

3.1. Motive edici örnek. Şekil 1'de gösterilen sistem, yığın besleyicisi (YB), dönen taşıyıcı (DT), işlemci makine (İM) ve üç tane çıkış hattından (ÇH) oluşmaktadır. Ürünler sisteme yığın besleyicisinden giriş yapar ve yığın besleyicisi tarafından dönen taşıyıcıya gönderilir. Dönen taşıyıcı saat yönünde ve saat tersi yönünde dönerek ürünü ikinci ve üçüncü çıkış hatlarına ya da makineye yönlendirir. Aynı şekilde makineden gelen ürünü alıp ikinci ve üçüncü çıkış hatlarına gönderir. Makineye gelen ürün birinci çıkış hattına ya da dönen taşıyıcıya gönderilir.

Sistemde bulunan her bir eleman için Şekil 2'de gösterildiği gibi sonlu durum modeli oluşturulur. Bu model her elemanın gerçekleştirebildiği olayları ve bu olaylara göre durumunu belirtir. İlk olarak yığın besleyicisi modeli, yığın besleyicisinin işlevlerini göstermektedir. Yığın besleyicisi sistemde tek yönlü çalışmaktadır. Bu tek yönlü işlev, yığın besleyicisinden dönen taşıyıcıya gönderilme olayıyla (yb-dt) modellenmiştir. Yığın besleyicisi gibi tek yönlü çalışan çıkış hatları da aynı şekilde tek olayla modellenmiştir. Birinci çıkış hattı için, makineden birinci çıkış hattına

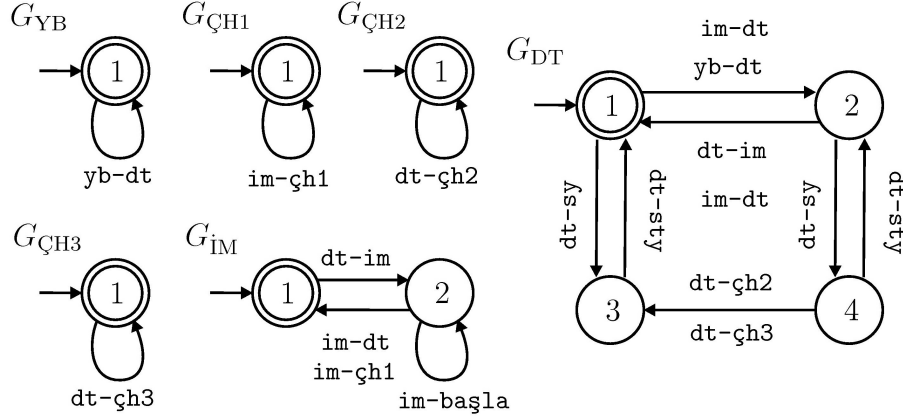


ŞEKİL 1. Üretim sisteminin genel görünümü.

(*im-çh1*); ikinci çıkış hattı için, dönen taşıyıcıdan ikinci çıkış hattına (*dt-çh2*) ve üçüncü çıkış hattı için, dönen taşıyıcıdan üçüncü çıkış hattına (*dt-çh3*) olayları ile modellenmiştir. İşlemci makine ise çift yönlü çalışmaktadır ve bu sebeple; makineden birinci çıkış hattına (*im-çh1*), makineden dönen taşıyıcısına (*im-dt*), dönen taşıyıcısından makineye (*dt-im*) gitme olaylarını içererek modellenmiştir. Aynı zamanda makine işlem yaptığı için, ‘makine çalışma’ (*im-başla*) olayıyla işlem durumu modelde gösterilmiştir. Son olarak dönen taşıyıcı modeli; makineden dönen taşıyıcıya (*im-dt*) ya da yığın besleyicisinden dönen taşıyıcısına (*yb-dt*) olaylarıyla başlar. Bu durumda saat yönünde dönme olayı (*dt-sy*) ya da dönen taşıyıcısından makineye gitme (*dt-im*) olayı gerçekleşebilir. Saat yönünde dönmüş olan taşıyıcı dönen taşıyıcısından çıkış hatlarına olan olayları (*dt-çh2* ve *dt-çh3*) gerçekleştirir ve tekrar saatin tersi yönünde dönme olayıyla (*dt-sty*) eski haline gelir. Bütün elemanların modelleri paralel birleştirme yöntemiyle bir araya getirilerek bütün sistemin genel modeli G elde edilir:

$$G = G_{YB} || G_{DT} || G_{İM} || G_{ÇH1} || G_{ÇH2} || G_{ÇH3}$$

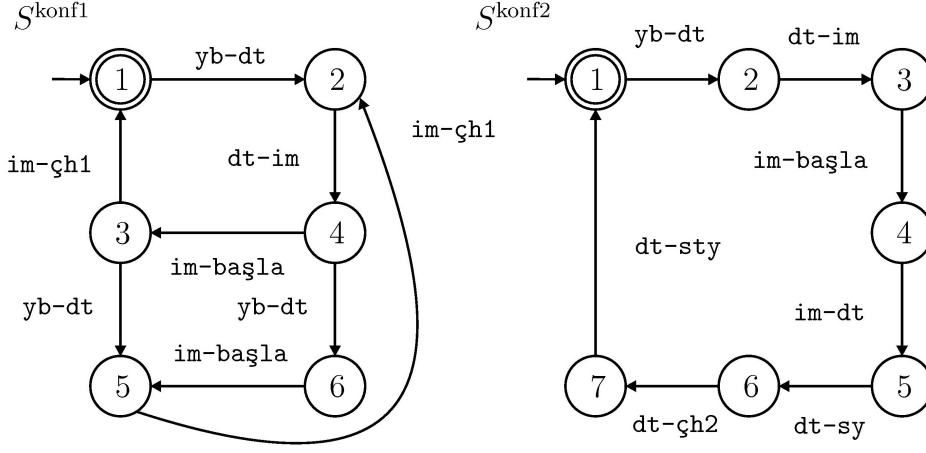
Sistemde kontrol edilebilir ve kontrol edilemez olmak üzere iki çeşit olay vardır. Bu sistemdeki kontrol edilemeyen olay *dt-çh2* ve *dt-çh3*'tür.



ŞEKİL 2. Üretim sisteminin elemanlarının modelleri.

Sistemin istediğimiz şekilde çalışması kontrolcüler tarafından sağlanır. Kontrolcülerini oluşturmak için önce sistemin spesifikasyonları belirlenir ve bu spesifikasyonlar sistemin nasıl çalışması gerektiğine göre modellenir. Sistemin genel modeli ve spesifikasyonları bir araya getirilerek kontrolcüler elde edilir. Kontrolcüler sistemi kontrol ederek, sistemin çalışması istenilen şekilde çalışmasını sağlar. Şekil 3’de konfigürasyon 1 (S^{konf1}) ve konfigürasyon 2 (S^{konf2}) için iki kontrolcü gösterilmiştir. Konfigürasyon 1 kontrolcüsü sistemi şu şekilde işletir: Ürün sisteme yığın besleyicisinden giriş yapar. Gelen ürünü dönen taşıyıcıya yönlendirir. Ürün buradan makineye gider ve işlemi bitince birinci çıkış hattına (ÇH1) gider. Konfigürasyon 2’de ürün tekrar yığın besleyicisinden sisteme girer ve dönen taşıyıcıya gönderilir. Dönen taşıyıcı makineye ürünü gönderir ve makede işlemi tamamlanan ürün tekrar dönen taşıyıcıya yönlendirilir. Taşıyıcı döner ve ürünü ikinci çıkış hattına (ÇH2) gönderir.

3.2. Problemin formülize edilmesi. Çözülmesi gereken problem, konfigürasyon 1 ya da konfigürasyon 2 işlemine devam ederken konfigürasyonun değiştirilmesidir. Diğer bir deyişle konfigürasyon 1 işlemini yaparken kontrolcününün değişip, sistemin konfigürasyon 2’nin işlemine göre işleme devam etmesi ya da tam tersi durumudur. Karşılaşılan sorun operatörün, işlemekte olan konfigürasyonun herhangi bir durumunda konfigürasyonu değiştirmek istemesidir. Örneğin konfigürasyon 1’e göre çalışan bir sistemin çalışması sırasında operatör konfigürasyon 1’in herhangi bir durumunda konfigürasyon 2’ye geçmek isteyebilir. Bu durumda sistemin çalışmasının durmaması gerekmektedir.



ŞEKİL 3. Üretim sisteminin konfigürasyonları.

Problemi aşağıda verilenlerle formüle edebiliriz. Sistem: G , kontrol edilemez olaylar: Σ_u , konfigürasyon kümesi: \mathcal{C} , yeniden yapılandırmayı başlatan olayların kümesi: Σ^{yeni} , her bir konfigürasyon $\rho \in \mathcal{C}$ için yeniden yapılandırmayı başlatan olay: $\rho_{\text{başla}}$, yeniden yapılandırmayı bitiren olayların kümesi: $\bar{\Sigma}^{\text{yeni}}$, her bir konfigürasyon $\rho \in \mathcal{C}$ için yeniden yapılandırmayı sonlandıran olay: ρ_{bitir} , bütün olası olayların kümesi: $\Sigma^{\text{hepsi}} = \Sigma \cup \Sigma^{\text{yeni}} \cup \bar{\Sigma}^{\text{yeni}}$, her bir konfigürasyon $\rho \in \mathcal{C}$ için konfigürasyonun spesifikasyonu: K^ρ , başlangıç konfigürasyonu: $\rho_0 \in \mathcal{C}$, her bir konfigürasyon $\rho \in \mathcal{C}$ için konfigürasyon kontrolcüsü: S^ρ örneğin $L_m(S^\rho) = \sup C(K^\rho, L(G), \Sigma_u)$ olarak tanımlıdır.

Bölüm 3.1'deki örnek için, $\Sigma_u = \{\text{dt-çh2}, \text{dt-çh3}\}$, $G = G_{\text{YB}} || G_{\text{DT}} || G_{\text{İM}} || G_{\text{ÇH1}} || G_{\text{ÇH2}} || G_{\text{ÇH3}}$, $\mathcal{C} = \{\text{konf1}, \text{konf2}\}$, $\Sigma^{\text{yeni}} = \{\text{konf1}_{\text{başla}}, \text{konf2}_{\text{başla}}\}$, $\bar{\Sigma}^{\text{yeni}} = \{\text{konf1}_{\text{bitir}}, \text{konf2}_{\text{bitir}}\}$ olarak bulunmuştur. S^{konf1} ve S^{konf2} Şekil 3'de verilmiştir.

Yukarıda verilenlere göre yeniden yapılandırma problemini formülize edebiliriz.

Problem 1. Yukarıda tanımlamalar verilmiştir. her bir konfigürasyon ($\rho \in \mathcal{C}$) için bir kontrolcü (R^ρ) oluşturulur. Bütün kontrolcüler ($S^{\text{rec}} = \|\rho \in \mathcal{C} R^\rho$) aşağıdaki durumları sağlamalıdır:

1. S^{yeni} 'de engelleme olmamalıdır.
2. İlk olarak S^{rec} , başlangıç konfigürasyonu ρ_0 'nun spesifikasyonlarını sağlamalıdır:

$$s \in L(S^{\text{rec}}) \cap \Sigma^* \Rightarrow s \in K^{\rho_0}$$

3. Herhangi bir konfigürasyona $\rho \in \mathcal{C}$ yeniden yapılandırıldığında, sınırlanmış sayıdaki olaylar gerçekleştikten sonra S^{rec} yeni konfigürasyonun spesifikasyonlarını sağlamalıdır. Sınırlama sayısı: $N < \infty$. Diğer bir deyişle her bir

$$s \in L(S^{\text{rec}}) \cap (\Sigma \cup \Sigma^{\text{rec}} \cup \overline{\Sigma}^{\text{rec}})^* \rho_{\text{st}} (\Sigma \cup \{\rho_{\text{bitir}}\})^*$$

için; aşağıdaki iki durumdan biri sağlanmalıdır:

- (a) $s = s' \rho_{\text{başla}} u$ ile $u \in \Sigma^*$ ve $|u| < N$ (bu durumda yeniden yapılandırılırken geçişler N ile sınırlanmıştır.)
 (b) $s = s' \rho_{\text{başla}} u \rho_{\text{bitir}} u'$ ve $|u| < N$ ve $u' \in K^\rho$ (bu durumda ise yeniden yapılandırılırken geçişler sınırlanmıştır ve yeni konfigürasyon yeniden yapılandırma işlemi bittiğinde tamamlanmalıdır.)

4. Yeniden Yapılandırma Kontrolcüsünün Yapısı

Bu bölümde, Problem 1'in çözümünü gösterilmiştir. Bunu yapabilmek için, pratik varsayımlar ile çözüm basitleştirilmiştir.

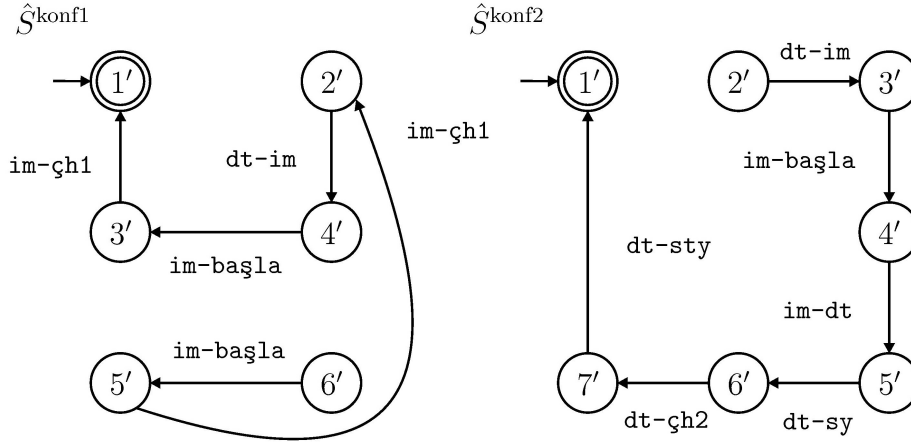
4.1. Varsayımlar. Bu makalede aşağıdaki varsayımlar yapılmıştır:

1. Her konfigürasyon sisteme aynı başlangıç durumundan başlar. Bütün $\rho \in \mathcal{C}$ için; $L_m(S^\rho) \subseteq L_m(G)$ doğrudur.
2. Her konfigürasyon sistemin başlangıç durumunda sonlanır. Bütün $\rho \in \mathcal{C}$ için; S^ρ 'deki $\{q_0^\rho\}$ için Q^ρ bir zayıf çekimdir. $\hat{S}^\rho \sqsubseteq S^\rho$ bir durum geri beslemeli kontrolcüsü için; $\{q_0^\rho\}$ in S^ρ için Q^ρ bir güçlü çekimdir.
3. Hiçbir konfigürasyonda engelleme yoktur: $\overline{L_m(S^\rho)} = L(S^\rho)$.
4. ρ_1 'den ρ_2 'ye ($\rho_1, \rho_2 \in \mathcal{C}$) konfigürasyonu değiştirirsek, ρ_2 başlayana kadar tekrar konfigürasyonu değiştirme olayının gerçekleşmediği varsayılır.

Varsayımların açıklanması: Varsayım 1) sistemin genel olası davranışı için bir kısıtlamadır, zaten pratikteki birçok örnek bu koşulu sağlamaktadır. Birçok üretim sisteminin AOS modelinde, sistemin başlangıç durumu bütün elemanların çalışmama durumudur ve sisteme ürün giriş yapmamıştır.

Varsayım 2) Varsayım 1)'in açıklamasını göz önünde bulundurarak, konfigürasyonun tekrar başlangıç durumuna, yani herhangi yeni bir ürünün sisteme giriş yapmadığı ve bütün eski ürünlerin sistemden çıkmış olduğu duruma, dönmesi olası bir davranıştır. Bu varsayım bizim uygulama örneğimizde sağlanmaktadır. Durum geri

bildirimli kontrolcüler \hat{S}^{konf1} ve \hat{S}^{konf2} Şekil 4'te gösterilmektedir. \hat{S}^{konf1} 'de kontrolcünün yeniden başlangıç durumuna ulaşabilmesi için, başlangıçta verilen durum 3 ve 4'teki **sf-rb** olayının silinmesi gerekir. \hat{S}^{konf2} ise herhangi ek kontrol gerektirmeden başlangıç durumuna dönmektedir.



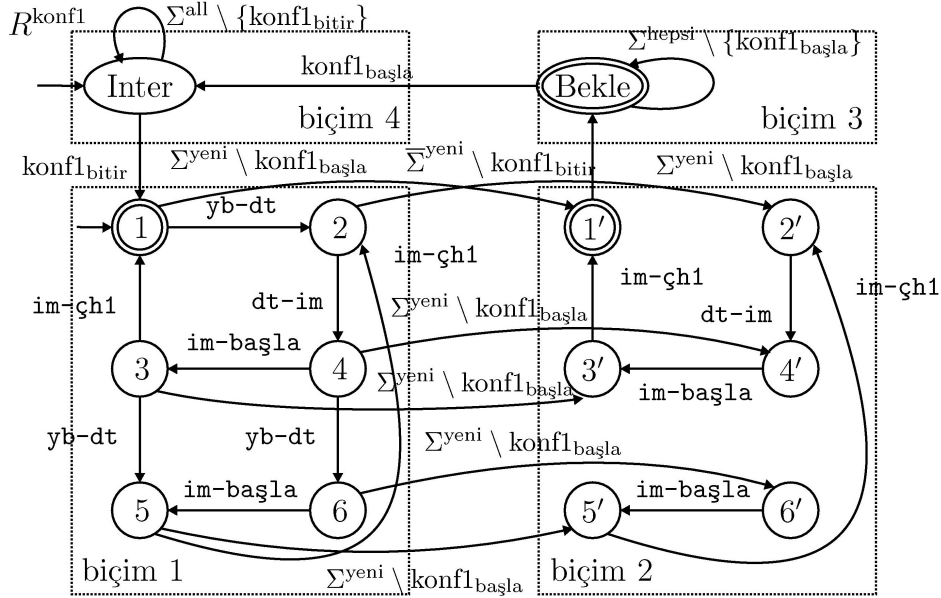
ŞEKİL 4. Sistemin başlangıç durumunun çekimi için durum geri beslemeli kontrolcüsü.

Varsayım 3) her zaman sağlanması gereken bir varsayımdır.

Varsayım 4) ise genel olarak pratik uygulamalarda geçerli olan bir durumdur. Operatör konfigürasyonu değiştirmeye karar verirse, sistem daha önce değiştirilmesi istenen konfigürasyonu tamamlamadan yeni konfigürasyona geçişi yapamaz.

4.2. Yapılandırma algoritması. Bölüm 4.1'da tanımlanan varsayımlara ve gereksinimlere göre, herbir yeniden yapılandırma kontrolcüsü (R^ρ , $\rho \in \mathcal{C}$) için dört farklı operasyon biçimi öne sürdük. Örneğin konf1 konfigürasyonunu ve oluşturulmuş yeniden yapılandırma kontrolcüsünü R^{konf1} ele alalım. Birinci biçimde, kontrolcü Şekil 3'te gösterilen S^{konf1} 'e göre davranacak şekilde konfigürasyonu sağlar. İkinci biçimde, konfigürasyon değişir ve R^{konf1} konfigürasyonun başlangıç durumuna dönmek zorundadır. Yani yukarıda belirtilen \hat{S}^ρ durum geri besleyici kontrolcüsünün belirlediği şekilde hareket eder. Bu yüzden konfigürasyon kontrolcüsünde (S^{konf1}) (örneğin birinci durumunda) konfigürasyon değiştirilmişse, durum geri besleyici kontrolcüsündeki (\hat{S}^{konf1}) belirlenmiş duruma (durum 1') gitmesi gerekmektedir. Konfigürasyonun başlangıç durumuna ulaşılmışsa, bekleme durumuna (Bekle), üçüncü biçim) yani konf1'in aktif olmadığı, yeni konfigürasyonun ise aktif olduğu duruma

gitmesi gerekmektedir. Bu durumda, aktif konfigürasyonu sınırlamayan bütün olaylar için R^{konfl} ulaşılabilir olmalıdır. Son olarak geri konfl'e dönmek istediğimizde, bir ara bekleme durumu (Inter) olmak zorundadır. Çünkü konfl hala aktif olan konfigürasyonun başlangıç durumuna ulaşmasını beklemelidir. Bu ara bekleme durumuna geldiğinde, R^{konfl} tekrar konfigürasyon kontrolcüsüne (S^{konfl}) dönmek zorundadır. Şekil 5 yapılan açıklamalara göre yeniden yapılandırma kontrolcüsünü (R^{konfl}) göstermektedir. Yeniden yapılandırmayı başlatan olay $\text{konf2}_{\text{başla}}$ biçim 1'den biçim 2'ye geçmek için, $\text{konf2}_{\text{bitir}}$ biçim 2'den biçim 3'e geçmek için, $\text{konf1}_{\text{başla}}$ biçim 3'ten biçim 4'e geçmek için ve $\text{konf1}_{\text{bitir}}$ biçim 4'ten tekrar biçim 1'e geçmek için kullanılır.



ŞEKİL 5. Konfigürasyon 1 için yeniden yapılandırma kontrolcüsü R^{konfl} .

Sıradaki bölümde algoritmayı özetleyerek yeniden yapılandırma yapısı açıklanmıştır.

Algoritma 1. Girdi: Bütün $\rho \in \mathcal{C}$, Σ^{yeni} , $\bar{\Sigma}^{\text{yeni}}$ için G , \mathcal{C} , S^ρ ;

Çıktı: her bir $\rho \in \mathcal{C}$ için R^ρ

Alt yordam

Her bir $\rho \in \mathcal{C}$ için

1. $\{q_0^\rho\}$ in S^ρ için Q^ρ bir güçlü çekici olacak şekilde $\hat{S}^\rho \sqsubseteq S^\rho$ kontrolcüsü oluşturulur.
2. Her bir $\rho' \in \mathcal{C} \setminus \{\rho\}$ için, $R^{\rho'}$ 'daki, her durumdan ($q \in Q^\rho$) ilişkili duruma ($\hat{q} \in \hat{Q}^\rho$), $\rho'_{\text{başla}}$ ile geçiş tanımlanır.

4.3. Doğrulama. Bu makalenin kapsamı içinde Problem 1'in biçimsel kanıtı açıklanmamıştır. Ama kanıtın nasıl kurulabileceği aşağıda verilmiştir. $S^{\text{yeni}} = \|\rho \in \mathcal{C} R^\rho$ koşulu göz önünde bulundurulursa:

1. İlk olarak, yeniden yapılandırma kontrolcüsü R^{ρ_0} biçim 1'deyken, diğer bütün $\rho \in \mathcal{C} \setminus \{\rho_0\}$ için yeniden yapılandırma kontrolcülerini R^ρ bekleme durumundadırlar.
2. Yeni bir yeniden yapılandırma olayı olmadığı müddetçe, R^{ρ_0} biçim 1'deki S^{ρ_0} 'ye göre hareket eder, çünkü diğer bütün yeniden yapılandırma kontrolcülerini bütün olaylara izin vermektedirler. Bu da gösterir ki; $s \in L(S^{\text{yeni}}) \cap \Sigma^* \Rightarrow s \in K^{\rho_0}$ ve Problem 1'in 2) koşulunu onaylamaktadır. Ayrıca varsayımdaki gibi S^{ρ_0} engelsiz çalıştığı müddetçe, bu konfigürasyondaki işaretli duruma her zaman ulaşılır.
3. Yeniden yapılandırma olayı $\rho_{\text{başla}} \in \Sigma^{\text{yeni}}$ oluşursa, R^{ρ_0} biçim 2'ye, R^ρ biçim 4'e geçer ve diğer bütün konfigürasyonların kontrolcülerini bekleme durumunda kalırlar. \hat{S}^{ρ_0} 'in başlangıç durumuna ulaşana kadar R^{ρ_0} biçim 2'yi takip eder. Varsayım 2)'de öngörüldüğü üzere bu durum belirli sayıdaki geçişten sonra oluşur.
4. \hat{S}^{ρ_0} 'in başlangıç durumuna ulaşılırsa, bütün yeniden yapılandırma olaylarındaki tek olası olay ρ_{bitir} 'dir. Bu nedenden dolayı, R^{ρ_0} biçim 3'e (bekleme), R^ρ biçim 4'ten biçim 1'e (aktif) geçer ve diğer bütün yeniden yapılandırma kontrolcülerini ise biçim 3'te (beklemede) kalırlar. Böylece Problem 1'in 3) koşulu sağlanmış olur.
5. Şimdi 1. adımla aynı duruma sahibiz. Varsayım 1)'den dolayı bir yeniden yapılandırma kontrolcüsü kendi konfigürasyonunu başlangıç durumundan başlayarak takip eder (R^ρ biçim 1'de) ve diğer bütün yeniden yapılandırma kontrolcülerini biçim 3'tedir. Bu durumda her bir yeniden yapılandırma durumu için aynı operasyon aşamaları tekrarlanır.
6. Bunlarla birlikte, Problem 1'deki koşulların yapılanmış kontrolcüler tarafından sağlandığı sonucuna varabiliriz.

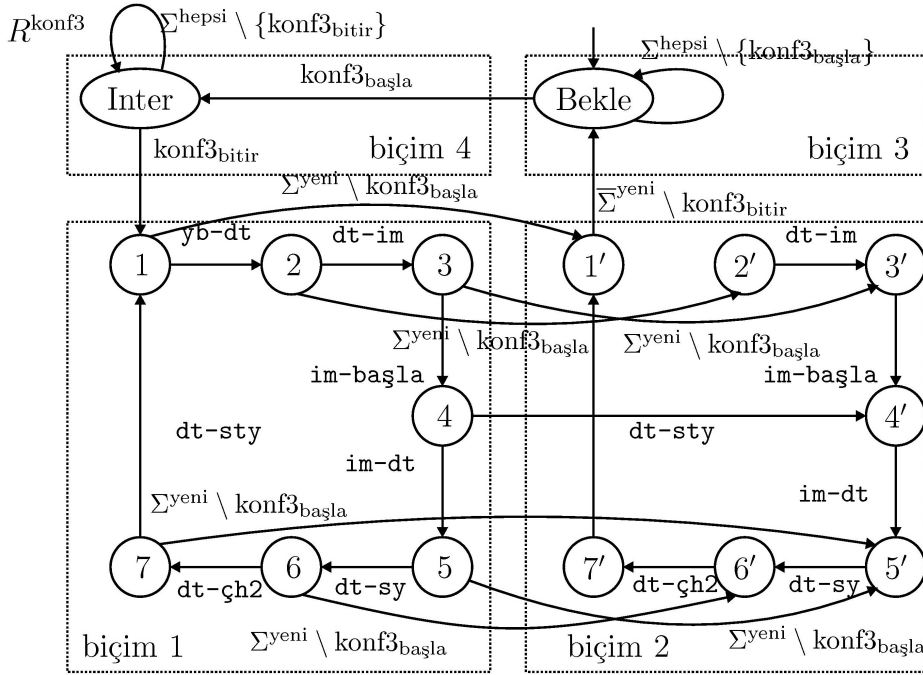
4.4. Tartışma ve genişletimler. Bir önceki bölümde açıklanan yöntem, sistem tasarımı bilinen her bir konfigürasyon için yeniden yapılandırma kontrolcülerini oluşturabilmemizi sağlar. Buna rağmen, çalışmakta olan bir sisteme yeni konfigürasyonlar eklenebileceği de göz önünde bulundurulmalıdır. Şekil 1'de gösterilen ele aldığımız örnekte, ikinci konfigürasyonla aynı yolu izleyen fakat ürünlerin ÇH2 yerine

ÇH3'den çıkış yaptığı, üçüncü bir konfigürasyon eklenebilir. Bu durumda, baştan sona yeni bir sistem oluşturmak istenmeyen bir durumdur. Bunun yerine sistemin istediğimiz gibi çalışması için yeni bir yeniden yapılandırma kontrolcüsünü sisteme eklemek daha basit ve kabul edilebilir bir yöntemdir.

Bölüm 4.2'de anlatılan yöntem, pratikte yeni bir konfigürasyonun sisteme eklenmesine aşağıdaki şekilde izin verir.

1. Yeni konfigürasyonun başlama ve bitiş olayları Σ^{yeni} ve $\bar{\Sigma}^{\text{yeni}}$ kümelerine eklenir.
2. Her yeni konfigürasyon için Algoritma 1'e göre yeniden yapılandırma kontrolcüsü oluşturulur.
3. Yeni kontrolcü de varolan kontrolcülerle eş zamanlı çalıştırılır.

Yukarda anlatılan yöntem incelediğimiz örnekte konfigürasyon 3 (conf3) uygulanır. Bu durumda yeniden yapılandırma olay kümeleri $\Sigma^{\text{rec}} = \{\text{conf1}_{\text{st}}, \text{conf2}_{\text{st}}, \text{conf3}_{\text{st}}\}$ ve $\bar{\Sigma}^{\text{rec}} = \{\text{conf1}_{\text{fin}}, \text{conf2}_{\text{fin}}, \text{conf3}_{\text{fin}}\}$ olarak değişir. Ayrıca conf3 için yeniden yapılandırma kontrolcüsü R^{conf3} , konfigürasyon 2'ninkine (R^{conf2}) benzer şekilde oluşturulur. Sonuç Şekil 7'de gösterilmiştir.



ŞEKİL 7. Eklenen konfigürasyon 3 için yeniden yapılandırma kontrolcüsü R^{konf3} .

Daha önceden varolan yeniden yapılandırma kontrolcülerini (örnekteki R^{konf1} ve R^{konf2}) tekrar değiştirmek gerekmez. Bunun nedeni Σ^{yeni} ve $\bar{\Sigma}^{\text{yeni}}$ kümelerindeki

tek olaylara bağılı olmamalarıdır. Yalnızca kendi yeniden yapılandırma olaylarına ve bütün olarak yeniden yapılandırma olaylarının kümesine bağılıdır. Bu ifade Şekil 5'te gösterilen kontrolcüyle doğrulanır. Bu kontrolcüdeki yeniden yapılandırmanın olaylarının geçişleri yalnızca kendi yeniden yapılandırma olaylarına ($\text{konf1}_{\text{başla}}$ ve $\text{konf1}_{\text{bitir}}$) veya kendi yeniden yapılandırma olaylarının dışında kalan bütün yeniden yapılandırma olaylarının kümesine ($\Sigma^{\text{yeni}} \setminus \{\text{konf1}_{\text{başla}}\}$ ve $\bar{\Sigma}^{\text{yeni}} \setminus \{\text{konf1}_{\text{bitir}}\}$) bağılıdır. Yani ayrı ayrı $\text{konf2}_{\text{başla}}$, $\text{konf3}_{\text{başla}}$, $\text{konf2}_{\text{bitir}}$ ve $\text{konf3}_{\text{bitir}}$ olaylarına bağılılık yoktur. Bu nedenden yeni bir yeniden yapılandırma kontrolcüsü R^{konf3} eklendiğinde, R^{konf1} kontrolcüsünü değiştirmek gerekmez. Aynı durum R^{konf2} için geçerlidir.

5. Sonuç

Bu makalede, ayrık olaylı sistem için gözetimli kontrol çerçevesinde yeniden yapılandırılabilir üretim sistemlerinin kontrolü ele alınmıştır. Üretim sisteminin elemanlarının ve her bir konfigürasyonun kontrolcüsünün otomat modelinden başlayarak her bir konfigürasyon için yeniden yapılandırma kontrolcüsü oluşturulmuştur. Eğer konfigürasyon aktif ise kontrolcüsü konfigürasyonu çalıştırmakta, ama başka bir konfigürasyon aktif duruma geçirilirse kontrolcü aktif olmayan duruma geçmektedir. Farklı konfigürasyonların yeniden yapılandırma kontrolcüleri eşzamanlıdır, bu da sınırlı bir zamandan sonra konfigürasyonlar arasında geçiş yapılmasını ve kontrol edilen üretim sisteminin engelsiz çalışmasını sağlamaktadır. Yöntemimizi bir üretim sistemi örneğinin yeniden yapılandırma kontrolcüsünün tasarımı ile açıkladık. Ek olarak geliştirdiğimiz çalışmanın, diğer yeniden yapılandırma kontrolcülerinde bir değişiklik yapmadan, yeni bir konfigürasyon için yeniden yapılandırma kontrolcüsünün sisteme eklenebildiğini gösterdik.

Kaynaklar

- [1] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy and H. Van Brussel, Reconfigurable manufacturing systems, *CIRP Annals – Manufacturing Technology* **48** (1999), 527–540.
- [2] M. G. Mehrabi, A. G. Ulsoy, and Y. Koren, Reconfigurable manufacturing systems: Key to future manufacturing, *Journal of Intelligent Manufacturing* **11** (2000), 403–419.
- [3] Y. Koren, *The Global Manufacturing Revolution*, Wiley, 2010.
- [4] H. A. ElMaraghy, *Changeable and Reconfigurable Manufacturing Systems*, Springer Series in Advanced Manufacturing, 2009.
- [5] P. J. Ramadge and W. M. Wonham, Supervisory control of a class of discrete event processes, *SIAM Journal on Control and Optimization* **25** (1987), 206–230.

- [6] R. Kumar, S. Takai, M. Fabian and T. Ushio, Maximally permissive mutually and globally nonblocking supervision with application to switching control, *Automatica* **41** (2005), 1299–1312.
- [7] E. W. Endsley, E. E. Almeida and D. M. Tilbury, Modular finite state machines: Development and application to reconfigurable manufacturing cell controller generation, *Control Engineering Practice* **14** (2006), 1127–1142.
- [8] J. Li, X. Dai and Z. Meng, Automatic reconfiguration of petri net controllers for reconfigurable manufacturing systems with an improved net rewriting system-based approach, *IEEE Transactions on Automation Science and Engineering* **6** (2009), 156–167.
- [9] H. E. Garcia and A. Ray, State-space supervisory control of reconfigurable discrete event systems, *International Journal of Control* **63** (1996), 767–797.
- [10] K. W. Schmidt, Computation of supervisors for reconfigurable machine tools, *Workshop on Discrete Event Systems*, 2012.
- [11] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Springer, 2008.
- [12] W. M. Wonham, *Supervisory Control of Discrete-Event Systems*, Lecture Notes, Department of Electrical and Computer Engineering, University of Toronto, 2010.
- [13] S. Lafortune and E. Chen, The infimal closed controllable superlanguage and its application in supervisory control, *IEEE Transactions on Automatic Control* **35** (1990), 398–405.
- [14] Y. Brave and M. Heymann, Stabilization of discrete-event processes, *International Journal of Control* **51** (1990), 1101–1117.
- [15] Y. Brave and M. Heymann, On optimal attraction of discrete-event processes, *Information Sciences* **67** (1993), 245–276.